Modeling, Solution and Application of Complex Supply Chain Networks

by

Ugur Kaplan

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the of the Requirements for
the Degree of

PhD

in

Industrial Engineering and Operations Management

Koc University
July 2012

ABSTRACT

In this thesis, the main focus is the supply chain networks in terms of modeling, solution and applications. Networks are one of the main representation tools for the systems in engineering, science and management. Supply chain networks are the specific networks that are used to represent the flows of materials, information and financials from one physical node to others. The supply chain context is very broad, we consider it in 3 parts: Modeling, solution and applications, during this thesis.

In the modeling part the development of new modeling technique is described in detail. In conventional supply chain network modeling, the flows and the inventories are modeled separately for each flow and node. This makes modeling and re-modeling difficult to maintain. In this new modeling technique, the network topology that represents the flows, entities and inventories are stored in multi-dimensional matrix. Therefore model-data independence is succeeded. This makes models lean and easy to maintain.

In the solution part, we have focused on the non-integer representation of the batch behaviors in the supply chain network. In conventional modeling, this batch behavior is modeled by using integer variables. However this makes the model combinatorial and difficult to solve. Therefore, in this thesis, we have developed new solutions techniques to model batch behavior without using integer variables.

Finally, in applications part, we have developed a new version of the VRP (Vehicle Routing Problem). The capacities for this new version of the VRP show combinatorial behavior. A new algorithm is developed to solve this new VRP problem. It is known that VRP is one of the most difficult problems in the literature. With addition of new combinatorial capacity constraints, some models can become computationally intractable. With this new developed algorithm, the capacity constraints can be handled in polynomial time.

ÖZET

Bu tezin temel odak noktası tedarik zincirlerinin modellenmesi, modellerin çözümü ve tedarik zincirlerinin uygulamasıdır. Ağ yapısı gerek temel bilimler ve mühendislik gerekse yönetim bilimi tarafından sıkça kullanılan bir yapıdır. Tedarik zinciri ağları ise, bilgi, mamul, yarı mamul, ve finansalların bir yerden bir yere akısını izleyen, ağ yapısının daha özel bir halidir. Tedarik zinciri sistemleri ve ağı içerik olarak çok geniş olduğu için, 3 temel baslık altında inceleyeceğiz: Modelleme, çözüm ve uygulama.

Modelleme kısmında geliştirilen yeni bir modelleme tekniği detaylı olarak incelenmiştir. Tedarik zinciri modellemesi sırasında konvansiyonel modelleme teknikleri her bir akis ve envanteri ayrı ayrı modeller. Bu modelleme surecini ve var olan modelin geliştirilmesini çok zorlu hale getirir. Yeni geliştirilen teknikle, bütün ağ topolojisi çok boyutlu bir matriste saklanır. Bu yöntemle hem model-data bağımsızlığı sağlanmış olur, hem de model yaratılması ve geliştirilmesi daha yalın ve kolay hale gelir.

Çözüm kısmında tedarik zincirlerindeki yığın davranışlarının tamsayılı programlama kullanmadan çözüm ve modellenmesi üzerine duruldu. Konvansiyonel modelleme sırasında yığın davranışlar hep tamsayılı değişken kullanılarak modellenir. Fakat bu tamsayılı değişkenlerin kullanımı, modeli ve sistemi çözümü çok zor hale getirebilir. Bu nedenle, tezin bu kısmında, yığın sistemleri tamsayılı değişken kullanmadan çözme üzerine geliştirilen metotların üzerinde durulacaktır.

Son olarak uygulama kısmında, Araç Rotalama Probleminin (ARP) özgün bir versiyonu üzerinde çalışılmıştır. Bu yeni özgün ARP'de kapasite birleşisel bir karakter göstermektedir. Bu kısımda özgün ARP problem açıklanmış ve çözümü için bir algoritma dizayn edilmiştir. Bilindiği üzere ARP en zor eniyileme problemlerinden biridir. Yeni

eklenen birleşsel kapasite kısıtları ile problem bazen uygulamalı olarak çözülemez hale gelmiştir. Tezin bu kısmında yeni geliştirilen algoritmalar ile, birleşsel kapasite kısıtları, polinom zamanda çözülebilir hale gelmiştir.

ACKNOWLEDGEMENT

First I thank my advisor Metin Turkay for his continuous support during my B.S. and M.S. years. He always listened to me and gave invaluable advices in all steps of my thesis. In addition to optimization techniques, he thought me how to express my ideas, how to approach a problem and how to bring out good ideas. He shared his experience with me both in academic and real life and he supported me during each phase of my academic and personal development. I thank to Alper Erdogan, Onur Kaya, Sibel Salman and Fadime Uney Yuksektepe for participation in my thesis committee and for their supportive comments on my thesis.

I am also grateful to all my friends, Gokcen, Fatih, Onay, Gokhan, Can and Sener for being wonderful classmates, officemates, homemates and especially friends. Their close friendship makes this thesis enjoying.

During the course of my thesis, at Koc University (2007-2012) I was supported by TUBITAK's scholarship program for PhD students. I am thankful to TUBITAK for their support for my academic studies.

Last, but not least, I thank my family, my father Huseyin, my mother Kamile and my broter Ozgur, for their unconditional support and encouragement. I could not succeed anything without their support. I am very lucky because I have such a wonderful family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

A supply chain system (SCS) or a supply network is a set of entities connected with each other with a pre-defined relationship. In conventional understanding, a supply chain system includes one or many, producers, central and regional warehouses, retailers and customers [1]. The products (sometimes service) flow from production site to central warehouse to regional warehouses and finally to retailers [2]. As soon as the product arrives to retailers, they become available for customers. The spread of product or service is realized from central entity to distributed customers. Therefore this system can be classified as a distribution system. However, there exist two main differences between a supply chain system and distribution system: the production activities and the flow of financials. Conventional distribution systems does not focus on the production or creation of service. Also the financial flow is not one of the main interests in the distribution systems. In contrast, flow and optimization of financial entities is very important for the supply chain networks. In addition, the production is one of the main concerns of the supply chain system or network. The concept of the supply chain network is illustrated in the Figure 1.1.

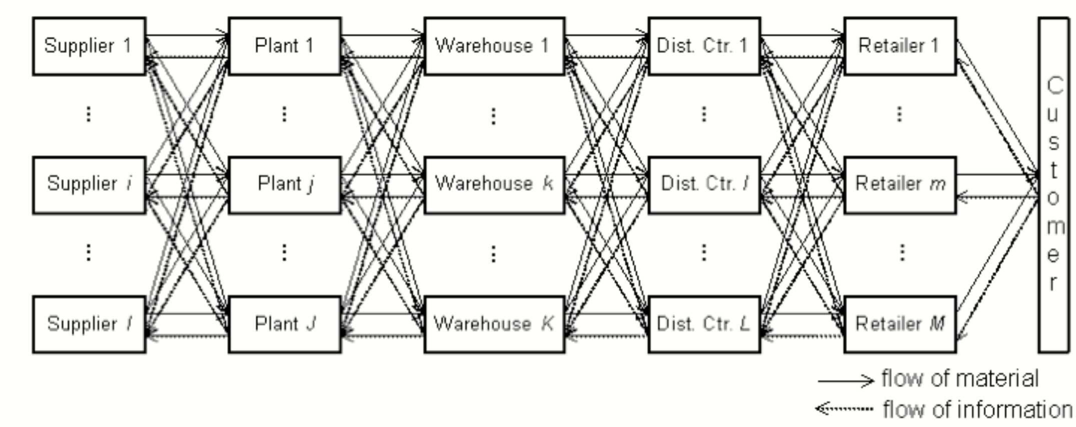


Figure 1.1: Sample supply chain network. Flow of materials and products is illustrated bold arrows and the flow of information and financials are shown with dashed arrows.

The supply chain system that is illustrated in the Fig 1.1 is a conventional one. In other words, the end product flows from upstream (producer or supplier) to downstream (customer) and the financial and information flow from downstream to upstream[3]. As seen from the representation, this system is an open system. In other words, the products, information and financials do not return to system and they are disappeared either in downstream or upstream nodes. In addition some supply chain networks are classified as a closed loop supply chain system. The representation of a Closed Loop Supply Chain system can be seen in Figure 1.2. The difference of the closed loop system from the open one is the bi-directional structure of flow of products. As seen in Fig. 1.2, the end products are distributed from upstream to downstream. However besides this, the end products, which have defects or the ones that is used, are collected by different nodes to refurbish, repair or recycle them. The system becomes closed loop because the produced and distributed products are collected and they do not disappeared in the system. After collected they are re-used and re-inserted into system in the form of raw material or refurbished product.

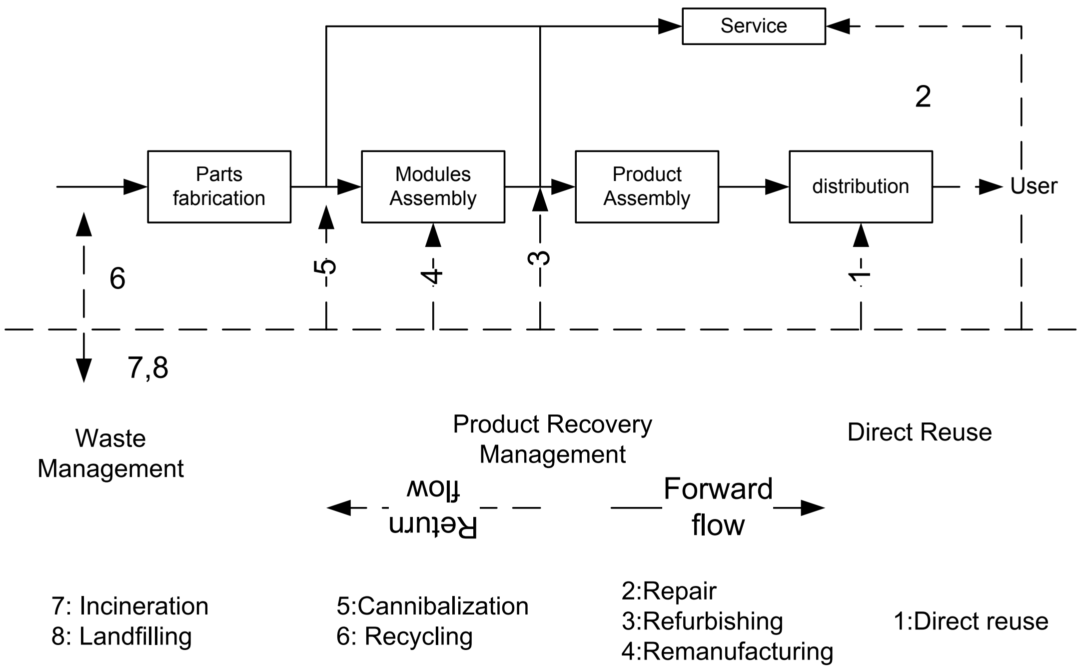


Figure 1.2: A closed loop a supply chain system. Products are both distributed and collected.

The system represented in Fig. 1.2 can be classified as a Closed Loop Supply Chain System. Sometimes this system is called as a Reverse Logistics Network or System. Indeed this naming is wrong because a reverse logistics network and closed loop system are not the same. The closed loop system is a meta-system and reverse system is the specific version of it. The reverse logistics network begins with the collection of the end products. Even if the reverse logistics network can be seen as a reverse of the forward chain at first glance, it is different and both computationally and as a way of modeling more difficult than the forward one.

The reverse logistic network design phase involves first the definition of the actors and actions; possible actors are the forward logistic network, demand points for returned products, new processing units, producers and many others [4]. The actions that take place in the reverse logistic are the collection, transportation, storing, processing, recycling, incineration and disposal of end products. Different from forward logistics, the activities and actors are not always the same. In some of the reverse logistics networks, one actor can take a role but in another one it cannot play any role. This is due to uncertain characteristics of the end products.

In general, it is not known where and how the end products come from. In addition, the condition of the products cannot be known in advance. Some of the products can be refurbished, but some of them can only be recycled depending on the condition of the end product. Also the quantity of the returned products shows uncertainty. This high level of uncertainty complicates the design of the reverse logistics network. Both quality and quantity of the products returned are the determinants for a suitable network structure. The quantity of the returned products determines the size and locations of the warehouses. Because the return rate shows stochastic behavior, simultaneous transportation of end product to the processing unit will decrease the performance of the system. Therefore, at some strategic points warehouses should be built. Quality of the products will also affect the network design. Some of the products requires high level processing; therefore without dividing it to all its micro-components (on module level) required operations can be completed. However, some of the returned end products require low-level processing. Because different operations takes place on the different locations, specific characteristics of the products affect the network design [5]

Also the forward network cannot be used without any modifications because reverse and forward logistics shows difference [6]. First of all, working philosophy is totally different for reverse and forward logistics. Forward logistics systems are mostly pull systems. It is triggered with customer demand. The main aim of the forward logistics systems is the customer service. All the parties of the chain including suppliers adjust itself with respect to this aim. However, reverse logistics systems are both pull and push system because there are two clients in the system, disposer and the re-user. In this multi-client system, reverse logistics system both have to satisfy disposal rates and customers needs.

In the forward logistics, all activities are done to produce single product. All production maps are pre-determined and responses of all parties are known with respect to customer trigger. However in the reverse system, there is not only one trigger unit. Besides the relation between systems triggers (end user - disposer) is not symmetric. Therefore during the reverse logistics process, the discarded products produce uncertain amount of subassemblies and demand to these subassemblies does not show systematic relation between the clients of the reverse logistics. In addition, the values and complexities of these subassemblies are not distributed homogeneously. Therefore efficient design of reverse logistics network should include multi-echelons. However the possible number of echelons in forward logistics is limited.

Although the reverse logistics and forward logistics shows different characteristics, the optimal operations of reverse logistics can only be achieved by integrating it forward logistics. One of the possible integration schemes is shown in Fig. 1.3 [4].

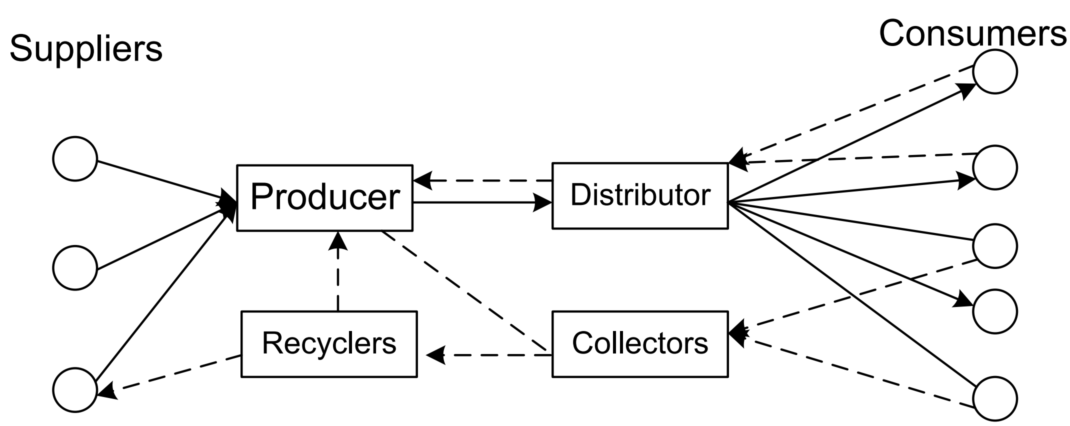


Figure 1.3: Integration of reverse and forward logistics [adopted from Fleishmann et. al., 1997].

As presented in Fig. 1.3, the parties in the network have more than one task. Even if the distributor and collector are shown as a two different parties, they can be same company or service provider. The main aim of this division is to show the difference of their task.

Although in Fig.1.3 the integration scheme of forward and reverse logistics is presented, at the present there are very few models treating reverse and forward logistics simultaneously. These models succeed the integration by locating multi-purpose facilities for both reverse and forward logistics systems. However integrated routing systems are not combined with facility locations systems. In industrial practice, sector based application of integration can be found. This integration is done for collection and distribution of beverage bottles. In this application, because bottles show homogenous behavior and the containers used for distribution can also be used for collection, the integration can done by only adding an extra transportation cost adding to forward channel. However, integration is not done for the complex closed-loop systems similar to the one in Fig. 1.2 [7].

During the production planning in the reverse logistics systems, the activities in Fig. 1.2 take place. Dependent on the product produced in the system, the combination of these activities can be planned. In the production planning of the reverse logistics systems, the schedule and level of these activities are determined. In the inventory control part, the level of inventory of each module, subassembly and end product is determined for each facility that the activities in Fig.1.2 take place. Inventory level modification is occurred as presented in Fig. 1.4 [4]. The inventory management scheme in Fig. 1.4. is very generic one, but it should be pointed that, with respect to different products more simpler or complex schemes can be obtained.

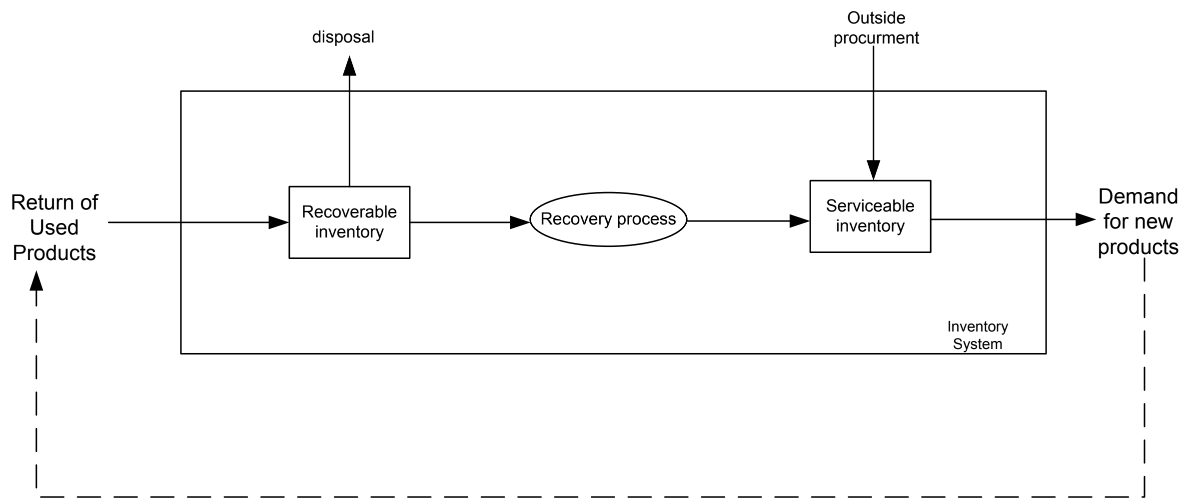


Figure 1.4: Inventory management scheme for reverse logistics.

If when the reason for people to return their product is analyzed, the inventory problem for reverse logistics systems can be understood. According to result of the study of the [8] the main reason of the returns:

a.) Return to vendor for repair
b.) Service / Maintenance
c.) Sales agent ordering error
d.) Customer order error
e.) System processing error
f.) Shipping wrong material
g.) Incomplete shipping
h.) Wrong quantity
i.) Duplicate shipment
j.) Duplicate customer order
k.) End of life

The inventory scheme and production schedule will show variation according to the reason of return. For example if a product is returned because of the reason *k* then, this product should be disassembled into its parts or it should be recycled. Then the inventory of micro subassemblies will increase. However, if it is returned because of the order errors, then without doing any operation on the returned product, the inventory of end product will increase by 1.

Because of these complications, inventory control and planning and scheduling of the reverse logistics systems are difficult. Daniel Guide, V et al. (2000) determine seven characteristics of the reverse logistics system that complicate the management and control of it [9]:

a.) Uncertain timing and quantity of returns
b.) The need to balance demands with returns
c.) The need to disassemble the returned products
d.) The uncertainty in materials recovered from returned items
e.) The requirement for the existence of the reverse logistics network
f.) The complication of the material matching restrictions
g.) The problems of stochastic routings for materials for repair and remanufacturing operations and highly variable processing times.

Also as stated before, during the reverse logistics activities, different subassemblies coming form different products can be combined with new parts to produce totally different products. This makes the production planning very complicated with the corresponding inventory control system.

Traditional MRP for re-manufacturing is not always feasible. The generic re-manufacturing scheme is presented in Fig.1.5.

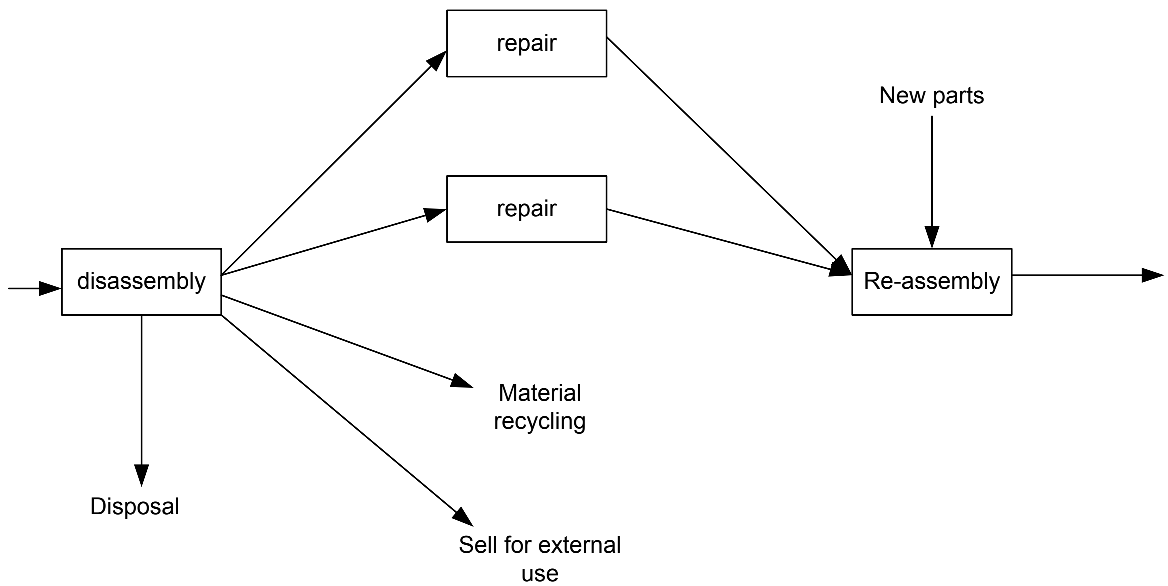


Figure 1.5: Remanufacturing scheme.

There are two main problems in the remanufacturing scheme in Fig.1.5: The first one the mismatches between supply and demand [4]. The main reason of this mismatch is the different characteristics of the components in the disassembly of the returned components. Some of these components are wanted but some of them are unwanted components. The second major problem is the trade-off between reusing return components or outside procurement. During the remanufacturing all of the parts can be obtained from disassembly, or some of them can be obtained from outside procurement, or all of them can be obtained from outside. The main decision that should be given at this stage is the level of usage of outside procurement and disassembly part [10]. The decision of always using disassembly components are not always feasible, therefore use of outside procurement is inevitable. There are two factors that determines decisions between these two options; quality of disassembly and economic considerations. Therefore the quality of materials obtained as a result of disassembly is also important.

In the previous part, the concept of the supply chain system and the versions of it are described. Some of them classified as forward supply chains. Some of them classified as reverse supply chains. In generic, we accept that a close loop supply chain system is the most general supply chain system and the forward or reverse ones are the specific version of it. The modeling of each network, on the other hand, shows many similarities. The main reason of this, independent from direction of the materials and financials, the flow logics are same. For that reason, the models of them are very similar. The main difference between the supply chain systems is the understanding behind the systems. The difference is can be classified as a perceptional difference. This perception difference is realized in the way of the modeling. Each supply chain understanding is modeled in different way. These different models are also solved in the different ways. In the next sections, the modeling, solutions different supply chain systems are discussed and basic applications of it are described.

1.1 Modeling

Although there are many definitions of the term of “model”, the one that we have used during this thesis is the mathematical model. Also there are many types of the mathematical model, such as forecasting models or simulations models [11]. However the scope of this thesis only includes the optimization models of the supply chain systems or supply chain networks. These models are concerned with managing production, transportation, warehousing, and inventory stocking activities. Also sometimes the retailer or customer behaviors are added into system. By this way the financial inputs of the system are introduced and the objectives of the models can be minimization of the cost or maximization of the profit.

In the models of the supply chain, solutions of the different problems in the context are focused. For the some of models, the main concern is the solution of the logistics problems, in some of them production management problems and some are inventory management problems [12].

Logistics is mainly concerned with the flow of the materials or products. It can be classified as a distribution models. Logistics models are complex models that involve complex decisions about transportation modes, carriers, vehicle scheduling, routing and many other activities that serve to move products through supply chain. Optimization models and modeling systems have been successfully applied to these problems and applied research continues to seek faster and more powerful models and methods. Most of these models are concerned with the minimization of the cost of transportation, warehousing, inventory, order processing and information systems and lot quantity cost, while achieving a desired customer service level [13].

Product management models are also varies between industries or problems. Some models are classified as a continuous and some are classified as a discrete model. Process manufacturing, which characterizes production in facilities such as oil refineries that is run continuously. In the discrete models, the discrete production systems are represented. At the tactical and strategic planning models, the various classes of productions planning problems can often modeled by general purpose models and modeling systems that address multistage and multi-period planning decisions. These models are very important for coherently linking detailed operational models with more aggregate tactical models [14].

Inventory management models are interested with the holding costs, shortage costs, and demand distributions for the products specified at a detailed stock keeping units level. In this thesis, the inventory models are the concern of the optimal inventory is integrated with other supply chain decisions, because inventory cost or holding cost is the only one part of the whole supply chain network. Incorporating inventory decisions into supply chain optimization models is difficult because they involve parameters and relationships, which are not easily represented in optimization models.

As described above, different models exist in the literature for different scope of the supply chain. Each model focuses on the different aspect of the system. However this thesis focuses especially on the production and distribution models. Therefore main aspects of this thesis are the purchasing, production, scheduling, inventory, warehousing, and transportation. All these aspects are integrated in the models that we are focused on in this thesis. In other words, in one model all these aspects are considered simultaneously. For that reason these kinds of models are generally used in the scope of strategic or tactical level[15].

Tactical or strategic supply chain models are generally used to model a network. In this network generally entities exist. For example, producer, warehouse, or/and customer. If a closed loop supply chain or reverse logistics network is modeled, the collector or recycler entities are added in to the system[16]. Generally these models are built in multi-period format. In other words first the network is created and built. Then the numbers of time steps or periods are determined. This means that the entities whose roles are determined in the previous step, replicates their duties during these time steps with respect to relationships or flows between them. This relationship generally shows the flows of products, financials, and information and their rules.

The models that this thesis focuses on are the special types of the models whose specifications are determined in the previous paragraph. A simple representation of this model is seen in Fig. 1.6. The mathematical representation of the model shown in the Fig. 1.6 is the general form of the networks that this thesis is interested in. After the network and relations are established, the mathematical model formulations are developed. In general this model is built for each entity and relations separately. For that reason, modeling part of the process is one of the most important parts of the supply chain management and optimization process. Mathematical model should be one of the most concrete parts of the management and optimization process because the characteristics of the system are mostly represented by the mathematical model.

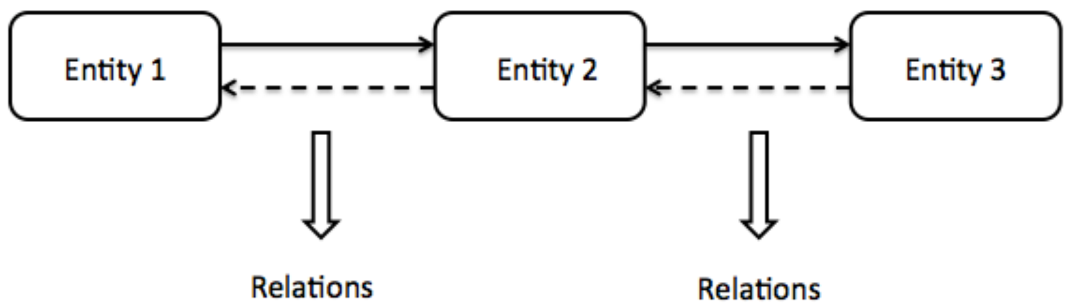


Figure 1.6: Basic supply chain model.

However all of the systems can be changed as the time passed. This property comes from the nature of the all systems. Therefore even if it is pointed out that, the mathematical model should be the most concrete part of the system, it can be shown changes as time passing. Good models should have some properties. Some basics of these properties are the [17]

- Persimony
- Tractability
- Conceptual insightfulness
- Generalization
- Falsifiability
- Empirical consistency

- Predictive precision

Many other properties can be added to this list but these 7 properties are the most important ones. These properties will be discussed in detail. However this list shows the properties of the good model in a subjective way. In other words, for example one model can be general enough for one person, but the reverse can be possible for the other person. Therefore they are important but subjective. Also these properties are not general enough. For that reason it cannot be said that all these properties are the sine qua non for a model.

According to this thesis a good mathematical model should also include the following two properties (other than written 7 properties):

- Modeling with less effort
- Extension with less effort.

Modeling part of supply chain management is very important and should be done carefully. Also time required to model a system during supply chain management process has a big ratio. For that reason the effort spending to model a system should be decreased. Also models are not unchangeable. They are changed, upgraded or extended. Some time this re-modeling part requires time close to time for modeling. Besides during extension of model some mistakes can occur. Therefore a modeling procedure should be developed that decrease the total time for modeling and re-modeling and with this style the error occurs during modeling and remodeling should be decreased too.

In this thesis we have developed a new modeling technique for tactical or strategic supply chain network modeling. With this new technique, both modeling and remodeling requires less effort and they will become lesser prune to small mistakes. This technique will be discussed in the Chapter 2 of thesis with examples and philosophy behinds it.

1.2 Solution

Modeling is the first step of the supply chain management process. As soon as model is built the result of the model is required. To obtain the result of model, the mathematical model should be solved. To solve a model, some algorithms are required.

The scope of this thesis is the modeling, optimization and application of the supply chain networks. Therefore, the models existing in the thesis are related with supply chain systems. For that reason, the algorithms required and developed on the scope of thesis are for supply chain systems.

Role of models in supply chain decisions making is very important, but without solving them a model is by itself is just a tool that shows the dynamics of the system visually or logically.

The solutions methods that are usable depend on the characteristics of the model. In other words, there is not one solution technique for all of the models. Each kind of model requires different solutions techniques. The context of this thesis is strategic and tactical level supply chain modeling and decision-making. Even if there exists many different model types in this context we focused on aggregate modeling and supply chain network optimization.

For this thesis we made the classification of models in the following types:

- Linear models
- Integer linear models
- Non-linear models.

Even though aim, scope and types of the supply chain models are different, the types of the mathematical models can be classified into these 3 types[18].

Different types of solution algorithms exist for all of these model types and the complexity of them show difference between each others. Also exactness of algorithms also shows difference. Some algorithms give exact optimal solution while others can give non-exact (possibility of sub-optimality). For each kind of models the preferred solution algorithms are discussed in details one by one.

For the linear models, simplex or interior optimization are the exact solution methods used. LP models and methods for optimizing them play central role in all types of supply chain applications. The model and methods were originally devised to optimize the allocation scarce resources in complex systems.

In manufacturing company or supply chain network, many activities compete for resources, such as machine capacity at a plant or inventory of a finished product at a DC. The available quantities of some resources may be insufficient to accommodate all the demands placed on them. Moreover some activities may consume several resources in producing desired outputs. Linear programming models allow resources to be allocated across the entire system being analyzed to determine how scarce resources can be optimal used.

In the linear programing like in supply chain management, the problem has an objective of maximization or minimization of some predefined objective function subject to some linear constraints. For several reasons, LP models and algorithms are the mainstream of the supply chain management and mathematical optimization. The main reason is that, LP models are very easy to model and solve. The five fundamental properties of linear programming models are as follows [19]:

- Linearity
- Separability and additivity
- Indivisibility and continuity
- Data known with certainty.

Because of these properties the solution of the linear programming systems are the easiest system that is solvable. Therefore LP models are frequently used during the supply chain system management.

The LP models are solved generally by simplex method. Simplex method is empirically efficient algorithm much more so than we would believe from a theoretical analysis of its performance. If there is *m* equations and *n* variables in the system, than theoretical upper bound on the number of iterations is $n!\ (n\text{-}m)!\ /\ m!$. This number grows exponentially with *n* and *m*. The efficient empirical behavior can be explained by two mathematical properties of method. First the method limits its analysis to a certain class of solutions, called basic feasible solution. These solutions are set of (m x m) invertible matrices. The second property is that the objective function monotonically increases. With each increase, the number of non-singular matrices made up of m columns of the model that might yield a still higher objective function value decreases, thereby limiting the number of possible basis changes at the next iterations. In the simplex method, theory of linear equations is used. Therefore the constraints in the form of in-equality are first converted into equality by using slack variables and than the linear algebraic rules are applied to these sets of linear equations to find the optimal value.

The other most frequently used models in the supply chain management models are mixed integer-programming models. Mixed integer models are the generalized form of the linear models. Only difference is the existence of integer variables in the constraints. Because it is a generalization of the linear programing models, the variables other than integer ones are continuous that takes any non-negative values.

There are two kinds of integer variables: binary and general integer variables. The binary integer variables can only take value of 0 or 1. The general integer variables can take any integer variables. The binary variables are generally used to represent the logical impressions that cannot be captured by linear programming. In the context of supply chain binary variables can be used in many ways: sequencing, routing decisions, scheduling of machines or work force, to show the fixed cost, economies of scale or policy restrictions.

Branch and Bound is one of the best algorithms that solve the mixed integer programming models [20]. It is a method that systematically searches the solutions space to find the optimal solution. The solution time of the branch and bound procedure is less predictable than the simplex method, which it employs as a subroutine. This method is computationally NP-Hard [21], therefore the running time of the algorithm increases exponentially as the instance size if the problem increases. The branch and bound method can be guaranteed to find an optimal mixed integer programming solution by an implicitly exhaustive search of the set 0-1 solutions to a mixed integer programing model. This method is very flexible and personal experience is required in selecting and implementing intelligent branching and search strategies. The efficacy of branch and bound search is very dependent on finding a good initial feasible solution quickly. Until initial feasible solution has been found, its goal is to eliminate fractions and achieve mixed integer feasibility as quick as possible. Once a feasible has been found, the method seeks a better feasible solution to prove that it is optimal or demonstrably good. By demonstrably good feasible solution, we mean one that is proven to be within a specified tolerance of an optimal solution as measured by its objective function value. Specifically,

Tolerance (or gap) = (upper bound-best feasible value)/ best feasible value.

Once this gam decreases to 0 as branching continues, the optimal value is obtained. However, sometime the feasible solution whose gap is greater than 0 can also be optimal.

In the branch and bound, the branching is done on the relaxed main problem. Generally lp-relaxed version of main problem is branched in a smart way, and without tracing all feasible region, an optimal solution can be obtained.

The linear supply chain systems are the main concern of this thesis, however. However some SCM models are non-linear and in some part, this thesis interested with non-linear models. The dynamic behaviors in the supply chain systems are modeled by using non-linear models. The model is generally discretized in the time domain to deal with the non-linearity. The supply chain models in the context of this thesis show dynamic behavior. Therefore at first, all the flows (information-financial-material) are modeled by using non-linear differential equations. Then to deal with this non-linearity, all the differential equations are converted into difference equations. By this way a linear model is obtained. The non-linear models in the scope of this thesis are hybrid model. In other words, they include both continuous and discrete behaviors at the same time. They are classified as differential algebraic equations. The problem classes that are represented by differential algebraic equations are called as an optimal control problem.

Besides bi-linear terms are also in the context of this thesis. A function f(x,y) is called bi-linear if it reduces to a linear one by fixing the vector x or y to a particular value. In general, a bilinear function can be represented as follows: $f(x,y) = a^T x + x^T Q y + b^T y$ where $a, x \in R^n$, $b, y \in R^m$ and Q is a matrix of dimension $n \times m$. As seen, the bilinear functions compose a subclass of quadratic functions. The optimization problems with bilinear objective or constraints are referred as bilinear problems, and they can be viewed as a subclass of quadratic programming. Many 0-1-integer programs can be formulated as bilinear problems. In addition piecewise linear and fixed charge network flow problems are very common applications of bilinear functions in the supply chain management.

Because bilinear programming is a kind of piecewise linear concave minimization problem, any solution algorithm that solves the later can solve the former. For the solution of those kinds of problems, cutting plane algorithms can be developed and used. However different from general problems, bilinear programming problems have a symmetric structure and for that reason more efficient cuts can be developed.

In this thesis, we focused on the solution mechanisms of batch production or procurement constraints in the context of the supply chains. Generally integer or binary variables are used to model these kind of constraints during the conventional modeling.

However as known, integer variables makes the system NP-hard. Therefore during the corresponding chapter of this thesis, we have discussed the alternative ways of modeling and solution of problems with these constraints. Different modeling techniques are discussed and suitable solutions are applied to these models.

1.3 Application

There are many applicative studies in the context of the supply chain systems. The context of this thesis is the strategic and tactical supply chain management models and their solutions. Supply chain management by itself is a macro or meta-topic. In other words by itself (with out applications) the concept of supply chain management only works for the birds view of the management issues. Applications are the concepts that make supply chain management alive and applicable.

Applications of the supply chain systems can be evaluated in three levels: Strategic, tactical and operational. The goal of the strategic level supply chain management applications is to identify and evaluate resource acquisition options for sustaining the company's competitive position over the long term. By this way, the sensitive points for the long-term efficiency increase can be found as a result of the strategic level supply chain management applications. Strategic network optimization, including the number, location and size of warehousing, distribution center and facilities, strategic partnership with all of the stakeholders, product life cycle management, for new and existing products, where to make and make-buy decisions are result of the main applications of the strategic level supply chain management [22].

Although tactical and strategic applications are classified in different levels, in the real life applications they are classified as a same. Therefore instead of dividing planning span into 3, it is decided that, there are two levels of planning: short term, long or mid term planning. In other words all of the application of the strategic level supply chain management can also be classified as an application of tactical level supply chain management.

Even if there are many applications of strategic and tactical level supply chain management, most of the real life applications exist on the operational level. The most common applications are the, production planning and scheduling, vehicle routing and scheduling, work force scheduling and real time planning [18]. Operational level refers to short-term decision problems facing supply chain system. The common two properties of the operational level applications are the existence of timing and sequencing decisions. In all of the applications listed above includes these two properties in their definitions. Operational level supply chain applications or the applications that require timing and sequencing can be modeled and solved by using linear programming, mixed integer programming or non-linear programming approaches. However, due to operational and highly complex nature of the problem, some heuristics and meta-heuristics are used to get a solution [23]. It can be said that modeling and solution systems for operational planning require more customization than those for tactical and strategic level supply chain application. In this kind of problems, optimization models are needed to support operational decision making throughout the company's supply chain. We have discriminated the operational level applications from the strategic and tactical level ones. The main parameter that determines this discrimination is the demand over the planning horizon. However this time interval is often known with large degree of certainty. In hear demands means requirements from downstream stages of the supply chain network.

Besides, another important characteristics of the operational level supply chain applications are that they are concerned with detailed execution of activities within single or a few segments of the supply chain rather than the entire supply chain.

Out of these operational level applications in thesis we have focused on the vehicle routing and scheduling part of problems. Routing refers to the physical paths and sequence of stops visited by the vehicles. The path that gives the minimum cost is the optimal assignment [24] [25]. In the vehicle routing problem both paths and the vehicle assignment to each path is a dynamic process. In other words, for each parameter set different assignment can be occurred. During the creation of paths, the capacities of the vehicles are very important for the assignment. Besides capacity itself (as a number), the interpretation of the capacity value is also very important. In this thesis, instead of using a linear capacity value, combinational capacity is used as a capacity of vehicle. By this way, the capacity is not represented by single number, instead all possible combinations of the products carried. In this thesis, we focused on the modeling and solution of this kind of VRP problems. This kind of VRP is more complex than conventional VRP, because of the interpretation of the capacity. This interpretation makes the capacity combinatorial. Each capacity combination should be represented by integer variables. The VRP with continuous capacity is NP-Hard problem and computationally very complex and with the addition of the discrete capacity constraints, the problem becomes more complex. For that reason in this thesis a new way of modeling combinational capacity constraints is developed. This re-modeling decreases the computational complexity of the problem, without disturbing the optimality of the result.

1.4 Contributions

This thesis made contribution to the field in 3 branch of the supply chain management: model, solution and application. First of all a new modeling technique is developed to model the complex supply chain networks. One of the most important rules of the modeling and software development is the model data independence [26]. This means that during the model development, the data should be separated from main model. However in conventional studies and modeling practices, the system topology is considered as a part of model and it is embedded into model. However it is known that topology is data or parameter and it is given rather than derived. Also if studied in detail during the modeling most of the equations blocks are used to model the topological constraints. In this part of thesis a new modeling technique is developed to model supply chain networks. With this new modeling technique model maintenance and remodeling becomes easier. In addition, because of this new modular model structure, the GUI integration becomes easier. The model becomes leaner and the model only includes model entities rather than data. This makes model more understandable and pure. Because topology is considered and behaved like data, during the modeling of supply chain network with different topologies, there will be no need to change the model. Rather than, the data set is updated and with this update, the new topology is modeled.

In the solution part, our main study focuses on the modeling of batch production constraints in the context of the supply chain management and network optimization problems. The batches or the use of batches is very common on the supply chain management literature and application. Most of the products are sold and produced by batches and most of the raw materials or semi-finished products are procured in batches. In the conventional supply chain modeling this batches are modeled by using integer or binary variables. However this modeling practice makes the model computationally difficult. In this part of the thesis, we basically focus on the modeling of batch production constraints without using integer variables.

First of all batches it self are modeled without using integer variables. The batches or the batch behavior are modeling by using bi-level linear model. To solve this model, a method inspired from low of strong duality is developed. During the development of this method, a linear programing method is developed to model modular arithmetic. After the batches it-self modeled with LP, then this new model is used into supply chain context. However, this time 3 level system is obtained. Because of the static structure of the duality, the dynamic behavior of the supply chain cannot be caught and the method developed and used during the solution of the 2-level system cannot be applied.

After this trial the MIP model that is used to model supply chain management model with batch production is converted into bi-linear model. The binary variables in the supply chain can be re-modeled by using bi-linear models. However bi-linear systems are non-convex, therefore to solve them exactly, branch and bound based algorithms are required. The branch and bound is also NP-Hard algorithm therefore passing from MIP domain to bi-linear domain does not bring any benefit in terms of computational difficulty. Therefore another method should be applied to this bi-linear system. First of all an underestimator scheme is applied to the system. However because underestimator is a kind of approximation scheme, therefore it is highly depends on the system or model parameters. First of all McCormic underestimator is applied to the model [27][28]. However to successfully apply this technique to the system, the bounds of the variables that form the bi-linear term should be known. For our case, because we do not know these bounds, this approximation algorithm does not give the satisfied results.

After underestimator, to solve the bi-linear model a genetic algorithm based hybrid heuristic algorithm is developed [29]. These heuristic considers the feasible region of the problems independently and parallel. The tests are done between commercial non-linear solvers, Cplex and the heuristics developed [30]. After the results, it is seen that the algorithm is working correctly.

In the application part, a unique Vehicle Routing Problem, VRP with combinational capacity constraints are developed and studied [25]. In the standard VRP models, the capacities are either represented continuously or by using their geometric shape. In this chapter a unique representation is used, combinational capacity limits. In standard way, the combinations are modeled by using integer variables. However it is known that VRP problems are highly complex problems and they are one of the most computationally difficult problems in the literature and with the addition of new integer variables, the model can be computationally intractable. Therefore in this chapter, for this unique problem, we have developed an algorithm that converts combinatorial capacity constraint to the linear or continuous ones. By using this algorithm we can decrease the number of integer constraints for sure and for the some problems the problem can be modeled and solved by using 0 integer variables without disturbing optimality. After that an approximation scheme is developed to make direct shipment without routing to decrease the run time. Even if the developed algorithm is an approximation algorithm, empirically it gives high quality results. The results of the empirical studies are shown in the corresponding chapter.

1.5 Outline

In the next chapters of the thesis the following topics are studied. In the Chapter 2, the supply chain network and the batch production constraints are considered. The methods are developed to model and solve the supply chain network optimization problems without using integer variables to model the batch behavior. In the Chapter 3, a new modeling technique is developed for the supply chain network problems. In this technique, all of the data especially network topology is isolated from model to make it leaner. In the Chapter 4, an original Vehicle Routing problem, VRP with combinational capacity constraints is studied. New algorithms are developed to solve this unique problem in this chapter.

Chapter 2:

MODELING AND SOLUTION TECHNIQUES TO DEAL WITH INTEGER BATCH PRODUCTION VARIABLES IN MULTI-ECHELON SUPPLY CHAIN SYSTEMS

Introduction:

Integer variables are used frequently in the supply chain modeling. They are used mostly to model fixed cost, quantity discount, opening and closing decisions. Even if, usage of integer variables indispensible, they can make the model intractable [20]. Therefore, there is a trade off between using and not using them. In the supply chain network modeling, they are generally declared as continuous variables despite they take integer values in real life [31]. Because the network modeled in the supply chain systems is complicated the models available in literature becomes tractable by treating integer variables as continuous at the expense of unrealistic models.

In this thesis, the integer variables to represent batch production in the supply chain systems are chosen as a point of focus. The methods and models to deal with the existence of these variables are studied. New methods are developed. The weakness and strength of these methods are discussed in detail.

2.1 The Batch Production Constraints:

One of the most important constraints that should be considered during the SCS modeling is the batch production constraint. Some of the products are produced in batches; therefore, the production size cannot be any real number. During the raw material procurement, raw materials can be bought in fixed quantities known as lot sizes. However, if you do not have an enough space in your warehouse, to hold raw materials, independent from exact amount of raw materials bought, the cost that should be paid will show step-wise behavior (Fig. 2.1).

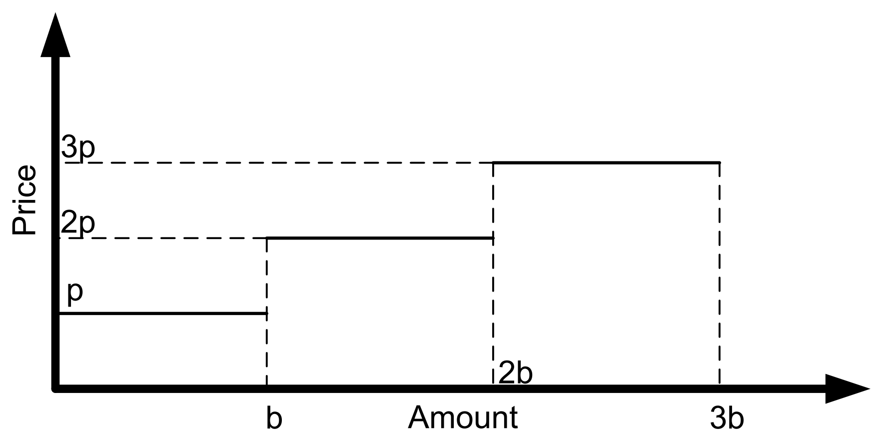


Figure 2.1: Price-quantity graph for batch procurement where b is batch size- p is price of one batch.

As presented in Fig. 2.1, independent from how many items bought, the customers have to pay the multiples of p. The pattern in Fig. 2.1, can be seen during production, procurement or quantity discount scheme. In the literature, this kind of behavior is modeled by using integer variables [32]. In this thesis, new modeling and solution approaches are studied to eliminate the use of the integer variables to during the modeling of batch behaviors in the supply chain system optimization models.

The batches can be seen in every part of the supply chain network. The raw materials are bought by batches, they are stored in batches. The production of the some products are made by batches and they are sold by batches. In this kind of setup, if the price of one batch is *c* then it cannot be said that, price of half batch is *c/2*. Instead it is known that whether 1 product or 1 batch of product is bought, the price is always *c* or the multiple of the c. In this thesis, we accept this rule, applied all algorithmic approaches with respect to this acceptance.

We have studied the behavior of multi-echelon supply chain systems with batch behaviors. These batch behaviors are conventionally modeled by integer variables. However we try to eliminate these integer variables during finding optimal solution for the supply chain optimization problem. To model the behavior demonstrated in the Fig. 2.1, for each segment we have to use an integer variables, mostly binary variable to show which segment is active. At the end mixed integer linear programming model is obtained and to solve this model a specific application of Branch&Bound (BB) should be applied [32]. However as known, as a solution algorithm, BB is an NP-Hard algorithm. Therefore, if this integer variables can be converted to linear ones, the algorithm required to solve problem is changed from B&B to simplex and practical computational complexity is decreased from NP-Hard to P (polynomial).

In this chapter, we basically focused on the step-wise functions that are used to represent batch production or batch procurement. Main aim of us is to find a method, modeling technique, or algorithm to eliminate the usage of the integer or binary variables.

2.2. The Graphical Derivation Method:

The behavior in Fig 2.1, can be remodeled by applying graphical subtraction as shown in Fig. 2.2. If second graph is subtracted from the first graph, the third graph is obtained. However as seen in Fig. 2.2, the third or result graph is the graph of a step-wise function. Therefore we can get the step-wise function by using graphical algebra. If we can represent the elements of this graphical subtraction, then we would represent piece-wise function in different way.

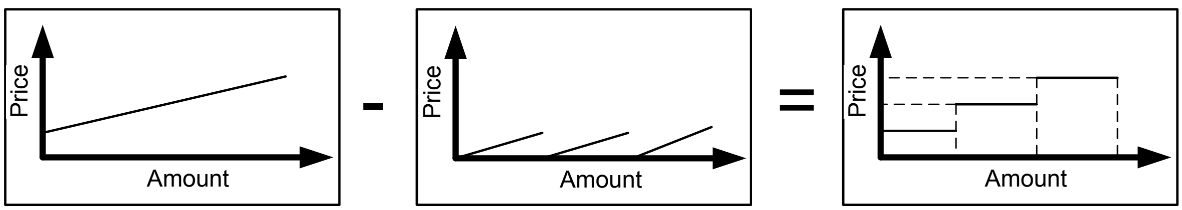


Figure 2.2: Graphical derivation of step function.

The representation of the first graph is easy to represent because it is a traditional linear function. However the representation of second function (saw function) is not easy to represent (Fig 2.3).

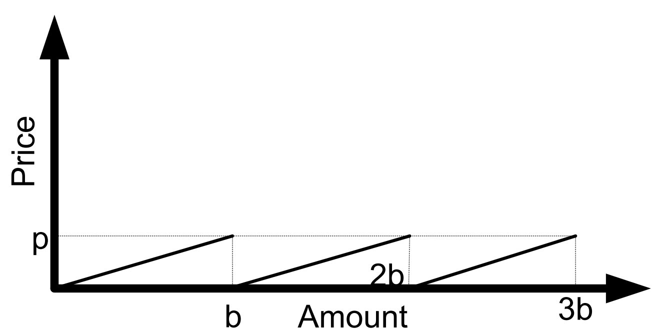


Figure 2.3: Saw function.

Only one segment of the function in Fig. 2.3 can be represented by the Eq .2.1.

$$y = \frac{p}{b} x \qquad (2.1)$$

where *p/b* is the slope of the one segment. However if the multiple segments are tried to be modeled, Eq. 2.2 should be used.

$$y = \frac{p}{b} x \ (\text{mod b}) \qquad (2.2)$$

Therefore if the modula operator is modeled as a linear programming problem (LP), then the step function can be modeled as a linear problem rather than a combinatorial problem with integer variables.

To model modula operator in a single linear model, the following transformations should be applied. Because during procurement, or production there exist an upper value, in other words, infinite amount of raw material cannot be procured or infinite amount of product cannot be produced.

Let assume that maximum 4b amount of material can be procured, and price of one batch (*b*) is equal to *p*. Therefore, cost of the procurement can be *p, 2p, 3p* or *4p*.

The result of the LP problem in Eq. 2.3 gives the value of the saw function in Fig. 2.3. As seen in Fig. 2.3, the *x* axis goes to infinity. However the *y* axis (the value or result of the function) is limited by *p*. The *x* value in the mathematical model is equal to the value of *x* axis in the Fig. 2.3 and the value in the *y* axis for the corresponding *x* value is equal to value of *K* in Eq.2.3. c_i, r_i *and* y_i's are helper general continuous variables. The *M* in the model (Eq. 2.3) is big number whose value is greater than *max* procurement or production amount and ε is a small number whose value is less than minimum production amount.

For a certain value of the *x*, the corresponding *y* value in the Fig. 2.3 is equal to value of *K* that is returned by the model Eq. 2.3. The s_i values can take some specific values. For the case in Eq. 2.3 these values are the multiples of the *b*. The value of the *b* is equal to length of one segment of the saw function in Fig. 2.3. In the r_i, the values of slacks are hold after multiple of *b*'s are subtracted from corresponding *x* value. Then with respect r_i values the positive y_i values are obtained. The reason of usage of two kinds of y_i values is to obtain absolute value of r_i. The model represented in Eq. 2.3 is a bi-level system. In other words there are nested models that are inner and outer model. In the outer model there is only one constraint and the purpose of that model is the selection of the minimum of the positives one of y_i's. Indeed the purpose of the bilinear model in Eq. 2.3 is obtaining minimum of positive y_i's which is equal to value of *y* axis in Fig. 2.3 for the corresponding *x* value.

The system modeled in Eq. 2.3 is classified as a bi-level system. In other words it can ben classified as a hierarchical model. Because there are two different optimization problems. The inner optimization problem is considered as a constraint for the outer optimization problem. Bi-level programs were initially considered by Bracken and McGill [34][35][36]. At that time bi-level programs is called as mathematical programs with optimization problems in the constraints, which exactly reflects the situation in our case. Being generally non-convex and non-differentiable, bi-level programs are computationally hard. Jeroslow (1985) shows that even the simplest instance of linear bi-level system is NP-hard [37].

$$
\begin{aligned}
&\max \ K \\
&st \\
&\quad \min \ \sum_i c_i \\
&\quad s.t. \\
&\quad r_i = x - s_i \\
&\quad c_i = y_i^+ + M.y_i^- \\
&\quad r_i = y_i^+ - y_i^- \\
&\quad y_i^+ \geq \varepsilon \\
&K \leq c_i \\
&where\ s_i \in S,\ S = \{0, b,\ 2b,\ 3b,\ 4b\} \\
&y_i^+, y_i^- \geq 0
\end{aligned}
\tag{2.3}
$$

Most of the MIP problems can be formulated as a bi-level system. As an illustration a simple program with binary variable can be formulated with bi-level program. For instance Eq. 2.4 is a linear system with binary variable.

$$
\begin{aligned}
&\max cx + eu \\
&s.t. \\
&\quad Ax + Eu \leq b \\
&\quad x \geq 0, \text{ u binary valued}, \\
&\quad \text{where c, e, A, E, b } \in \text{ R}
\end{aligned}
\tag{2.4}
$$

The binary requirement can be re-written in the following way:

$0 = \min\{u, 1 - u\}$

After this conversion the linear integer system can be converted into following bi-level system.

$$
\begin{aligned}
&\max cx + eu \\
&s.t. \\
&\quad Ax + Eu \leq b \\
&\quad x \geq 0 \\
&\quad y = 0 \\
&\qquad \max \sum y_i \\
&\qquad \text{s.t.} \\
&\qquad\quad \text{y} \leq \text{u} \\
&\qquad\quad \text{w} \leq 1\text{-u} \\
&\text{where y} \in \text{R}
\end{aligned}
\tag{2.5}
$$

As seen the binary nature of the system is reflected by using bi-level system. The main reason of usage of bi-level system in Eq. 2.3 is the same, not to use integer variables. The behavior represented in Fig. 2.3 can be modeled by using integer variables. However, our main aim in this thesis is the finding out new methods to prevent the usage of integer variables. For that reason bi-level programming is used instead of integer programming.

The system modeled in Eq. 2.3, is classified as a bi-level system and in this thesis new methods will be developed to solve this kind of systems. To solve it, we use the theory of the strong duality. With respect to strong duality, if primal problem has an optimal solution then the dual problem has a feasible solution and the optimal values of both objectives are equal to each other. The main reason of use of duality is to decrease the level of the system. According to duality, the only feasible solution for the dual problem are those that satisfy the condition for optimality for the primal problem.

The dual problem may be viewed as restatement in linear programming terms of the goal of the simplex method, namely, to reach a solution for the primal problem that satisfies the optimality test. Therefore by using the complementarity condition, the dual and primal system can be used to find the optimal solution.

As stated above, the rule of strong duality is used to make the bi-level system single level. According to strong duality for a solution, if both primal and dual system is feasible, then this solution is optimal. By using this, the inner level optimization problem can be converted into set of equations. We have made this conversion because if an optimization problem is converted into set of equations, it loses its level information. The prevented algorithm is demonstrated in Fig. 2.4.

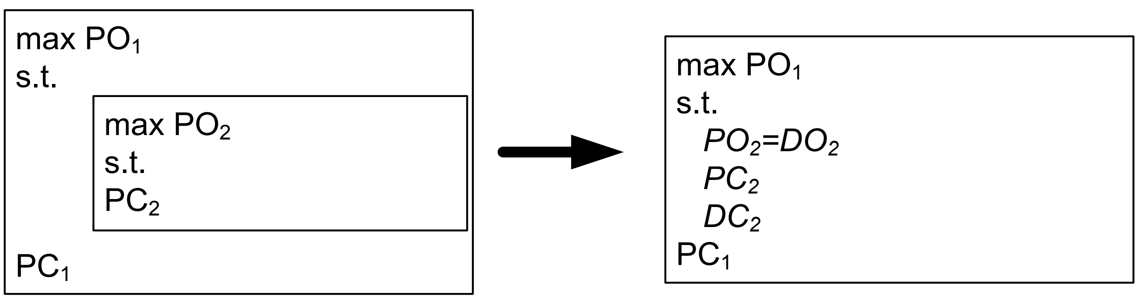


Figure 2.4: The algorithm of converting bi-level system to single level system. PO_1 is the primal objective of the 1st level equations, PO_2 is primal objective of the 2nd level system, PC_1 is the constraint of the 1st level primal system, PC_2 is the constraint of the 2nd level system, DO2 dual objective of the 2nd level system, and DC_2 is the dual constraint of the 2nd level system.

According to algorithm in Fig. 2.4, to decrease the level of the optimization system, we have used the theory of the strong duality. Dual of the 2nd level (inner) system is found, and then the constraints of the dual system and primal system are added to 1st level primal system as new constraints. Finally because primal optimal value is equal to dual optimal value, a constraint that shows this equality is added to system. With this algorithm level of system is decreased.

The model in Eq. 2.3 is developed to model the step function in in Fig.2.1. and the algorithm demonstrated in Fig.2.4 is developed to decrease the level of optimization model in Eq. 2.3. Therefore, this new method that uses rule of strong duality for now is valid only during the modeling of batch production or step function. However it is known that, the step-wise behavior is only valid in the context of supply chain management model. In other words, the step function is only a part of the system, the supply chain system. To make this method usable, the step function should be inserted into the supply chain management model, which is called as a master problem. However, addition of batch production or step function increases the level of the system. If the supply chain model with batch production and batch procurement model is modeled without used any integer variables, one more level is added to system in addition to the bi-level system that models the batch production. In the following section, this idea is demonstrated in a sample and simple supply chain network.

2.3. Optimization of Supply Chain Network:

In the previous section, the algorithm and model developed is used to model only the step function. However, the step function is only a part of the actual optimization model whose aim is to maximize profit in the SCS. In this part, a simple model (Fig. 2.5) is developed and the algorithm developed in the previous section is applied to this model.

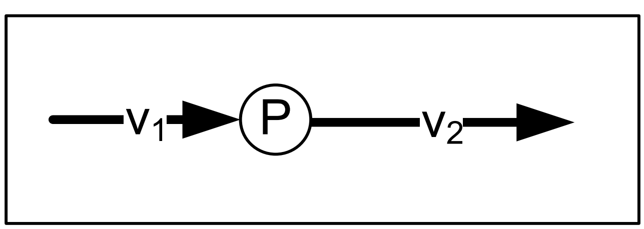


Figure 2.5: Sample supply chain network.

In the SCN presented Fig. 2.5, there is a one product. It is bought and sold without any transportation or operation. Let assume that product P is bought in the batches of 5 and sold one by one. Max procurement of P is 15 (3 batches), price of 1 batch is equal to 5 and selling price of 1 product is equal to 2 (Fig. 2.6).

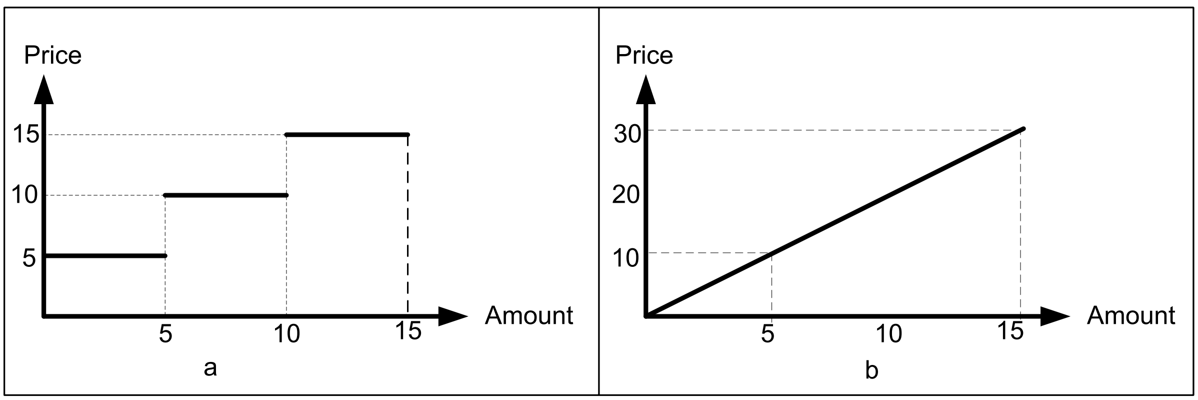


Figure 2.6: a.) Price-quantity relationship for procurement. b.) price-quantity relationship of sale.

With respect to this problem, presented in Fig. 2.5 and Fig. 2.6, the corresponding supply chain model is presented in Fig. 2.7:

Level 1

$$\max \sum_{t} 2v_2(t) - cost(t)$$

$$I(t) = I(t-1) + v_1(t) - v_2(t)$$

$$I(t) \leq I^U$$

$$cost(t) = 5 + v_1(t) - K(t)$$

Level 2

$$\max \ K(t)$$

$$st$$

Level 3

$$\min \sum_{i} c_i(t)$$

$$s.t.$$

$$r_i(t) = v_1(t) - s_i$$

$$c_i(t) = y_i^+(t) + My_i^-(t)$$

$$r_i(t) = y_i^+(t) - y_i^-(t)$$

$$y_i^+(t) \geq \varepsilon$$

$$K(t) \leq c_i(t)$$

$$where\ s_i \in S,\ S = \{5,\ 10,\ 15\}, t \in T$$

$$y_i^+, y_i^- \geq 0$$

$$v_1(t), v_2(t), I(t) \geq 0$$

Figure 2.7: SCN overall formulation where I(t) is the inventory of the product P at time t, $v_1(t)$ and $v_2(t)$ is the corresponding flows at time t, cost(t) is the cost of procurement at time t, I^U is the upper limit of the inventory. The remaining symbols are taken from Eq. 2.3, but only multiplied in the time (T) domain.

As seen in Fig. 2.7, the formulation includes 3 levels. However to solve it we have to decrease total level count to the 1. To do this, the modified version of algorithm in Fig. 2.4 is applied. The new algorithm is demonstrated in Fig. 2.8.

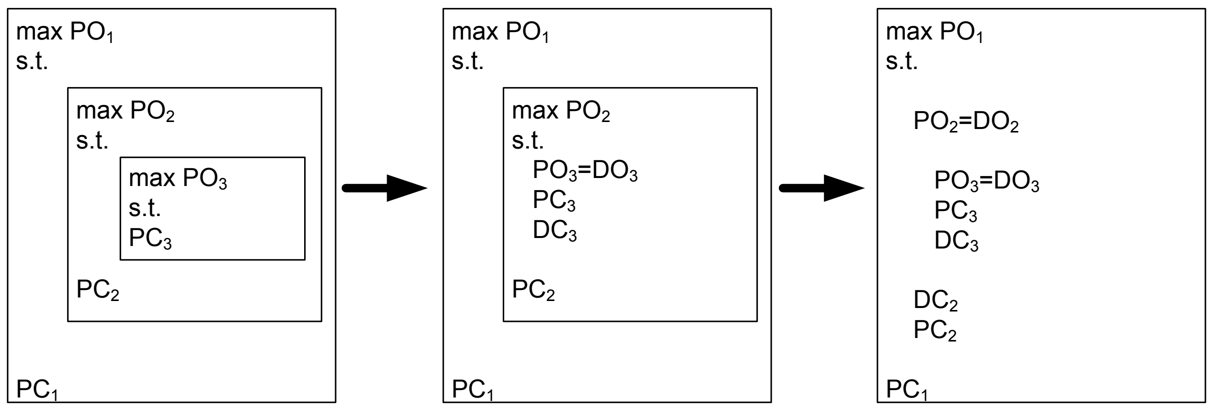


Figure 2.8: Level reduction algorithm where PO is primal objective, PC is primal constraint, DO dual objective, DC dual constraint and the indices shows the level of the equations.

As seen in Fig. 2.8, the similar of algorithm applied to the bi-level system, is applied to the supply chain network optimization model in Fig. 2.7. This algorithm is developed by also using the rule of the strong duality. The only difference is the level of the application. This time an optimization problem is converted into the set of equations two times instead of one time. This algorithm is applied to the simple network in Fig. 2.5 to see the benefit and defects of the system in detail. However it is seen that the method developed does not work correct for the 3 level system.

This method did not become a general solution technique for our problem. The reason of that will be discussed in detail. Even if the 3 levels of the method does not work correctly, 2 levels of system still works and by using this method modula arithmetic of a number can be calculated. However, the aim of solving batch production constraints in a polynomial time does not reach to result in the context of supply chain.

The main motivation behind the development of the 3 level system is to obtain an integer feasible solution without using integer variables. By this way all model only includes linear variables, and as a result, it is solved in polynomial time in practice. Therefore, large Supply Chain models can be created without considering the solution time of the model.

During the development of the 3 level system, the main idea is the use of rule of the strong duality. By using strong duality, the levels of the system can be decreased. In the model we developed we have used law of strong duality 2 times and decreased whole 3 level model into single level problem.

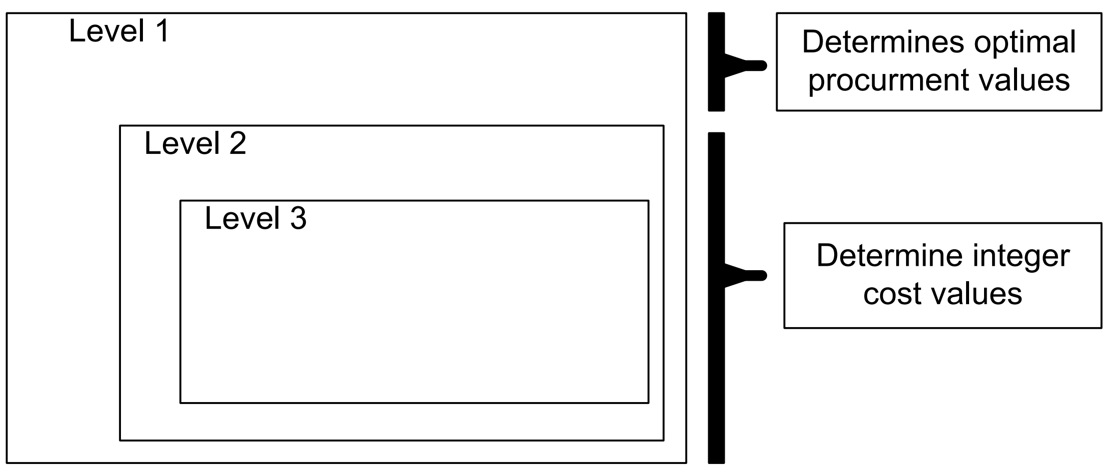


Figure 2.9. Meanings of levels.

As seen in Fig. 2.9, by using level decrease algorithm one times, the whole system is decreased into 2 level (bi-level) problem. The upper level model determines the optimal procurement value and the inner level model by using this optimal procurement value, determines the integer value corresponds it. However, the level 2 model is by itself an optimization model and it is created by using law of strong duality. It is known that, the dual problem and the primal problem intersect only in one point. In other words primal problem and the dual problem only have one feasible common points and this point is the optimal solution. In the 3 level model, it is seen that the optimal procurement values are only included in the level 1, and it is not included in level 2 and 3. Therefore, these 2 levels give integer solution blindly. Because the optimization of Supply Chain System is a continuous procedure, it is not allowed to pause the optimization process to determine integer value for optimal procurement value. Indeed, this action can be applied, but by doing this, another form of the branch and bound procedure is obtained and it is known that this kind of combinatorial algorithms for the optimization problems are NP-Complete. Therefore a solution for this problem is the inclusion of the procurement variable into the level 2 and 3. However this time infeasibility occurs again because of the low of strong duality. As discussed above, the level 2 and 3 finds its own optimal procurement and corresponding integer value, and if this value contradicts with the optimal procurement value of the level 1, infeasibility occurs. Therefore, use of the law of the strong duality in 3 level systems gives some infeasibility. However this will not shadow the success of the 2 level systems and use of duality.

In the 3 level system, the level 2 and 3 systems are used to obtain integer value for the optimal procurement value obtained by the level 1. This is succeeded the use of modular arithmetic and graphic interpretation of it. Therefore, 2 level system basically gets a number and returns its modula value by using an LP. During the modeling of the LP model of the modular arithmetic, the value whose mod value should be returned is inserted into system without addition of any optimization problem. Therefore because of this nature of the model, infeasibility does not occur.

The Eq. 2.3 basically returns value of the x mod (b). The value of the K is the modula value for the given x. The reason of success of 2 level system and failure of 3 level systems is the lack of information flow between a dual of a system and the dual of the dual of that system.

As a mid-result it seen that the method developed by using duality can be successfully applied to the modeling and solution of the step function or batch production. Also graphically it is proven that the step function is the graphical interpolation of the saw function and a linear function. However if it is studied in detail, it is seen that the saw function is the graphical representation of the modula arithmetic. Therefore if modular arithmetic is modeled by an LP, then the batch production can also be modeled. The method represented in Fig. 2.4 is valid for both batch production and modula arithmetic.

However problem occurred once this method is applied to the overall supply chain network model. To decrease the 3 level supply chain model into the single level problem, the algorithm in Fig. 2.4 is applied two times and this makes the dynamic nature of supply chain optimization, static. This causes infeasibility in the system. In the remaining part of this chapter, other methods developed to model supply chain network without integer variables are discussed in detail.

2.4. Bi-linear Conversion:

The aim of the previous section is the prevention of the use of the integer variables. Instead of integer variables we try to use linear variables. By this way the computational complexity of the model can be decreased. In this section our aim is also similar. New modeling techniques are used to prevent use of integer variables to model the batch production in the supply chain systems. In this part, the integer behavior of the system is modeled by using bi-linear system.

Because to decrease 2 levels by using strong duality gives inconsistent or wrong results, we have tried a new modeling technique to obtain integer solutions without using integer variables. We succeed this by using bi-linear terms in the objective function. For the batch production case, for each step function one bi-linear term is added into the objective of the optimization problem. If this problem is solved by using nonlinear model solvers, an integer feasible solution is obtained. However it is known that the solution times of the non-linear models is NP-Complete. The bilinear NLP model is shown in Eq.2.6

$$\begin{gathered} \max \; Revenue(x) - \textstyle\sum_i c_i z_i \\ r_i = x - s_i \\ c_i = y_i^+ + M.y_i^- \\ r_i = y_i^+ - y_i^- \\ z_i \leq 1 \\ \textstyle\sum_i c_i = 1 \\ remaining\ network\ constraints \\ where\ s_i \in S \end{gathered} \tag{2.6}$$

S is the discrete steps in the cost function for batch production. Because of the nature of objective of the Eq. 2.6 one of z_i will equal to 1 and others will be 0. Therefore the step-wise batch production function is modeled here without using any integer variables. Instead of this, bi-linear objective function is obtained. This model can be solved with NLP solver but, it is proven already that optimization of the bi-linear model is an NP-hard process.

By modeling the supply chain system with bi-linear terms instead of integer variables, we changed our domain from linear to non-linear. The change of the domain should not be done for nothing. In other words, changing domain should be beneficial either in terms of modeling complexity or computational complexity. We are sure that modeling with non-linear or indeed bi-linear terms is not beneficial in terms of the modeling complexity, because the models are nearly same. Therefore to show the benefit of changing of domain, we have to test and compare it. However to test the system we have to create a new network other than the one in Fig. 2.5. because the model in Fig. 2.5 is very simple to make comparison between models and methods.

The new system for the comparison is modeled in Eq. 2.7. The topology in Eq. 2.7. is suitable for multi-echelon multi-product supply chain network. In the system there are multiple layers and multiple products. Also the system shows dynamic behaviors. This means that in the time domain the inventory and order accumulation is changing with respect to a rule. The system in Eq. 2.7. has integer variables for the representation of the batch production. The aim of this part is determining performance changes after converting integer batch production constraints to the bi-linear terms. To make conversion we will use the method represented in the Eq. 2.6. instead of using one integer variable for the one segment of the batch, we will use one bi-linear term. The details of the supply chain model are discussed in the following part.

2.4.1 Supply Chain Model Description:

The test supply chain is a derivative of the network illustrated in the Fig. 2.10.

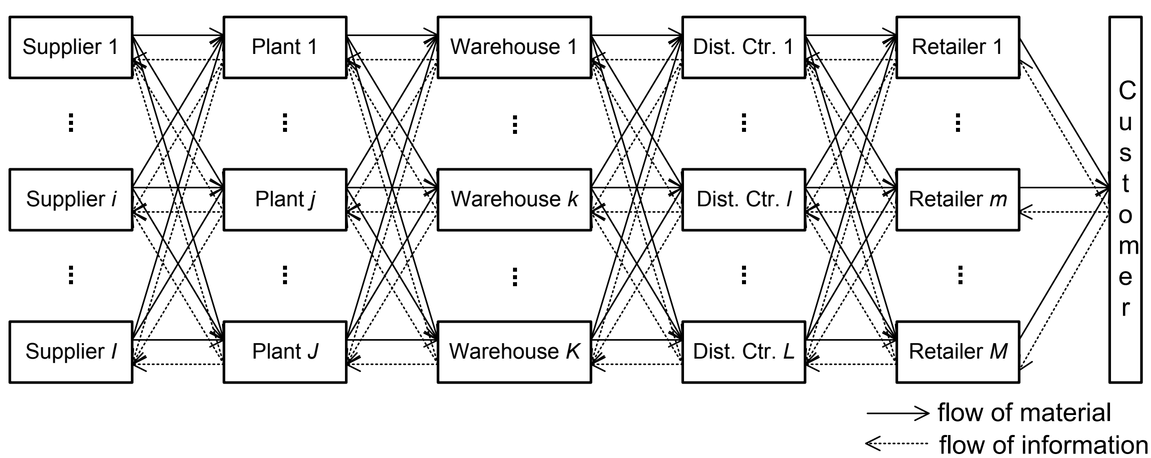


Figure 2.10: Multi-echelon, multi-product supply chain system.

In this system there are suppliers, plants, warehouses, distribution centers and retailers. The model is established on the order and inventory balance of the products. The basic of the model is given in the remaining part of this section. However the details and the more advanced part of this model can be found in the Chapter 3 of this thesis.

Inventory balance:

Final Product inventory at the warehouse:

$$I_{nkt} - I_{nk(t-1)} = \sum_{a} PR_{nkat} - \sum_{k''} yp_{nkk''t} \qquad \forall n \in N, \forall t \in T, \forall k \in N_{pw} \tag{2.7.1}$$

where k'' denotes a downstream nodes (distributor nodes).

Final Product at Distribution Center:

$$I_{nkt} - I_{nk(t-1)} = \sum_{k'} yp_{nk'k(t-\delta_{k'k})} - \sum_{k''} yd_{nkk''t} \qquad \forall n \in N, \forall t \in T, \forall k \in N_{ds} \tag{2.7.2}$$

where k' denotes upstream nodes (production facility nodes) k'' denotes downstream nodes (distributor nodes).

Final Product at Retailer:

$$I_{nkt} - I_{nk(t-1)} = \sum_{k'} yd_{nk'k(t-\delta_{k'k})} - \sum_{k''} yr_{nkk''t} \qquad \forall n \in N, \forall t \in T, \forall k \in N_{rt} \tag{2.7.3}$$

The final product inventory at the retailer k is a function of previous inventory level ($I_{nk(t-1)}$) and material transferred from distribution centers ($yd_{nk'k(t-dk'k)}$) with a time delay ($d_{k'k}$) and sales to customers ($yr_{nkk''t}$).

Order balance:

Order Balance at Plant Warehouse:

$$O_{nkt} - O_{nk(t-1)} = \sum_{k''} up_{nkk''t} - \sum_{k''} yp_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{pw} \tag{2.7.4}$$

Order Balance at Distribution Center:

$$O_{nkt} - O_{nk(t-1)} = ud_{nkk''t} - yd_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{dc} \tag{2.7.5}$$

Order Balance at Retailer:

$$O_{nkt} - O_{nk(t-1)} = \sum_{k''} ur_{nkk''t} - \sum_{k''} yr_{nkk''t} \quad \forall n \in N, \forall t \in T_f, \forall k \in N_{rt} \tag{2.7.6}$$

Batch representation:

$$\sum_{k''} yr_{nkk''t} = \text{Bs Bc} \quad \forall n \in N, \forall t \in T_f, \forall k \in N \tag{2.7.7}$$

where Bs is the size of one batch and Bc is an integer variable whose upper bound is equal to maximum batch count that can be sold. For our case it equals 500.

Objective function:

$Z = C_{RE} - C_{HO} - C_{TR} - C_{RM} - C_{PF} - C_{PV}$

$$C_{RE} = \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{rt}} \sum_{k'' \in N_{cs}} Prc_{nt} yr_{nkk''t} \tag{2.7.8}$$

$$C_{HO} = \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{pw}} I_{nkt} HC_{nkt} + \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{dc}} I_{nkt} HC_{nkt} + \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{rt}} I_{nkt} HC_{nkt} \tag{2.7.9}$$

$$C_{TR} = \sum_{t \in T} \sum_{n \in N} \left(\sum_{k \in N_{pw}} \sum_{k'' \in N_{dc}} yp_{nkk''t} TC_{nkk''t} + \sum_{k \in N_{dc}} \sum_{k'' \in N_{rt}} yd_{nkk''t} TC_{nkk''t} + \sum_{k \in N_{rt}} \sum_{k'' \in N_{cs}} yr_{nkk''t} TC_{nkk''t} \right)$$

(2.7.10)

$$C_{RM} = \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{pr}} \sum_{s \in N_{sp}} \sum_{a \in A} PR_{nkat} Req_{ns} RC_{ks} \qquad (2.7.11)$$

$$C_{PF} = \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{pr}} \sum_{a \in A} pr_{nkat} FC_{nk} \qquad (2.7.12)$$

$$C_{PV} = \sum_{t \in T} \sum_{n \in N} \sum_{k \in N_{pr}} \sum_{a \in A} PR_{nkat} VC_{nk} \qquad (2.7.13)$$

The nomenclature and the more detailed version of this model can be found in the Chapter 3.

2.4.2. Solver Comparisons:

As known that the bi-linear systems are non-convex, therefore global optimization solvers are needed to solve the system. For the integer case however, Cplex will be used to optimize the system. Comparisons will be made for the different solvers. To show the effect of the conversion from integer to bi-linear, 500 segments of batches are used. To decrease the effect of the main problem on the run time, 10 time intervals are used. We made comparisons for the 3 solvers: Cplex(for integer case), Baron and Dicopt (for the bi-linear case).

	Cplex	Baron	Dicopt
Optimal Result	56564	56564	54224
Time (sec)	21	65	18

Table 2.1: Bi-linear and integer comparison.

According to results in Table 2.1, both the Cplex and Baron give the same optimal result however, Dicopt gives the suboptimal result [38][39]. The main reason of this is the non-convexity of the bi-linear problem. Because Dicopt is not an exact solver, the result obtained from Dicopt can be suboptimal. The results obtained from Cplex and Baron are same. However, the runtimes of the Baron and Cplex are very different. As seen in Table 2.1, run time of the Baron is higher than Cplex. The reason of this is the optimal design characteristics of the Cplex on the Branch and Bound. As known, Baron is also using B&B to solves the problems. However, Cplex and its algorithms are much more effective than Baron. Other reason is the generality of Baron. Baron is more general solver than Cplex. It can be used more general types of the problems.

These results show that passing from integer domain to bi-linear domain does not bring computational effectiveness. However we made this comparison by looking at the runtimes. Also we are comparing Baron and Cplex which the former one is one of the most general solver and the later one is the one of the most professional and specialized solver. We try to compare the number of iterations, however the operations made in each iterations for each solver are totally different. For that reason comparison of the number of iterations is not logical. In the next part an underestimator scheme is used to solve bi-linear system

2.5. Underestimators for Bi-Linear System:

Many non-linear optimization models include bi-linear terms. By applying underestimator constraints in to the original system a tight convex relaxation of it can be obtained. One of the best-known method for finding underestimator is proposed by McCormic [40]. According to this method each bi-linear term is replaced by its relax constraints. McCormick used the following bi-linear optimization model to show his approach.

$$\begin{gathered} \min \ z \\ z - x + w = 1 \qquad (2.8) \\ w = xy \\ (x^L, y^L, x^L, w^L) \le (x \ , y \ , x \ , w \ ) \le (x^U, y^U, x^U, w^U) \end{gathered}$$

For this standard form formulation, to obtain the convex relaxation, each bi-linear terms are replaced by it McCormick convex relaxation:

$$\begin{gathered} w \ge x^L y + y^L x - x^L y^L \\ w \ge x^U y + y^U x - x^U y^U \\ w \le x^U y + y^L x - x^U y^L \qquad (2.9) \\ w \le x^L y + y^U x - x^L y^U \end{gathered}$$

As seen in Eq.2.9, the relaxation is totally depends on the upper and lower limits of the bi-linear variables. Also all of the bilinear terms can be replaced by a linear ones by using this method. For our model, there are only bi-linear terms in the objective, therefore the bi-linear terms are changed with a new variable (w), and the remaining constraints that are represented in the Eq. 2.9 is added into system. After this conversion, all the terms in the system become linear and the computational complexity of the system is changed from NP-hard to polynomial.

However it is known that underestimators are an approximation method, therefore results cannot be good enough. In addition, the success of the McCormic underestimator is very tightly depends on the quality of the bounds of the variables that exist in the bi-linear term

For our model represented by Eq.2.6, the bounds of y variables can be determined tightly, however, for the z variable, the bound determination cannot be done tightly. This underestimator determination technique has been applied to model represented in Eq.2.6 however, promising results cannot be obtained. The main reason of this is the relationship between success of the method and bounds.

The quality of the solution obtained from this method is measured by looking at the two parameters: the cost's or revenue's cut point (to make the batches complete) and the objective functions' value. The results for the both parameters are not good enough. First of all the batches occurred after optimization are not on the end point. In other words, results are against to logic of the batches, because in the batches, the revenue or the cost becomes a multiple of the batch value. However the result obtained from McCormic underestimator, is not an integer multiple of the batch value. For that reason the optimal value obtained after optimization is not the optimal value.

The McCormic method is used to obtain an optimal solution for the bi-linear system by reducing it to linear system. However, this method is an approximation method and the success of it depends on the upper and lower bounds of the variables. If the bounds cannot be known in tight, the result obtained becomes low quality and that is case for our problem.

Because our bi-linear system is non-convex standard non-linear solvers do not guarantee the optimality of their results. As an exact non-linear non-convex solver, Baron is used to solve the problem. However even if the optimal result is obtained, the runtime becomes higher. The approximation techniques used to solve system, however, it gives low quality results. We passed from MIP domain to non-linear domain to provide computational benefit, however none of these techniques provide this benefit.

On the next part, a genetic algorithm based hybrid meta-heuristic is developed to solve the bi-linear system. The details of this heuristic and the results are discussed in the next part.

2.6 A Meta-heuristic for bi-linear non-convex systems:

After addition batch production constraints, the supply chain model becomes a bi-linear and non-convex. Therefore, exact optimization methods become incompetent to obtain optimal solution. This kind of optimization problem is classified as a global optimization problem [41]. Context of global optimization is very large, and usage of it is very common both in real life and literature. Global optimization is one the hottest topic of the last 2 decades. It refers to finding of the extreme points of a non-convex continuous function, on a given interval. Global optimization problem can be constrained or unconstrained.

In real world, if a system is wanted to model without too much approximation, system will certainly be continuous and non-convex. The most of the models used in the process system engineering or in bioinformatics are these kinds of systems. Therefore, OR specialists and computer scientists are spending big effort on the solution of continuous and non-convex systems.

However, the solution of this system is not easy, because of the non-convexities. The global optimization problems include many local extremes and most of the methods can be stagnated in one of these local optima.

Despite to complexities of the global optimization problems, there are some exact methods, such as Branch and Bound, Bayesian search, trajectory methods, complete enumeration methods, etc. Although these methods find the exact minimum or maximum of the optimization problem, they are NP hard algorithms. For the large solution space, finding the global optimum is nearly impossible.

Because of that, meta-heuristics are frequently used in the solution of the global optimization problems. The main processes in all meta-heuristics are the exploiting of the solution space by diversifying the initial solutions and exploration by intensifying the search in some promising areas.

Extensive research was done about the solution of global optimization problems in the literature. Most of methods are hybrid heuristic methods rather than standalone meta-heuristics method and most of them includes local search.

The main aim of usage specially designed local search operators is finding of an optimum very quickly. However, the found solution is mostly local one because of the non-convexities in the model. Therefore the local search operators are mainly used as intensifying agent in the hybrid heuristic method.

Yang, Xu and Chee developed a hybrid evolutionary programming algorithm [42]. According to most researchers, because Evolutionary Programming algorithms do not include crossover operator, it shows weakness on the exploiting of the solution space. Therefore in this algorithm, Macro Mutation and crossover operator are embedded in to evolutionary programming [EP] algorithm. To increase the speed of the convergence, they hybridize this EP with a local linear bisection search. They also use Simulated Annealing in the selection phase of the algorithm to prevent the chromosomes in the early generations from being trapped in the local optima.

In the Yang, Xu and Chee's method, crossover and mutation operator are used as a diversifying agent to explore the search space. Local search used to intensify the preliminary solutions.

Abdel and Masao used a hybrid of Tabu Search and a local search procedure [43]. They classify the meta-heuristics as a point-to-point heuristic and population based heuristics and they said that the main weakness of the Tabu Search, as a point-to point method, is the lack of diversifying scheme, but the main strength of method is the long and short term memory. In their method, to deal with the weak side of the algorithm and to use the strong sides, they invent new data structures.

First of all, they used Multi-Ranked Tabu List. Instead of holding just one characteristic of data, they use multidimensional data in their tabu list. In their application, in tabu list they hold function value and recency based information of current data at the same time by giving certain values to them. By this way, during the solution generation, we can use the information of these two kinds of data at the same time. To deal with the lack of diversifying scheme of the tabu search, they used a new data structures, Visited Region List, Tabu Region and Directed Tabu Region. With Visited Region List [VRL], the main aim is the elimination of revisit of highly visited solutions. Also with Tabu Region and Semi Tabu Region, the re-visit of the old regions can be prevented.

In this article again, Tabu Search is hybridized with 2 local search procedures. One of these procedures is Nelder-Mead Search and the other is Adaptive Pattern Search Strategy [44]. Both of these procedures are the derivative free optimization [DFO] tools. These DFO tools are both used to diversify and intensify the solution in the algorithm. Because these DFO tools make improvement by creating neighborhood and local trial solution points, by using these trial points, diversifying scheme of the algorithm can be met.

Kaisa, Marko, and Heiki developed a hybrid of the Simulated Annealing with a new local search strategy, Proximal Bundle Method [45]. They classify their hybrid as parallel synchronous, because the order of the hybrid algorithm remains same during the execution.

The most important development of their method is the use of the hybrid strategy. In a traditional way, to hybridize global and local search algorithm, the global method is followed by local method. This method is called sequential hybrid method. The main weakness of this strategy is the computational cost of the local search [LS] strategy. Decreasing the cost of LS causes to termination of the global phase very early and because this decreases diversification level of the solutions, the probability of finding global solution will decrease.

To deal with the foregoing problem, they embed one of the algorithms into other one. In their first hybrid, they embed Simulated Annealing into proximal bundle method and call this hybrid as *biased proximal bundle method*. The main weakness of this method is the lack of globality of the method. Because proximal bundle method is not versatile enough in the descent direction, the method cannot diversify the solution space. In the second hybrid they embed proximal bundle method into SA and this method gives better result because of the diversification and intensification phases becomes more efficient.

In their research, total 34 different hybrids are used and only 3 of them make improvement or give good solution. The others are failed to give acceptable solution.

Cehellapilla, applies an Evolutionary Programming strategy to the global optimization problem with different mutation operators [46]. They compare 4 different kinds of probability distribution functions on the mutation of the solutions. First they look at the advantages of using Cauchy mutation and Gaussian mutation. Then they derive different kinds of mutation operators by using these foregoing 2 pdfs (mean and adaptive mean mutation operators).

2.6.1. Background:

In this report, we will introduce a new way finding global optimum solution in non-linear non-convex problems. Our method is the hybrid of both point-to-point heuristic algorithms and population based heuristic algorithms. As a population based algorithm we have used Genetic Algorithm and as a point-to-point algorithm we have used simulated annealing. Also to increase the convergence speed we also embed local search strategy to the Genetic Algorithm.

The main improvement with our method is the introduction of the division of the solution space into small spaces. We call each of these small spaces as sub-spaces. For each sub-space we run a sub-heuristic. By this way instead of exploring the whole space by high number of iterations of heuristic methods, we try to iteratively explorer the sub-solution space of the problem (Figure-2.11). With this iterative approach, our main aim is the elimination of some sub-problems in the case of the long non-improving solution. We succeed this by introducing Manager Heuristic.

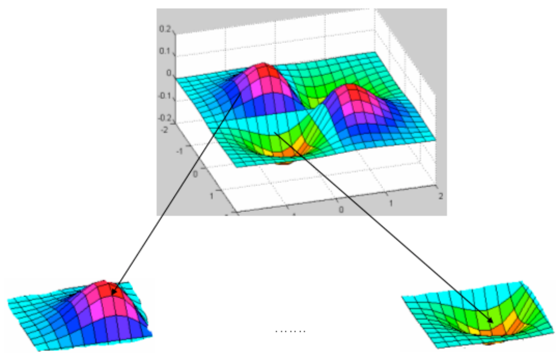


Figure-2.11: Division property of algorithm.

The manager heuristic is the extension of the Simulated Annealing. The main function of it, iteratively decreasing the acceptance probability of non-improving sub-solutions, as the execution of the algorithm continues. This is done by temperature reduction [47].

There are 2 way relationships between sub-heuristics and manager heuristics. After certain iteration of sub-heuristics in each sub-problem, they pause to execute algorithm and send all information it has to the manager heuristic. Manager Heuristic evaluates the all sub-problems and their execution trace and decides to continue with this sub-problem or kill it. With respect to this decision sub-heuristic stops execution or continue to execution from where it suspend.

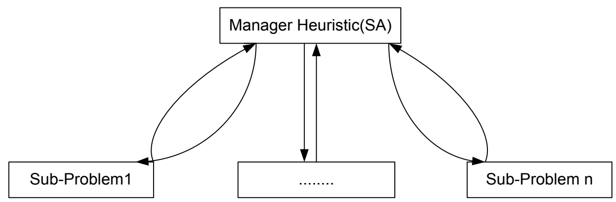


Figure 2.12: Sketch of algorithm.

Each sub-problem can be solved with any kind of method, but in our application we used single method for each sub-problem.

On the solution of sub-problems, because of its convenience for global optimization problems, we have used hybrid genetic algorithms with local search. The main function of local search is the increase of the convergence speed of the algorithm.

2.6.2. Algorithm of Sub-problems:

A global optimization problem can be written as follows:

$$\min \; f(x), x = (x_1, x_2, \ldots, x_n)$$

st

$$F = \left\{ x \in R^n \left| \begin{array}{l} g_j(x) \le 0, j = 1 \ldots l \\ h_k(x) = 0, k = 1 \ldots m \end{array} \right. \right\}$$

$$x_i^L \le x_i \le x_i^U, i = 1 \ldots n$$

In here, the objective function is the non-linear and non-convex objective function, the first constraint set is the equality and inequality constraints and the last constraint is the box constraints. The aim of this study is the development of the method that finds the global optimal point of this system.

Rest of this section gives detailed description of proposed algorithm. First the representation scheme and initial population creation is introduced. Then the variation operators are described with selection operator. Finally the repair scheme is discussed with its abilities and disabilities.

2.6.2.1. Procedure of Proposed Algorithm:

Algorithm includes two separate parts: manager algorithm and sub-problem algorithm. The sub problem algorithm works in the following way.

The process starts with the creation of the new feasible population. Actually this point is one of the important part and weakness of the proposed algorithm and will be discussed later. After that step, the variation operators are applied in a random and sequential way. The hybrid that we used is a sequential hybrid. Therefore once a method is applied, the other methods cannot be applied till first method ends its execution. Also the execution order of variation operator is determined in a random way. Each parent chromosome is assigned a random number uniformly distributed from 0 to 1. Depending on which range this number is located in i.e. $[0,p_L]$, $(p_L, p_L + p_C)$ or $[p_L + p_C, 1]$, local search or crossover or MM operator is applied to the parent to generate a new chromosome where p_L, and p_C are the probabilities of local search and crossover operators, respectively. After that a tournament selection is applied to form the new generation.

After the foregoing algorithm is applied a predetermined amount of time, it suspends execution and returns to manager heuristic. After that step manager heuristic decides to future execution scheme of the system. The detailed flowchart of the algorithm is shown in the Figure 2.13.

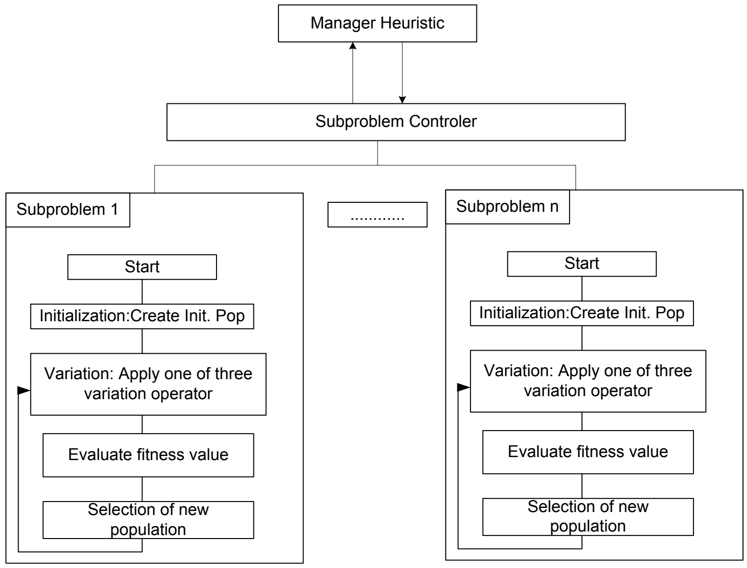


Figure 2.13: Steps of algorithm

2.6.2.2. Representation Structure:

In the most combinatorial optimization problems, a solution scheme is decoded in binary way. The main reason of this is the nature of the problem. As we know the combinatorial problems includes a countable number of solutions. For that reason the 0-1 representation of that kind of solutions is more efficient. However, for continuous systems, any encoding scheme does not work, because first of all the non-linear systems are not combinatorial or the number of solutions are not countable, therefore 0-1 representation is not convenient. Also if we make crossover to the 0-1 representation on the case of the nonlinear system, we may not continue the good properties of the parents. For that reason instead of using 0-1 representation we have to use the real value of the variables.

In our method the representation of the solution is in the format of the (x,s). In here the 'x' is the real value vector of the variables. 's' is the strategy variable. In our case it equals the range of the box-constraints. i.e. if $-1 \leq x_1 \leq 3$ then s_1=3-(-1)=4. This 's' value will be useful in the local search step of our method.

With this representation scheme at the beginning of the algorithm a set of initial population should be produced to start the algorithm. However although this operation seems to be an easy task, to succeed this for problems that has many constraints is very difficult. For this kind of problems, the feasible region includes disconnected many small sub-spaces. On that kind of situation finding an initial population that includes a solution in each feasible region is a very difficult process. Also if the initial population does not include solutions in every region of the feasible region, the quality of the solution may be bad, because initial population located at some specific point can give misleading results and the success of the algorithm cannot be evaluated objectively.

2.6.2.3. Variation Operators:

There are 3 kinds of variation operators applied to hybrid of genetic algorithm for the sub-problems. These are local search, mutation and crossover operators.

2.6.2.3.1. Local Search:

As a local search, Local Linear Bisection search (LBS) is applied in our method. Although, there are applicable many kind of local search procedures in the literature, the local linear bisection search has many advantages. First of all, we can apply this method to any kind of functions (suitable for differentiable and non-differentiable equations). Besides the convergence speed of this method is very good. In a couple of iteration a better solution can be found. The weakness of it, with this method probability of finding the best solution is less than other more precise methods. However, because we hybridize it with genetic algorithm whose diversification scheme is very powerful, till end of iterations, we do not need such a precise and exact local search method.

The working style of method can be seen in the Figure 2.13.

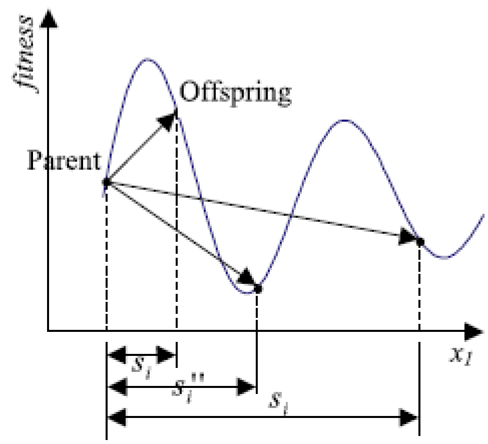


Figure 2.14: Logic of local search.

The algorithm is:

```
SingleLBS()
if (gradient>0)
        x'=x+s;
else
        x'=x-s
endif
```

```
LBS()
n=0;
While(f(x')<f(x))
        S=s/2^n;
        LBS();
        n++;
```

End while

In the above algorithm 'n' is the number of iteration in the LBS. As said before 's' is the range of the box constraint. According to algorithm, we have to divide s by 2 till getting better value.

However because most of the systems are non-differentiable, finding gradient is not possible. In that case, the numerical gradient is enough for LBS operator.

2.6.2.3.2. Mutation Operator (MM):

As said before the aim of the LBS is to intensify the solution. However the main aim of the MM is the diversification of the solution set. One of the most promising properties of the genetic algorithms is the power of diversification scheme of it because by this way system can escape from local optima.

After the MM operator is applied to a chromosome the respective change is the following:

$x' = (x_1 + U(x_1^L, x_1^U), x_2 + U(x_2^L, x_2^U) \ldots. X_n + U(x_n^L, x_n^U))$

Where, U is a uniform distribution. Therefore at the end of the mutation, x' will be uniformly distributed between

$$\begin{cases} x_i^L, x_i^U + x_i, if(x_i > 0) \\ x_i + x_i^L, x_i^U, if(x_i > 0) \end{cases}$$

By this way diversification scheme is succeeded.

2.6.2.3.3. Crossover:

Because in the representation, we use the exact numbers, the crossover operation becomes very easy. The two new offspring is created in the following way:

$$x_i^{1\prime} = \alpha_i x_i^1 + (1-\alpha_i)x_i^2, \quad x_i^{2\prime} = (1-\alpha_i)x_i^1 + \alpha_i x_i^2,$$

$$s_i^{1\prime} = \alpha_i s_i^1 + (1-\alpha_i)s_i^2, \quad s_i^{2\prime} = (1-\alpha_i)s_i^1 + \alpha_i s_i^2,$$

As seen above formula the new offspring is created by taking the convex combination of two parents. The (‘) prime means the new offspring. In formula α is uniform random number between [0-1].

2.6.3. Handling Constraints:

As seen from the variation operators the main effect of all of them is adding a ‘move’ value to the old solution.

x’=x+move.

But adding this ‘move’ value is made without caring the any feasibility constraints. Therefore at the end some infeasible solution can be seen.

Because the ‘move’ value is the main source of the infeasibility, by scaling up and down the move value we can get a feasible solution. The solution algorithm of the repairing scheme is the following:

```
scale=1;
x’=x+move*scale
While(x’ is not feasible)
        scale=scale*U(0,1);
        x’=x+move*scale
End while
```

The above iteration is made a certain amount of time. If a feasible solution cannot be get then this indicates that the initial point is located at the boundary of the feasible set. If that is the case, the direction of move is flipped and same algorithm is re-applied.

The main weakness of this repair scheme is the possible lose of the good characteristic of parent solutions. Because, as seen in the algorithm, the repair scheme breaks all the connection between original solution and new solution. Therefore to increase the convergence speed and solution quality of method new and more powerful repair scheme is required.

2.6.4. Selection:

A deterministic tournament selection is applied to select the new population. A q random offspring are selected from the population set. Then the best element from this q random population subset is selected. This process continues until total number of selected offspring reaches to pop-size.

2.6.5. Feedback Relation Between Sub-heuristics and Manager Heuristic:

The main function of sub-heuristic is the exploring the pre-defined sub-spaces of whole solution spaces. In other words sub-heuristic applies certain amount of hybrid of genetic algorithm. After that, it returns to manager heuristic and wait for the order of it. Therefore there are not any termination criteria for sub-heuristics. The decisions about the future of these sub-heuristics are given by manager heuristic.

In our method, manager heuristic is the main decision maker of whole method. Mainly, it is Simulated Annealing based decision maker scheme. In the algorithm the first of all the range size of sub-problems are decided. In our approach we mainly take '1' as a range of box constraint of each sub-problem. Therefore, for instance if our problem has two variables and box constraint of them are $-1<x<2$ then each sub-problem's solution spaces' range is one by one (a square), and total 3*3=9 sub-problems exist. For that reason after manger heuristic gives the run command, total 9 copy of hybrid genetic algorithm starts to run in their sub-region of the whole solution space. After a certain number of iteration they return to manger heuristic for evaluation. Manager heuristic as said before decides to whether the sub-problems continue to execution by using an extension of SA.

Also Manager Heuristic holds the best fitness value that each sub-heuristic finds till current iteration. If at the current return of a sub-problem to the manager heuristic [MH], it improves the best fitness value, then MH updates the best-fitness array and send continue to execution where you left signal to the subproblem. However, if subproblem returns a worse fitness value to the MH, then SA scheme starts to work. According to SA probability of accepting the bad solution and permit to subproblem to continue to execution is the follwing:

$$P(accept) = e^{-\frac{(b-c)}{t}}$$

In the above formula, b is value of the best fitness that corresponding sub-heuristic find till current time. 'c' is the current fitness value that sub-heuristic returns and t is temperature.

If α, the uniform random number between 0 and 1, less than P(accept) then the corresponding sub-heuristic continues to execution. However on the opposite condition, MH sends kill signal to the sub-problem and sub-problem stops execution.

The temperature update scheme is the following:

$t^{new}=t^{old}*a$ where $0.8<a<1$

As seen from formula, at the beginning of the iterations with a high probability all sub-heuristics continue to execution. However at the end iterations the sub-problem that returns worse fitness value than its best fitness value will be killed by MH.

2.6.6. Tests and parameter determination:

2.6.6.1. Parameter Determination:

In our method one of the most important parameter that should be determined is the initial temperature. Because if the initial temperature is high then none of the sub-heuristics will be killed in other words we could not kill non-improving solutions. On the opposite case, if the initial temperature is very low, then we may lose optimum solutions. Because we generally made 6-8 outer loop evaluation, and as a cooling scheme we use, 0.8t, one of the most promising initial temperatures becomes 70-80. However this temperature is not the sharp value for the problem. Because if we divide the whole solution space into many sub-spaces then we will not need to kill any sub-heuristic because the main aim of kill of sub-problems is stopping the non-improving solutions. By this way the total number of evolutions will be decreased. Therefore if we increase the number of sub-functions then we have to increase the temperature to decrease the probability of kill any of sub-heuristic. However increasing the number of sub-function increase both precision and run-time. Therefore for the foregoing case, we have to decrease both outer loop iteration (Iteration count of manager heuristic) and inner loop iteration (hybrid genetic algorithm iteration count.) otherwise cpu time and number of function evaluation will explode.

Another most important parameter is the number of sub-problems. However this value is totally determined by the user. Also decision should be given with respect to characteristic of the function.

Besides these macro parameters that are determined in the MH, there are many parameters that should be determined in the GA part of the sub-heuristics. The tests of these parameters are made in the [1] by Yang at all. For our case we select population size as a 20 instead of 50, because our solution space for every sub-problem is very small with respect to whole problem. Also we make tournament size q=5. Besides these parameters, we take same probability values from Yang's article. We select p_c and p_l as 0.2

2.6.7. Result of Metaheuristic:

It is known that, the success of the heuristic methods is very dependent to the parameter values that are chosen. This is also true for our case. Therefore after tuning parameter values, the following result table is obtained.

	t=1	t=2	t=5	t=10	t=50
Opt	100%	100%	%100	%80	%70
≤10%	0%	0%	%0	%10	%20
≥10%	0%	0%	%0	%10	%10

Table 2.2: Result of the heuristics for variable parameters and time steps.

To determine the quality of the results of the algorithm, we made comparison with Baron and the heuristic. Because Baron is an exact solver, every time it gives the optimal result. However, we cannot be sure about the solution quality of the heuristics. Therefore 2 types of comparisons are made. First of all, for 50 sec of run time we try to determine the quality of the result given by heuristics. Then for different set of data we will find the time required to find the optimal result by heuristic algorithm. We made these comparisons for the supply chain network in Eq. 2.7. The system that we make comparison has variable parameters values except step sizes in the batches. We use total 500 steps of batches in the system as made in the section of non-linear vs integer comparison. For the first type of comparison, we applied variable time step number from t=1 to t=50 and variable price and profit variables to take the average of the results, and for the second type of comparison we also use variable t. The result of the first comparison is shown in the Table 2.2.

To obtain the results in Table 2.2, for each t values total 10 runs are made with different parameters. The t value corresponds the number of time period that supply chain model is optimized. In all of the cases, the global optimization solver Baron, solves the model optimally. But for the case of heuristics algorithm, we have limit the run time by 50 sec. The aim of this comparison is to determine the quality of the solutions that are given by heuristic algorithm. We fixed the run time by 30 sec to force the algorithm and to determine the real power of it.

According to result in Table 2.2, for t=1, t=2 and t=5 the heuristic gives the optimal results for the 10 different runs with 10 different parameter set. However as the total time step is increased the quality of the solution is decreased. For t=10, out of 10 runs, 8 of them are optimal, one's result has less than 10% gap and remaining one's result has more than 10% gap. These results actually promising, because Baron finds the optimal value in 61 sec for t=10. However the heuristic finds it with 50 sec run time limit. Also it should be remembered again that, Baron and the heuristics algorithm is applied to the bi-linear system.

To determine the speed of the heuristic algorithm we have made runs with Baron, Cplex, and the heuristics. We applied these methods to the model in Eq. 2.7. The model in Eq. 2.7 is a multi product, multi echelon supply chain network. We made the runs with different time steps and we will report the time required to find the optimal result for all of the solvers. The result is reported in Table 2.3.

	t=1	T=2	T=5	T=10	T=50
Cplex	3	4	9	21	63
Baron	25	25	42	61	362
Heuristic	18	26	36	43	186

Table 2.3: Run time comparison till optimal solution is found

As seen in the Table 2.3, in all of the comparisons Cplex beats all. The main reason of this as said before the power of it. Because it is very optimized and powerful solver for MIP problems it always gives the best result. However the Cplex is applied to the supply chain model with integer variable to represent the batches. Baron and heuristic as pointed before are applied to the bi-linear system. Therefore the real power of the heuristic can be seen by comparing heuristic with the Baron. According to Table 2.3, in all cases heuristic reaches to optimal result earlier than Baron.

2.7. Summary:

In this part, we are interested in the solution strategies in supply chain management models. The model that we are focused on is basically multi-echelon, multi-product supply chain network with batch production and procurement constraints. First of all we try to solve the system without using integer variables to represent the batch production and procurement behaviors. First of all, by using the low of strong duality we modeled 2 level and 3 level systems without integer variables. We solved the 2 level system. As a result of it, we obtained linear time solution of modular arithmetic and result of the step-wise system used in the representation of the batch production. However to get full scale supply chain model we have to superimpose this two level system into the main supply chain model which we called macro model. After this superimposition we obtained a 3 level system without integer variables, but this model still behaves in the same way as the MIP model behaves. To solve 3 level system we have applied the same method that is derived from low of the duality. However because of the lack of the dynamic behavior of primal-dual relationship, we get premature result.

Than the integer model is converted into the bi-linear system. This bi-linear system gives the same result with the MIP system. However bi-linear systems are non-convex and there is not an easy way of solving them. Therefore first of all we applied existing commercial solvers (Baron-Dicopt) to the bi-linear system to see whether there is an advantage of passing from MIP domain to bi-linear domain. However it is seen that the solvers for MIP is more successful than non-linear solvers.

After underestimator schemes are developed to solve the bi-linear system. Because the underestimators are the branch of the approximation algorithms and the results are highly dependent on the bounds of the variables, good enough results cannot be obtained.

As a result a hybrid genetic algorithm is developed and applied to the bi-linear system. Even if still the MIP solver (Cplex) is the best, the heuristic algorithm becomes more successful than the commercial non-linear solvers.

Chapter 3

A NEW MODELING PRACTICE FOR SUPPLY CHAIN NETWORK OPTIMIZATION PROBLEMS

Introduction:

In this thesis we focused on the modeling, solution and application of the supply chain systems. The modeling part is the main focus of this chapter. In this chapter the supply chain networks are modeled by using different perspective. Before explaining the details of the remodeling the supply chain networks, the types of models that are considered in this chapter are described. We considered multi-echelon supply chain networks to model. The networks are optimized in the multi-period system. Therefore dynamic behavior of the system can be caught. This dynamic behavior is modeled by using differential equations, at first. However to linearize the system, the differential equations are converted into difference equations. Therefore the models that we are considered in this chapter are a multi-echelon, multi-period supply chain network model whose dynamic behavior is represented by difference equations.

The supply chain networks are complex systems where a number of products are produced and transferred from one node to another with the objective of satisfying the customer demand. The supply chain systems are traditionally decomposed into different cycles such as procurement, manufacturing, replenishment and customer order cycles due to modeling and computational challenges [48]. The nodes of the supply chain network are considered individually in such decomposition approaches. As a result of this decomposition, the accuracy of the model in representing interactions among the multi-echelon supply chains becomes weak due to preference of an individual benefit instead of system wide benefit. The inefficiencies in the modeling of interactions among the nodes of supply chain network can be eliminated by using an integrative modeling.

Besides decomposition properties, there are many operational properties of supply chain networks. In a multi-echelon SCN, the production begins with arrival of raw materials from a supplier to production plant. Once a production of a specific product begins, an inventory of the corresponding product starts to accumulate in different stages of the SCN. The transportation of the end product is required to distribute this inventory in different stages of a SCN other than production plant. An important consideration in optimization of SCNs is the distribution of inventory and scheduling of production with transportation. In addition, integrative modeling should take into account that the information that is created by customer is transmitted in the opposite direction of material flow as shown in the previous chapter. The decisions in the optimization of operations in a SCN should achieve desired objectives. In some networks, the main objective is the minimization of cost, but in some others, it is the maximization of revenue. The decisions that are taken by an individual node are determined by other nodes in some SCNs; but in some other SCN this dependence is not observed.

During the modeling of supply chain systems that are briefly described, a different set of equations or constraints are used for each entity or node and flow. This means that the network topology is considered as a part of model. However in the approach that is developed in this thesis, the network topology is considered a data. The main rule of modeling is the data-model independence. This means that data of the system should be separated from the model itself to increase the re-usability and modularity. Therefore, because topology (networks and flows) is considered as data, this topology should be removed from the model it-self. To succeed this, the modeling approach that is used during the modeling of the biological network is used.

Instead of modeling each flow and node with separate set of equations, in this new developed approach, the matrix formation of the network is created and this matrix is used as a topology data. One claim against to our approach is using or increasing the number of indices in the model to decrease the number of equation blocks. However by indices or by using just one-dimensional vector, we can only model the entities or nodes. To also model the flows we need two-dimensional structure, which is matrix in our case.

The topology can be captured in the matrix format. Because basically topology includes nodes and the flows. Therefore, two-dimensional data is required to represent 2 parameter system (nodes and flows). However we can extend this idea to three or four dimensions matrices. For example we can add time issue to the system. If our supply chain reflects behavior with respect to node, flow and behavior, we can now capture topology in 3D matrices. For example a flow between 2 nodes cannot carry financial or product for certain times. Then this data can be captured in the 3D matrix, and instead of keeping this behavior in the equations block. Our claim in this approach is all given behavior or all the expected behavior for the system should be classified as a data. Data should be removed from the model. After this separation the model that we obtain is the pure model that does not include any trace from data. Because now we get the pure model, the extension of it becomes easier. Also pure models are easy to modularize, so any GUI integration becomes easier and more appropriate.

In the literature, systems are generally modeled without separating topological data from the model. As a one of the pioneer of the multi-echelon supply chain, Perea-Lopez et. al, models each flows and nodes separately [49]. This modeling technique is also true. Our aim in our method is developing philosophically true modeling technique.

In software engineering, there is a concept, model-view separation. According to this concept, the object or the objects that creates the model should not have any direct knowledge of the GUI objects [50]. In other words, the object that represents the logic should not have any information from the objects whose only aim is interacting with outer user. If we interpret this concept with our case, the model without data can be classified as logic of the system. Because obtaining equations and objective function that do not include any data from the outer world is our aim. However, the data itself does not include any logical information. Data only includes information from outer world, without including any logic information. For the sake of the correct design, this model-data separation should be realized. Other-wise model of a system can be used only once, and the ability of re-usability decreases as the bond between data and model becoming stronger.

Most of the models are developed to code. In other words, without solution, a model is useless. Todays, models are very complicated and without computer it is nearly impossible to get a solution. Therefore we can conclude that, almost every model is developed to code in one of the optimization library. So indeed optimization of a system can be classified as a software engineering process. Some of the rules of the software design can be used especially the modeling part of optimization process. The atomicity of the design is very important for design. The atomicity principle is known as information hiding [50]. In other words, during modeling process some part of model should be hided from other parts. This can be sometimes called encapsulation. Encapsulation is the grouping of data and the operations that apply them to form an aggregate while hiding the implementation of the aggregate.

In conventional modeling techniques, the entities of the supply chain are divided into groups according to their functionality. Yimer and Demirli model their supply chain networks with respect to this idea [51]. For each entity or node type for instance distribution center, retailer, they defined an index. Each index set indices for each entity one by one. Modeling, optimization, and solution are done with respect to this index separation.

In the modeling technique that is we developed, all the nodes are considered as same. Therefore, there is not any division between nodes. All of the nodes are considered as an element of set of nodes. Like conventional modeling we do not make any discrimination with respect to function or any other attribute.

Indeed in this modeling technique, we perceive an optimization model like a simulation model. In most of the simulation model, supply chain is divided into five types of constructs: nodes, arcs, components, actions and policies [52]. However optimization models look at supply chain systems differently. Different from simulation models, most optimization models look at the bi-nodal flows and optimize them. During the optimization, the policies are written in in terms of the equations. Also optimization models choose best from the feasible region. In contrast, simulation models create the feasible region. Our method, like simulation model, makes the same division for supply chain model. A simulation model assign attributes to each node or arc to determine their behavior. In our approach, for each type of construct, we can create a dimension and assign behavioral attributes to the optimization model by using the corresponding dimension of the matrix and its entries. Therefore, we do not make division within each type of construct, and consider each construct as it.

In supply chain models, the Bill of Materials (BOM) are also included [12]. By this way, network optimization can be made in the raw material level. The inventory and procurement values and periods can be determined by this model. Sometimes this kind of detailed model can be classified as a scheduling model. In this kind of model, the raw material and BOM part of the system are classified as a different entity and for them different kind of equation blocks are developed. However, this kind of modeling practice makes the modeling more complex. In our method, all of the raw materials are classified as entity or node in the supply chain model and the procurement of them does not show any difference from the flow of end product from any node in the supply chain. All the raw materials and their flows, including procurement and outsourcing, can be represented in the topology matrix that we have developed.

3.1. The Method of Modeling:

The Supply Chain Systems are modeled in a different ways in the past studies. However, all of them use the same methodology during modeling. First of all, the entities or nodes are determined in the network. Then, for the flow of commodities between any two nodes, a variable is assigned. These variables are generally real variables instead of integer variables. After this step, the supply chain network is modeled like a classical multi-echelon supply chain system. However, this modeling technique has some drawbacks. First of all extending the model (adding new nodes) is very difficult. During the extension of the past model new variables should be added into system and to add these new variables, the model should be known in detail. Otherwise, the model cannot work correctly. Also, the original model and the extended model are developed by different persons. Because modeling is an art, the way that model same system show difference from developer to developer and understanding the some others' modeling technique may require an expertise. Also way of modeling a supply chain network should be suitable for GUI integrated model development process. By this way the end users can develop model without much effort.

3.1.1 Logic of Modeling Technique:

The modeling technique developed in this thesis considers a supply chain system as a biological (metabolic) network. A standard metabolic network is represented by metabolites and reaction rates (Figure 3.1).

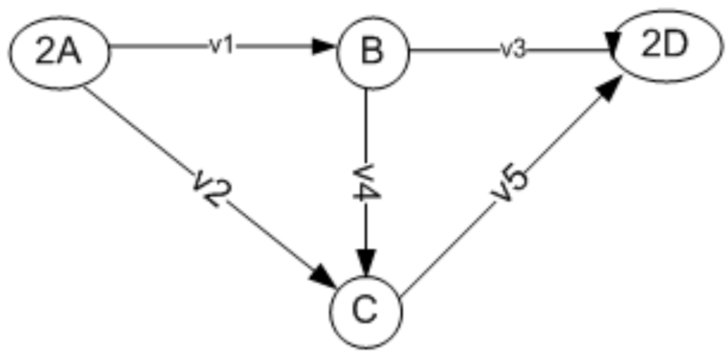
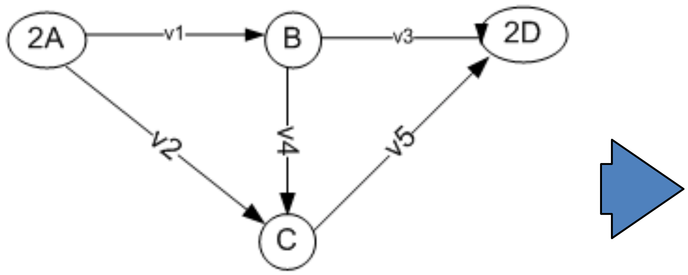


Figure 3.1: A sample metabolic network.

The network presented in Fig. 1, includes 4 metabolites and 5 reactions. Metabolites are the ingredient or result of the each reaction. The reaction is basically a flow between two metabolites set. The reactions rates of each reaction represented by *v* . This reaction rates is the total amount of consumption or production of a metabolite in unit time. For instance, the reaction 1, consumes 2 units of metabolite A and produces 1 unit of metabolite B and the rate this conversion is v_1 in a unit time. Therefore total $2v_1$ amount of metabolite A is consumed and total v_1 amount of B is produced in the unit time as a result of reaction 1. The consumption and production schema of any metabolic network can be represented in matrix format (Fig 3.2) [53].



	v1	v2	v3	v4	v5
A	-2	-2	0	0	0
B	1	0	-1	-1	0
C	0	1	0	1	-1
D	0	0	2	0	-2

Figure 3.2: A sample metabolic network and corresponding stoichiometric matrix.

In Fig. 3.2 we can see the matrix representation of the metabolic network shown in 3.1. This matrix is called as a stoichiometric matrix [54]. In this matrix, each raw corresponds a metabolite and each column corresponds a reaction. Minus sign stands for consumption and plus sign stands for production. If we look at the v1 column of the matrix we can see that the value corresponds to A is -2 and the value corresponds to B is 1. The remaining entries are 0. This means as a result of reaction 1, for each unit B that is produced, total 2 unit of B is consumed. At steady state, the rate of change of each metabolite is equal to 0 [55]. Steady state is the state that for each metabolite, total consumption and production is equal to each other. This state is steady because independent from time, due to this equality restriction, the composition of system stays same. Therefore, with the assumption of steady state, the Eq. 3.1 is used to model a metabolic network.

$$S.v = 0 \tag{3.1}$$

where S is the stoichiometric matrix, and v is the flux vector.

According to Eq. 3.1, the *S* matrix is parameter, in other words it is given data and basically represent the network topology and relative values of reaction rates in terms of metabolites. The variable is the *v*, the reaction rates or flows. As a result of the Eq. 3.1, the all-feasible set of values of reaction rates (*v*) can be obtained. For each of these reaction rates, the steady state can be reached.

There can be multiple solutions for Eq. 3.1 therefore as stated before we can obtain many *v* values that satisfy Eq. 3.1. To limit these solutions in a logical way or to get a *v* value that is more meaningful, an objective function can be added into the system presented by Eq.3.1. By this way, the feasible solution whose satisfaction rate of the objective function is the highest is selected as a behavior that resembles to the behavior of in-vivo system.

Without objective function we have many feasible solutions, but we cannot decide which one is the more significant. To determine this significant solution we need an objective. With respect to this objective, the more significant feasible v value can be obtained.

In the metabolic network context, the objective is can be maximization of certain metabolites, or minimization of the production of certain metabolites such as ATP. Therefore, with the addition of the objective function, the final optimization model for metabolic networks becomes Eq. 2.

$$\begin{aligned} &\max \quad M \\ &st \\ &S.v = 0 \\ &v \geq 0 \end{aligned} \tag{3.2}$$

where M is a metabolite whose concentration is maximized.

In Eq. 3.2 we add only a constraint to the system in Eq. 3.1. By this way we can obtain more meaningful value of *v*. In 3.2, the most important point is the selection of the objective because which solution that we will consider is depend on this objective. There are some assumptions for the biological organisms. According to these assumptions, it can be accepted that the main aim of a biological organism is maximization of ATP production or biomass production. If we put this objectives to the system, then the *v* values that we get becomes more tuned and significant.

The formulation in Eq. 3.2 is developed to model a metabolic network, but it has some problems. While the process in Eq. 3.2 progressing, some of the metabolites are released to environment and the concentration of outside put a pressure onto the system inside. This pressure is modeled by putting upper and lower bounds onto the values of the fluxes. The algorithm that models the feedback relation between the inside of cell and its environment is presented in Fig. 3.3.

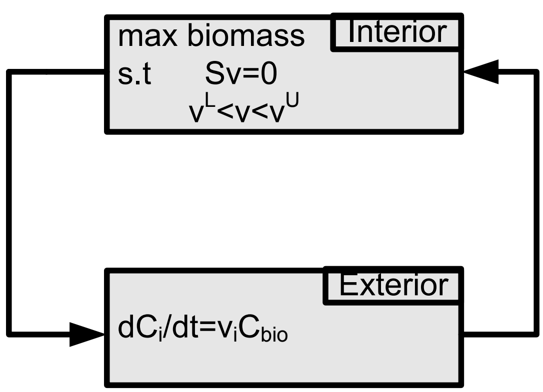


Figure 3.3: Algorithm that models cell inside and environment relationship.

In this part, the dynamic behaviors of the metabolic networks that take place in biological organisms are described. Even if all these reactions take place inside a biological organism, they are still represented by the networks, flows, equations and optimization models. In this thesis our main focus is on the supply chain networks and supply chain optimization models. We made this brief description of metabolic network to show the resemblance between the biological networks and supply chain networks. Both in supply chain network and biological network, there are metabolites or nodes and flows or reactions. In the metabolic network, reactions basically convert one metabolite to other metabolite. For the supply chain, flows convert a product or raw material from one state to other state. For example the product in warehouse and the product in the retailer are not same from the supply chain perspective. Even if they are the same products, even from the eyes from ERP systems and accounting systems, they are considered as different. So flows also converts products from one state to other one in the context of supply chains.

3.1.2 Remodeling in the Supply Chain Context:

The approach that is used to model supply chain system in this thesis is inspired from modeling of the metabolic networks. A production, distribution, collection and recycling operations together looks like a metabolic network (Fig. 3.4). As presented in Fig. 3.4, the A, B and C are the raw materials that are taken from environment. The procurement rates of these raw materials are v_1, v_2, and v_3. P_1 is the sub-module and it is produced by using 1 unit of A and B and the corresponding production rate is equal to v_4. P_F is the end product which is located in the factory. P_D is the product which is located in the distribution center and transportation rate of the it from factory is equal to v_6. P_R is the product in the retailer and its demand rate is equal to v_8.

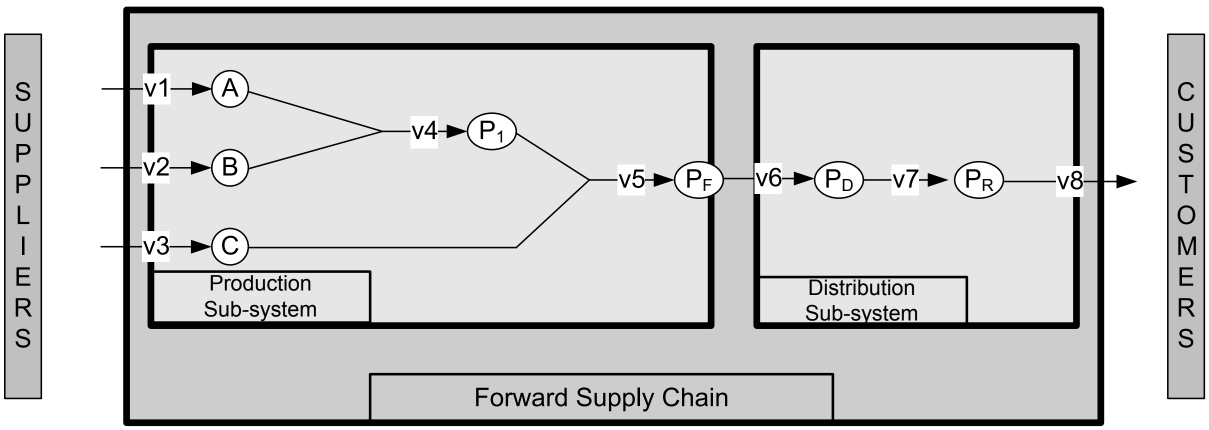


Figure 3.4: A sample forward production network.

As seen in Fig. 3.4, the system is a network of the reactions and products, semi-products and raw material. As defined before, in this system all the nodes, metabolites or in other words, all the states are considered as an element of same set. Also all flows or reactions are the element of the reaction set are the element of the same flow set. Therefore, the related topology can be represented by a matrix, so called stoichiometric matrix in the metabolic network.

The corresponding stoichiometric matrix for the network in Fig. 4, is given in Table 1.

	V1	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**
A	1	0	0	-1	0	0	0	0
B	0	1	0	-1	0	0	0	0
C	0	0	1	0	-1	0	0	0
P1	0	0	0	1	-1	0	0	0

PF	0	0	0	0	1	-1	0	0
PD	0	0	0	0	0	1	-1	0
PR	0	0	0	0	0	0	1	-1

Table 3.1: The stoichiometric matrix of the supply chain network (SCN) in Fig. 4.

As seen in Table 3.1, the matrix includes 2 behavior of the matrix: states of the material (nodes) and the flows (state converter). Columns are represented by the flows and the rows are represented by the states (metabolites in the biological networks). As described before the in the each column, the numbers represent the consumption and production of the products states by the corresponding flows. The minus sign is for the consumption and the plus sign is for the production.

As presented in the Fig. 3.4, the A, B and C are the raw materials. Therefore they are not produced; instead, they are procured by the suppliers. This means that for the network of our supply chain, these products are taken from outside by the corresponding flows: v_1, v_2, and v_3. In the first, second and the third columns of Table 3.1, only the values of A, B and C is 1 consecutively. The remaining entries are equal to 0. This means that these products are totally taken from outside without any in-house production. In contrast, if we look at the flow v_8, PR is the end state of the products and it is only emitted into the system by v_8, without any consumption. The counterpart of PR in the biological organism is the metabolite that is emitted to the environment like glucose for the consumption.

In the Table 3.1, only 2-dimension of the network is represented: flows and product states. This means that during the modeling if we use this matrix, system behavior only shows change with respect to flow and node. In other words, the matrix help to model a supply chain by manipulating the decisions with respect to node and flow.

Basically with the help of this matrix, the feasible set of solutions are obtained. By using the objective function we can obtain the best solution according to our concern.

However if we want to add another decision criteria, we can succeed this by increasing the dimension of the matrix. For example, if time is also very important for the decisions criteria for our system than we can add a third dimension to our matrix. This time, we can say that the topology is determined by nodes, flows and time. Topology does not only reflects the visual state of the network. Besides, for the supply chain the topology, it reflects the behavioral characteristics of the network. For example, if certain flows become active only during certain time spans, this behavior should also be reflected by topology. Because our aim is separation of the topology from the model, by adding the time dimension into the stoichiometric matrix, we can succeed our aim.

In the metabolic network, the objective function is chosen as a minimization or maximization of fluxes of certain metabolites. However, for the SCN in Fig. 3.4 the objective will be the maximization of the profit. The revenue of the SCN in Fig. 3.4 generated from the flux of the v8. Other remaining fluxes are the cost issues for the SCN. All of the cost and revenue issues are shown in the Table 2.

Cost Issues	**Abbreviation**	**Formulations**
Cost of Procurement of A	CP_A	$v_1.c_A$
Cost of Procurement of B	CP_B	$v_2.c_B$
Cost of Procurement of C	CP_C	$v_3.c_C$
Production cost of P1	CP_{P1}	$v_4.c_{P1}$
Production cost of PF	CP_{PF}	$v_5.c_{PF}$
Transportation cost between factory and dist. center.	CT_{FD}	$v_6.ct_{FD}$

Transportation cost between dist. center and retailer.	CT_{DR}	$v_7.ct_{DR}$
Revenue	R	$v_8.prc$

Table 3.2: Cost and revenue items.

where c_A is the price of raw material A, c_B is the price of raw material B, c_C is the cost of raw material C, c_{P1} is the production cost of the sub-module *P1*, c_{PF} is the production cost of end product P_F, ct_{FD} is the transportation cost of the end product from factory to distribution center, ct_{DR} is the transportation cost between distribution center and retailer, *prc* is the selling price of the product at the retailer.

With these cost items, the corresponding formulation is :

$$\begin{aligned} &\max\ R - (CP_A + CP_B + CP_C + CP_{P1} + CP_{PF} + CT_{FD} + CT_{DR}) \\ &s.t. \\ &S.v = 0 \\ &v \geq 0 \end{aligned} \tag{3.3}$$

The model in Eq. 3.3, maximize the profit without considering any constraints rather than network. However in real life application of the SCNs, besides network, the transportation constraints, fixed charges, quantity discounts, and inventory holding conditions should be considered. Otherwise, the result of the model will not be realistic and consistent.

In addition, the formulation in Eq. 3.3 only ensures steady state, which is the state that production and consumption of all production states are equals to each other. However for the supply chain system, the steady state cannot be the case. Because every steady state can be sub-optimal for the system. The state of the supply chain should be dynamic. In other words it can be said that there can be some steady state that is optimal, however it cannot be ensured every steady states are optimal.

Sometimes accumulation of some products can be required and sometimes, consumption of them can be required. For all these reason, the optimality of supply chain cannot be steady state. However at some time interval, the steady state of the system can be optimal. Therefore, our aim should be finding of all states independent from their steadiness.

To deal with this problem, a hybrid modeling approach is applied. In this approach instead of solving Eq. 3.3 only once, the optimization span is divided into sub-parts in the time domain. Between each part, the inventory of each raw material, sub-module and end product is updated. Also, because the inventory levels are hold, bounds for the fluxes are fixed by the inventory levels. The proposed algorithm is presented in Fig. 5.

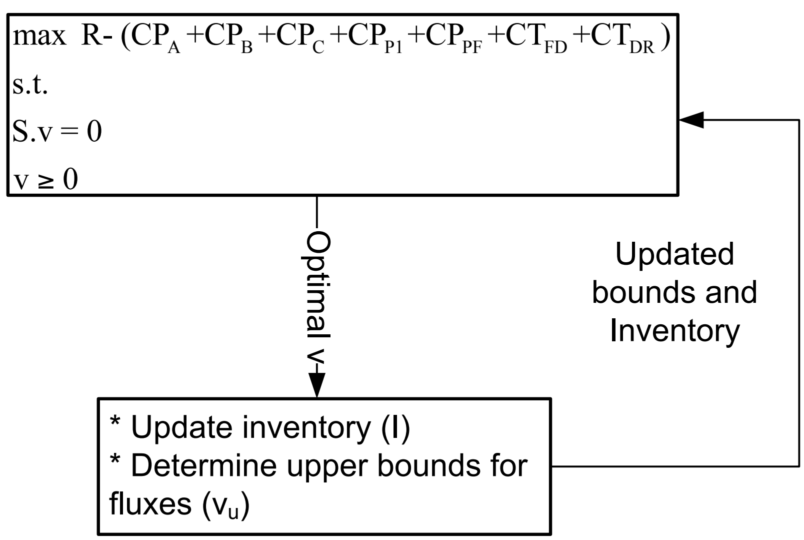


Figure 3.5: Algorithm for the solution of the hybrid SCN model.

As seen in Fig 3.5, we have proposed an iterative approach. First the assumption steady state is accepted for the very short length of time period. For this period, the production and consumption value of each supply chain state and node is determined, basically the v values are determined for each reaction or flow. Indeed this v values are the rate of the change of each state of product in unit time. We modeled our initial supply chains with differential equations, but then to provide the linearity, we converted them to difference equation in the domain of time. This v value that we determine using the assumption of the short time steady state of the system, is equal to incremental change of the state of the each product in the supply chain.

After this incremental change is obtained, the inventory or the state of the supply chain is updated. With this new state, the future state is obtained by again using assumption of the steady state. This incremental approach continues till the end value of the time value is reached. Actually, the incremental algorithm in Fig. 3.5 is demonstrated to show the logic of it. A non-iterative method can be developed that can makes the same job of algorithm in Fig. 3.5. The algorithm in the figure is divided in the time domain, therefore if an update function is developed that shows dynamic behavior in the time domain, then a non-iterative algorithm can be developed by this way.

After the proposed updates are applied to the algorithm in Fig. 5, the resulting model will be the following:

$$
\begin{aligned}
&\max\ R - \ (CI + CP_A + CP_B + CP_C + CP_{P1} + CP_{PF} + CT_{FD} + CT_{DR}) \\
&s.t. \\
&S.v \ = \ 0 \\
&I(m,t) = I(m,t-1) + S(m,v).v(t) \quad \forall m,t \\
&v \geq 0 \\
&I(m,t) \geq 0 \quad \forall m,t
\end{aligned}
\tag{3.4}
$$

Where *CI* is the inventory holding cost, *m* is the set of products, semi-products and raw materials it can be classified as states of the supply chain, *t* is the set of time, *r* is the set of flows, *I(m,t)* is the inventory of product *m* at time *t*, and *v(t)* is the flux of flow or incremental change of each state at time *t*.

With the set of equations in Eq. 3.4, the system will work 0 inventories, because SCSs holds equations to hedge itself from setup costs, quantity discounts, and batch production constraints. Therefore without the existence of foregoing reasons, any SCSs do not hold inventory. However, to make the model realistic, these properties should be considered in the model.

However, in this part our aim is the development of the method that excludes the topology of the network given out of conventional model. Therefore, the aim of the model in Eq. 3.4 is not the demonstration of the full-scale supply chain model. Indeed, it is the demonstration of method that shows the exclusion of the topology out of the model. This method basically makes the model more lean and clean. Also because by this way all given information is behaved like data, the maintenance of the model and system become easier. Extension of model becomes quicker. Actually, by this way the model is designed with respect to software engineering rules. Because one of the most important rules of modeling is the data-model independence. Therefore during model, the main aim should be isolation of data from model.

3.2. Case Study for New Modeling Technique:

In this part we will first introduce a full-scale supply chain model. The initial version of this model is shown in the previous chapter. However in this chapter we will give the model in detail. After model is introduced, it is remodeled by using the new modeling technique developed in this chapter. The aim of this demonstration is proving the benefit of new modeling technique. After applying the changes preferred by using the method developed, the proposed changes will be seen. To show all these changes, first of all, the model and its logic is described in detail.

In the supply chain that we will study, we consider that all nodes in the supply chain show independent behavior; so there is no specific rule that determines decisions taken by a specific node. The holding cost of each product at each node is calculated separately. Besides transportation delay exists for the transfer of the product from one node to other node. However, transfer of a product from plant to plant warehouse is instantaneous, since plant and plant warehouse are located in the same geographical region. During the production phase, the plant can take unlimited amount of raw material from any supplier and the additional cost occurred because of this transfer is transportation and raw material cost.

Although the demand to retailer is satisfied immediately, the supply chain system does not have to immediately satisfy the demand of the previous node for the remaining nodes. Therefore, unsatisfied order is accumulated at each node to be satisfied in the future. Once material is sent from an upstream node, it reaches to corresponding downstream node after some transportation delay. Inventory and order accumulation balance is updated with respect to the amount of sent and received material.

Each product has a different production time and the production facility can produce one of these products at each time interval. Production schedule is non-preemtive: once the production of one product begins, none of the other products can be produced till production of that product is completed. Besides, switching from one product to another requires a setup time.

Orders and inventory shows dynamic behavior in the SCN. The dynamic behavior of the SCN is formulated as multi-period system of equations. In addition, the price elasticity equations that establish the relationship between price and demand are used in the model formulation. The price elasticity equation is represented with differential equations; these differential equations are converted to difference equations in order to incorporate them into multi-period optimization problem. Also binary variables are used to model the production nodes since these nodes show discrete behavior by switching from one product to another in time.

3.2.1 Inventory Balance:

The inventory is the material that is held in the nodes of the supply chain system, after periodic material transfer ends. All supply chain systems should hold inventory to continue their operations to increase the customer satisfaction. Without inventory, the desired levels of customer satisfaction cannot be reached. The reasons of this are the uncertainty in the demand, existence of complex production systems, and delay in the transportation. The inventory balance of our system exhibits continuous behavior on the time domain and rate of change of inventory can be represented by differential equations.

However, existence of differential equations could make the system computationally intractable. Therefore, the rate of change of inventory is discretized in the time domain and differential equations are written with difference equations. The inventory balances for different nodes of the supply chain network are modeled with the following equations.

Final Product inventory at the warehouse:

$$I_{nkt} - I_{nk(t-1)} = \sum_{a} PR_{nkat} - \sum_{k''} yp_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{pw} \tag{3.5.1}$$

where k'' denotes a downstream nodes (distributor nodes).

Final Product at Distribution Center:

$$I_{nkt} - I_{nk(t-1)} = \sum_{k'} yp_{nk'k(t-\delta_{k'k})} - \sum_{k''} yd_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{ds} \tag{3.5.2}$$

where k' denotes upstream nodes (production facility nodes) k'' denotes downstream nodes (distributor nodes).

Final Product at Retailer:

$$I_{nkt} - I_{nk(t-1)} = \sum_{k'} yd_{nk'k(t-\delta_{k'k})} - \sum_{k''} yr_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{rt} \tag{3.5.3}$$

The final product inventory at the retailer k is a function of previous inventory level ($I_{nk(t-1)}$) and material transferred from distribution centers ($yd_{nk'k(t-dk'k)}$) with a time delay ($d_{k'k}$) and sales to customers ($yr_{nkk''t}$).

3.2.2 Order Balance

Our SCN is triggered once demand is realized. However, in some cases demand cannot be satisfied immediately. In that case, past demand is accumulated in the corresponding node of the supply chain system and once the enough material is obtained, the accumulated demand may be satisfied. The satisfaction of accumulated demand is determined with respect to given cost and production parameters to optimize overall supply chain profit. The order balance equations are originally differential equations and to make the system computationally tractable, the differential equations are re-written with difference equations similar to the inventory balance equation (Eq. 3.5.1 to 3.5.3). The equations of order balance are the following:

Order Balance at Plant Warehouse:

$$O_{nkt} - O_{nk(t-1)} = \sum_{k''} up_{nkk''t} - \sum_{k''} yp_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{pw} \tag{3.5.4}$$

Order Balance at Distribution Center:

$$O_{nkt} - O_{nk(t-1)} = ud_{nkk''t} - yd_{nkk''t} \quad \forall n \in N, \forall t \in T, \forall k \in N_{dc} \tag{3.5.5}$$

Order Balance at Retailer:

$$O_{nkt} - O_{nk(t-1)} = \sum_{k''} ur_{nkk''t} - \sum_{k''} yr_{nkk''t} \quad \forall n \in N, \forall t \in T_f, \forall k \in N_{rt} \tag{3.5.6}$$

3.2.3 Production

The production facility in our SCN is a single stage, multi-product facility that has multiple production lines. Therefore, only single product can be produced in one line of the facility at each time interval. Also preemption is not allowed in the plant; once a production of a product is started, the facility cannot produce any other product on the corresponding assembly line during the production time of selected product. In addition plant is designed to produce constant amount of product during each time interval once a produce order is received. In this production system, the switches between operational and non-operational phase of the manufacturing plant as well as the transition from one product to another are modeled by propositional logic. The production policies the production stage of SCN is modeled with the following equations (Mestan, et. al.):

$$PR_{nkat} \le pr_{nkat} PR_{nkat}^{U} \quad \forall n \in N, \forall t \in T, \forall k \in N_{pr}, \forall a \in A \tag{3.5.7}$$

$$\sum_{n} pr_{nkat} \le 1 \quad \forall t \in T, \forall k \in N_{pr}, \forall a \in A \tag{3.5.8}$$

$$\sum_{t'=t}^{t+l_n-1} pr_{nkat'} + \sum_{n' \ne n} \sum_{t'=t}^{t+l_n-1} pr_{n'kat'} \le 1 \quad \forall n, n' \in N, \forall t \in T, \forall k \in N_{pr}, \forall a \in A \tag{3.5.9}$$

Production of all products in an assembly line should be less than maximum production amount, PR_{nkat}^{U} in each time interval as given in Eq. (3.5.7). However, the batch production strategy cannot be applied since the production quantities should be equal to predefined batch size with this constraint. Therefore, if production node operates as a batch system, then the inequality in the Eq. (3.5.7) should be converted to equality. Production of single product in each time interval on every assembly line is guaranteed in Eq. (3.5.8). Production of only single product between t (production start time) and $t+l_n-1$ (production finish time) is assured with Eq. (3.5.9) for each production line.

3.2.4 Upper Bound on Demand Satisfied by the Retailer

In the modeled SCN, retailer does not have a capability of satisfying demand as much as it can by using its inventory. In the system, an upper bound exists for retailers that regulate the amount of flow of each product in each time interval that are referred to as the maximum flow (MF) constraints. Because of this regulation, system-wide average inventory increases.

3.2.5 Objective Function

The objective of considered supply chain is the maximization of the profit which is formulated as revenue minus cost of operations. In the system, revenue is only created by the interaction between the retailers and customers. Once the customer buys a product, revenue is created in the system. Although, there is only one source of revenue, there are many cost items in the system including holding cost of inventory in each node of system, transfer cost of materials between nodes, raw material cost that comes from supplier, fixed and variable cost of the production.

One of the most important goals in supply chains is the satisfaction of the demand generated by the customer. It is a well-known fact that the customer demand is highly sensitive to the price of the products (Simchi-Levi, et. al., 2003). In this paper, the relationship between the demand and price are modeled using the price elasticity concepts. Detailed analyses of the price elasticity of the demand as well as the formulation of objective function with required modifications are provided in the following section.

The objective of the supply chain system is the maximization of the profit. The total profit is given as,

$$Z=C_{RE}-C_{HO}-C_{TR}-C_{RM}-C_{PF}-C_{PV} \qquad (3.5.10)$$

The total profit is calculated by subtracting all of the cost items including the cost of holding inventory (C_{HO}), transportation cost (C_{TR}), raw material purchasing cost (C_{RM}), fixed and variable production costs (C_{PF}, C_{PV} respectively) from the revenue generated by the sales of the products (C_{RE}). The revenue is calculated from the following equation:

$$C_{RE}=\sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{rt}}\sum_{k''\in N_{cs}} Prc_{nt}\, yr_{nkk''t} \qquad (3.5.11)$$

The costs are calculated with the following equations:

$$C_{HO}=\sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{pw}} I_{nkt}HC_{nkt}+\sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{dc}} I_{nkt}HC_{nkt}+\sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{rt}} I_{nkt}HC_{nkt} \qquad (3.5.12)$$

$$C_{TR}=\sum_{t\in T}\sum_{n\in N}\left(\sum_{k\in N_{pw}}\sum_{k''\in N_{dc}} yp_{nkk''t}TC_{nkk''t}+\sum_{k\in N_{dc}}\sum_{k''\in N_{rt}} yd_{nkk''t}TC_{nkk''t}+\sum_{k\in N_{rt}}\sum_{k''\in N_{cs}} yr_{nkk''t}TC_{nkk''t}\right) \qquad (3.5.13)$$

$$C_{RM} = \sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{pr}}\sum_{s\in N_{sp}}\sum_{a\in A} PR_{nkat} Req_{ns} RC_{ks} \tag{3.5.14}$$

$$C_{PF} = \sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{pr}}\sum_{a\in A} pr_{nkat} FC_{nk} \tag{3.5.15}$$

$$C_{PV} = \sum_{t\in T}\sum_{n\in N}\sum_{k\in N_{pr}}\sum_{a\in A} PR_{nkat} VC_{nk} \tag{3.5.16}$$

3.3. Re-Modeling Process:

The multi-echelon, multi-period supply chain system is modeled by using the Eq. 3.5.1 to 3.5.16. If we look at the equations in detail we can divide it to 3 sub categories: related to network, related to specific cases and objective. The Eq. 3.5.1 to 3.5.5 are related with the networks. Therefore we can re-model them by using the method that we have developed. The Eq. 3.5.7 to 3.5.9 are related to the specific case of the system. In our case it is related with the pre-emption of the production system. Because it is a specific case for the supply chain, in any case for every kind of modeling system, it is modeled in such way. The last parts of the equation sets are related with objective function. However most of the cost and revenue issues are related with the topology of the supply chain network. In the S matrix that we are holding to represent system topology is 2-dimensional for the current case. The rows are the supply chain states. They can be raw materials or en product in the different levels of the supply chain. The columns are the flows. For example a flow between a product node in warehouse to the same products node in the distribution center can be interpreted as a transportation flow. Therefore the cost related with this flow is the unit transportation cost between warehouse and distribution center. Now lets assume that there is a flow between raw material and the product in product warehouse. This flow is classified as a production process and the cost related to it is the production cost. As seen, most of the cost issues can be related to unit changes of the flows, *v*.

For the new model, m is the set of all nodes in the system, *v* is the set of all flows and d_v is the delay amount for each of the flow, v_r is the reverse flow in other words it is the flow of the information. With this parameters and variables the Eq 3.5.1 to 3.5.5 is modeled like in Eq. 3.6.

$$\begin{aligned} &I(m,t) = I(m,t-1) + S(m,v).v(t-d_v) \quad \forall m \in M, \forall v \in V, \forall t \in T \\ &O(m,t)=O(m,t-1)+v_r(t) - v(t) \quad \forall m \in M, \forall v \in V, \forall t \in T \end{aligned} \tag{3.6}$$

The objective function can also be remodeled. Most of the cost issues are related with the flows and the remaining ones are either related with order accumulation or total inventory. If c_v is the cost of flow v, *ci* is cost of inventory and c_o is the cost of the order accumulation, then the remodeled case of the objective can be seen in Eq. 3.7.

$$\sum_t v(t).c_v(t) + \sum_{m,t} I(m,t)c_i(m) + \sum_{m,t} O(m,t)c_o(m) \tag{3.7}$$

As seen in Eq. 3.7, all of the objective function can be written with single block of equation.

3.4. Extension of the Stoichiometric Matrix:

The matrix that we have used during the remodeling is 2-dimensional matrix. We used it because of the logic of the problem. In the problem, the dynamic structure is under the effect of the two parameters: the supply chain stages and the flows or rate of flows between the two stages. However sometimes, o new third parameter can be added into system. For example time can be the third parameter that determine the supply chain behavior. We can put a constraint like the following: between t=t1 and $t=t_2$, the flow v_1 and v_2 should be zero. If that is the case, now we can say that the system's behavior is also determined by time. Therefore now we have 3 parameter that determine the behavior, time, flow and state. So to add this property into our stoichiometric matrix, we will add one more dimension. The S matrix now becomes 3-dimensional. On the first dimension we will put nodes, the second dimension is the flow and the third dimension is the time. Then the 2 slices of the matrix becomes all zero. The first slice is the plane that is determined by x=1 to N (all nodes), y=v1, $z=t_1$ to t_2 and x=1 to N (all nodes), $y=v_2$, $z=t_1$ to t_2. By using this matrix as S matrix, without adding extra standalone constraints and by separating all data from model, more lean and modular model is obtained.

As determined before, same modeling practice will not be obtained by using one more index in the model. Because index is one dimensional, however in our constraint set, the system behavior is changed with respect to flow, node and time. By using just one index, it is impossible to model. Therefore more dimensional data object which is matrix is required to make modeling more lean.

3.5 Summary:

In this chapter a new modeling methodology is developed and it is applied to sample supply chain system. In the conventional modeling practices, for each node and for each flow and even sometimes for each time step different sets of constraints are used. These constraints are generally used to model the topology of the system or network. However, in the best practice of the modeling, we know that data and model should be separated from each other. In our idea, the topology is also data, therefore model of different topologies should not be different. Only the data part of them should be different. However they should have same model.

To separate the topology of the model, or to separate topology related behavior from the model, we have used stoichiometric matrix to hold the properties of the system topology. In conventional supply chain, the behavior of the system is at least depending on the supply chain state and the rate of change of state of each entity of the system. Therefore at first a 2-dimensional matrix is used to hold the system topology. However some times more parameters are added into system to determine the behavior, such as time. If that is the case, a more dimensional matrix is used to capture topology.

The reason of use of this modeling technique is the applying the best practice of modeling methodology. In the best practice, the model and data should be separated so-called model-data independence. The reason of this independence is to make the model maintenance, re-modeling and model-to-model integration easier. Because with this new modeling technique, without changing the model, by just manipulating data set, a new extended model can be obtained.

3.6 Nomenclature:

Because the supply chain model that is used in this chapter is specific to itself, the nomenclature of it is given in here.

Indices:

a	production line a in production facility
k	node k in the SCN
k'	upstream node of node k in the SCN
k''	downstream node of node k in the SCN
n	product n
t	time t

Sets:

A set of assembly lines

N_{cs} subset of nodes k that represent customers

N_{dc} subset of nodes k that represent distribution center

N_{pr} subset of nodes k that represent manufacturing plants

N_{pw} subset of nodes k that represent warehouses at plants

N_{rt} subset of nodes k that represent retailers

N set of product n

T set of time intervals t

Decision Variables:

I_{nkt} inventory level of product n in node k at time t

O_{nkt} order accumulation level of product n in node k at time t

Prc_{nt} price of product n at time t

PR_{nkat} production level of product n in manufacturing plant k on assembly line a at time t

pr_{nkat} Boolean variable that shows whether production of product n in manufacturing plant k on assembly line a at time t takes place

$up_{nkk''t}$ amount of demand of product n come to plant warehouse k from downstream node k'' at time t

$ud_{nkk''t}$ amount of demand of product n come to distribution center k from downstream node k'' at time t

$ur_{nkk''t}$ amount of demand of product n come to retailer k from downstream node k'' at time t

$yp_{nkk''t}$ amount of flow of product n from plant warehouse k to downstream node k'' at time t

$yd_{nkk''t}$ amount of flow of product n from distribution center k to downstream k'' at time t

$yr_{nkk''t}$ amount of flow of product n from retailer k to downstream k'' at time t

Parameters:

C_{RE} revenue generated by the sales of the products

C_{HO} cost of holding inventory in the SCN

C_{TR} cost of transportation in the SCN

C_{RM} cost of purchasing raw materials

C_{PF} fixed cost of production in the production facilities

C_{PV} variable cost of production in the production facilities

C_{CS} cost of customer satisfaction

$FC_{n,k}$ fixed cost of production of product n at manufacturing plant k

HC_{nkt} holding cost of product n in node k at time t

l_n production time of product n

m price of end product that makes the demand one

RC_n raw material cost of product n

$TC_{nkk''t}$ transfer cost of product n from node k to node k'' at time t

VC_{nk} variable cost of production of product n at manufacturing plant k

ur_{nk}^{U} maximum amount of product n that can be send from retailers

$ur_{nkk''t}^{old}$ amount of demand of product n come to retailer k from downstream node k'' at time t in the previous planning period

$\delta_{kk''}$ transportation delay between k and k''

R the revenue function

ur_{max} the demand value that makes the difference between values of R_a and R_r maximum

Z the total profit generated by the SCN

Chapter 4

Combinational Capacity Constrained Vehicle Routing Problem

Introduction:

Vehicle routing problem is one of the most important combinatorial problems studied in the literature. The vehicle routing problem (VRP) calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to make distribution or collection from given set of customers. Although the main logic of VRP is the distribution of goods to the set of customers, with respect to characteristics of the parameters given in the system, the problem can be changed and classified as a different version of VRP. These parameters are the following:

- The arcs (roads) can be directed or undirected
- Periods of days (time windows) during which the customers can be served (only for a specific time of period, the customer can be accessible).
- Whether time is required for loading or unloading the goods (may be depends on the type of vehicle.
- Specific subset of vehicles can be assigned to the specific customers.
- Satisfaction rate of customer demand.

- Existence of one or more depots
- Assignment of one or more routes to one customer.
- Vehicle capacity
- Existence of devices required loading and unloading goods from vehicles.
- Arc and vehicle assignment (some vehicles cannot use some arcs)
- Cost association (per unit, per vehicle, per route..)
- Driver utilizations constraints.

These are the some of the characteristics of the parameters associated with VRP that determines the classifications of the problem. Since the introduction of VRP by Dantzig and Ramser approximately half century is passed [25]. After them Clarke and Wright proposed an effective greedy approach that improves the Dantzig and Ramser's approach [56]. After these two introduction papers, many models, exact and heuristic methods have been introduced to literature. Desrocher, Lenstra and Savelsbergh extensively survey the classification scheme of the VRP problems [57]. Laporte and Nobert made an extensive survey for the exact methods for the VRP [58]. There are different classes of VRP problems, which are: Capacitated VRP, Distance Constrained VRP, VRP with time windows, VRP with Backhouls, and the VRP with Pickup and Delivery. However the VRP that we have studied during this study is a version of Capacitated VRP (CVRP). Therefore we first make an introduction to CVRP.

4.1. Capacitated Vehicle Routing Problem:

In CVRP, all the customers correspond to deliveries and the demands are deterministic. The vehicles are identical and there is single depot. Only the capacity constraints for vehicles are imposed. The objective is the minimization of the total cost (weighted function of number of routes, their lengths or travel time) to serve all the customers. This problem may be described with the following notations.

Let G=(V,A) be a complete graph, where V={0,..,N} is the vertex set and A is the arc set. Vertices from 1..N correspond to customers and the vertex 0 is the depot. There exist a non-negative cost c_{ij} associated with each arch (i, j) $\in$ A. If the G is directed graph then, the cost matrix will be asymmetric, otherwise $c_{ij} = c_{ji}$ and the corresponding matrix becomes symmetric. In addition for the case that we have studied the triangle inequality holds. Therefore,

$c_{ik} + c_{kj} \geq c_{ij}$ for all i, j, k $\in$ V.

In other words, direct link is the closest between two vertices, therefore it is convenient to use direct link instead of a route that includes more than 2 edges to reach from one vertex to another.

Each customer i (i=1…N) have known demand d_i. Because i=0 is the depot, its demand is equal to 0. There is total T amount of vehicle is available in the depot waiting for route assignment.

The CVRP consists of finding T circuits (route) start from depot and end in the depot. The objective is the minimization of the total cost such that:

- Each route visits the depot (start and ends at vertex 0)
- Each customer vertex is visited by exactly one route (this will be changed in the case that we have studied)
- The sum of the deliveries of the vertices visited by a route does not exceed the assigned vehicle capacity C (this will change with the changing condition 2).

The cost can be the total distance travelled or total vehicle used (number of routes). The CVRP is a generalization of the Travelling Salesman Problem (TSP). In the TSP, a single route (a Hamiltonian circuit with T=1) that traverses all of the vertices with minimum total distance travelled is searched. Therefore all the relaxation and constraints valid for TSP is valid for CVRP.

Ribeiro and Laport studied a new version of the capacitated VRP, which named as a Cumulative Capacitated Vehicle Routing Problem (CCVRP) [59]. The only difference between corresponding problem and ordinary VRP is the objectives of them. In ordinary VRP, the objective is the minimization of the total distance travelled, but for the case of the CCVRP; the objective is the minimization of the sum of arrival times at customers. To solve the model developed, a derivative of the neighborhood search heuristic is used.

Perboli et. al. studied the the capacited VRP for the case of the multi-echelon supply chain systems [60]. In this different kind of CVRP, instead of product is distributed from depot to customer directly, they are first carried and accumulated on the middle echelon of the system. After accumulation, they are distributed to the customers. The 2-echelon instance of the corresponding system is illustrated in the study. The heuristic developed to solve the system is detailly discussed.

Zhang and Tang studied a CVRP with one difference from the ordinary one: just before the distribution, a disruption occurs on the commodities distributed [61]. Because there is not enough commodities to distribute and because the missing commodities will arrive depot as soon as they are available, the fleet and the routes should be re-calculated and re-assigned. The main objective of this kind of CVRP is the minimization of the cost with respect to newly scheduled routes and fleets.

4.2 The Specific Version of the VRP - The CRVP with Combinational Capacity:

In this study, we have studied a different version of the CVRP. There are 2 differences from the standard CVRP, one in the capacity interpretation and the other is in the route interpretation.

In the standard VRP, both the capacity of vehicles C and the demand d is continuous. Therefore if vehicle carry a product, it is known that how many percent of capacity is used. However the version that we have studied in this study, capacity is combinatorial rather than continuous. Instead, capacity is determined by the combination of the products. For instance, consider 3 products, A, B and D that has to be loaded into vehicle with combinational constraints. For instance, 3 of A and 4 of B, 4 of A and 1 of D are the two combinations that a vehicle can carry at most. This placement can be seen in Fig. 4.1. Because in combinational placement, the capacity usage values are not used, sometimes a part of truck could not be utilized. For instance, there is an empty space in the second placement scheme represented in Fig. 4.1. The reason of this; some times even the smallest SKU cannot fit into the space remaining. The producer or carrier determines all this carriage specs or combinations. This kind of capacity definitions is very common in Finished Vehicle Industry, consumer goods industry and electrical and mechanical equipment etc. The automobile producers are generally produce 3-4 models and the size of each models are different from each other and instead of giving a capacity usage rate, they generally gives the combination of each models that a truck carry at most. On the reason of this is the carrier (trucks) type that is used in Finished Vehicle Industry. Because the trucks are special (hydraulic and includes special pallets in it), the continuous capacity constraints are not suitable for this industry.

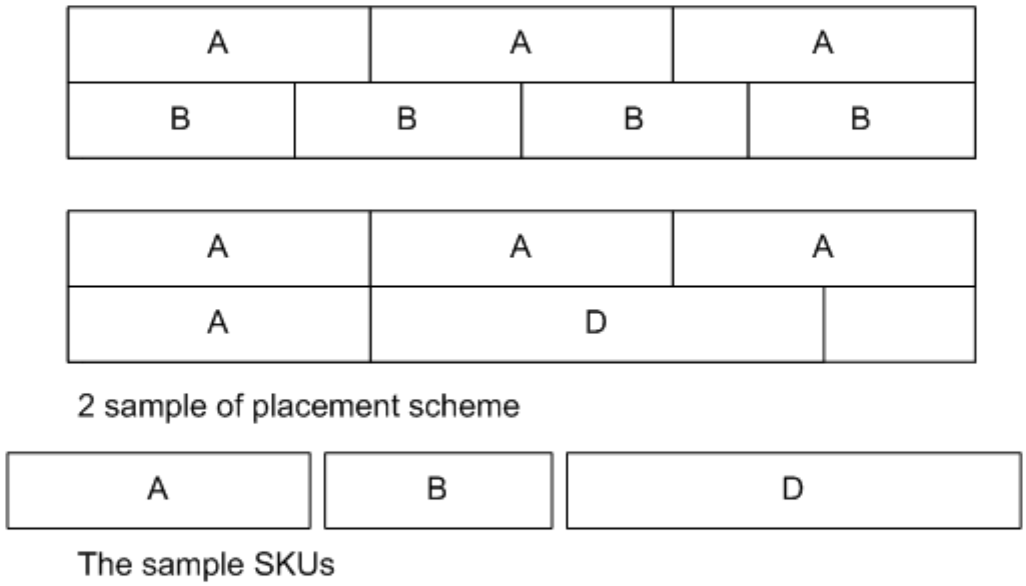


Figure 4.1: Sample combinational placement of SKUs.

Besides, in the standard CVRP the demand of each customer is satisfied by at most one vehicle (route-circuit). However in our case, more than one vehicle can satisfy the demand of a customer. The idea is represented in Fig. 4.2.The main reason of this is again the industry standards. Again for the case of the Industry of Finished Vehicle, this is a common practice. Due to size of finished vehicle, the total SKU (stock keeping unit) carried in a carrier is limited and generally it is impossible to for a truck to take total demand of one customer. For that reason in this industry, for example, 3 trucks or routes generally satisfy the total demand of the 2 customers.

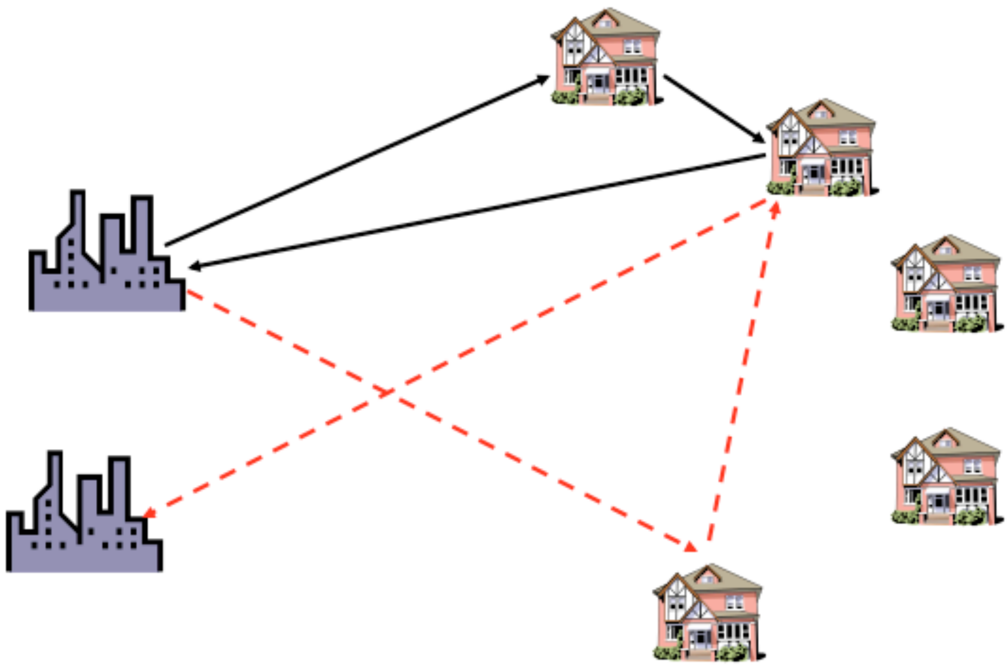


Figure 4.2: The sample distribution network.

4.3 Model:

A production firm is selected as a case study and data provider for the version of CVRP that is considered in this chapter. This firm is producing and distributing 3 different models of products with different sizes. Therefore different set of combinations of car carriage layout exists. These cars are carried by specific trucks and trailers. The trucks that carry cars are called as a low-bed truck whose height is less than conventional trucks. The reason of this is the bridges and the other road buildings that have specific heights. By decreasing the height of the truck the security of cars carried is increased and the set of feasible roads usable for the trucks are increased. Also the trailer that holds the cars are special tools too. A car cannot be placed randomly on the platforms in the trailer. For each model of product, the place and placement schema exists. Because of these constraints, specific combination of placement of products exists on the platforms in the trailer. These combinations for the firm we used as a case study are shown in the Table 4.1. Lets call the 3 models the firm is producing, model A, B and D. The A, B and D are the names of the models that are produced by the company.

	A	**B**	**D**
Combination 1	4	0	0
Combination 2	3	1	0
Combination 3	3	0	1
Combination 4	2	4	0
Combination 5	2	3	1
Combination 6	2	2	2
Combination 7	2	1	3
Combination 8	2	0	4
Combination 9	1	5	0
Combination 10	1	4	1
Combination 11	1	3	2
Combination 12	1	2	3
Combination 13	1	1	4
Combination 14	1	0	5
Combination 15	0	9	0
Combination 16	0	8	1
Combination 17	0	7	2
Combination 18	0	6	3

Combination 19	0	5	4
Combination 20	0	4	5
Combination 21	0	3	6
Combination 22	0	2	7
Combination 23	0	1	8
Combination 24	0	0	9

Table 4.1: Carriage combinations of the cars in the trailer.

According to Table-4.1, for example a carrier can carry 4 model A or 3 model A and 1 model B. We can have this information by looking at the first and second rows of the Table 4.1. These 24 combinations are the all of the combinations of the car carriages scheme. However it should be discussed before that to make good planning we should have as much as carriage combination. These cars are distributed to the all of the Turkey. One of the most important decision constraints in this system is the time to take a decision. Because shipment to the all of the Turkey is the common practice for this company and loading cars to the trucks require waste amount of time. Therefore, the planning of distribution and trucks routing should be done very quickly. According to company, quickly means in a 30 minutes or 1-hour at most. In other words the distribution and loading plans should be obtained 1 hour later than they are claimed. Therefore the model that optimizes the distribution system should be efficient. As known, VRP is one of the most difficult combinatorial optimization problems. Besides, the capacity constraints in our model is combinatorial too. Addition of this combinational capacity constraint makes the system computationally intractable. The objective of the model can be minimization of the trucks used or the maximization of the utilization of trailers. However for the case we have studied, the minimization of the transportation cost is chosen as an objective. The optimization model with respect to this objective is the following:

$$\min \sum_{i=0}^{N-1} \sum_{j=0}^{N+1} \sum_{k=1}^{T} x_{ijk} d_{ij} \quad (4.1.1)$$

$s.t$

$$\sum_{j=1}^{N+1} \sum_{k=1}^{T} x_{ijk} \geq 1 \quad \forall i \, [0 \ldots N] \quad (4.1.2)$$

$$\sum_{i=0}^{N} \sum_{k=1}^{T} x_{ijk} \geq 1 \quad \forall j \, [1 \ldots N] \quad (4.1.3)$$

$$\left(x_{ijk} + x_{mjk}\right) \leq 1 \quad \forall \, i, j, m \, [1 \ldots N] \, \forall k [1 \ldots T] \quad (4.1.4)$$

$$\sum_{j=1}^{N} x_{0jk} = 1 \quad \forall k \, [1 \ldots T] \quad (4.1.5)$$

$$\sum_{i=1}^{N} x_{i,N+1,k} = 1 \quad \forall k \, [1 \ldots T] \quad (4.1.6)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N} x_{ijk} + \sum_{\substack{m=1 \\ j \neq m}}^{N+1} x_{jmk} = 0 \quad \forall j \, [1 \ldots N], \forall k \, [0 \ldots T] \quad (4.1.7)$$

$$x_{ijk} - x_{jik} \leq 1 \quad \forall i \, [0 \ldots N+1], \forall j \, [0 \ldots N+1], \forall k \, [0 \ldots T], i < j \quad (4.1.8)$$

$$x_{ijk} = 0 \quad \forall i\ [0 \dots N+1], \forall j\ [0 \dots N+1], \forall k\ [0 \dots T], i = j \quad (4.1.9)$$

$$\sum_{i=1}^{N} x_{ijk}.C \geq s_{jkb} \quad \forall j\ [0 \dots N+1], \forall b\ [0 \dots B], \forall k\ [0 \dots T] \quad (4.1.10)$$

$$\sum_{k=0}^{T} s_{ikb} = d_{ib} \quad \forall i\ [0 \dots N+1], \forall b\ [0 \dots B] \quad (4.1.11)$$

$$\sum_{i=0}^{N+1} s_{ikb} = c_{kb} \quad \forall k\ [0 \dots T], \forall b\ [1 \dots B] \quad (4.1.12)$$

$$\sum_{j=0}^{N+1} x_{ijk} \leq z_{ik} \quad \forall i\ [0 \dots N+1], \forall k\ [0 \dots T] \quad (4.1.13)$$

The objective of the model is the minimization of the total distance travelled to distribute the products to meet the demand. In Eq. 4.1.2, it is assured that, every demand node should be visited and left by one route. According to Eq. 4.1.3, every demand node should be visited at least one route. By this way, more than one route can satisfy the demand of one node. With respect to Eq. 4.1.4, one route cannot come from 2 nodes to satisfy the demand of one node. According to Eq. 4.1.5 and 4.1.6, every route should start from depot and every route should end in the depot. Eq. 4.1.7 assured that, one route that visit a node other than depots, should leave this node. Eq. 4.1.8 and 4.1.9 are the basic feasibility constraints that represent the logic of routes. According to Eq. 4.1.10, if any demand of one node is satisfied, then the binary constraints that represent the flow to this node should be equal to 1. In Eq. 4.1.11, the demand of every node for each brand of product is satisfied. By using Eq. 4.1.12, the detailed carriage of each route is determined in terms of the brands of products. In Eq. 4.1.13, a new binary variable z_{jk} is introduced and if the route k visits node i, than the variable z_{jk} will be equal to 1.

However the model represented by Eq 4.1.1-13, is a part of the model responsible for just routing of vehicles and the capacity constraints are not included. As pointed before, the capacity of the trucks are not continuous. Instead of this, the capacity is represented by combination of the products that fill a truck, even though this carriage cannot be classified as Full Truck Load (FTL). Therefore, constraints that represent the combinational capacities should be added into the model. The constraint depending on the combination is given by,

$$\sum_{i=1}^{N} s_{ikb} \leq comb_{mb} y_{km} \quad \forall k\,[1 \dots T], \forall b\,[1 \dots B], \forall m\,[1 \dots M] \quad (4.1.14)$$

$$\sum_{m=1}^{M} y_{km} = 1 \quad \forall k\,[1 \dots T] \quad (4.1.15)$$

Adding the Eq. 4.1.14 and 4.1.15 and introducing the new binary variable y_{km} satisfy the combinational capacity constraints.

However as discussed before, the VRP problem is one of the most difficult combinatorial optimization problems even with continuous capacity constraints. By adding these combinatorial capacity constraints, the problem can be computationally intractable even for small instances. Therefore a new constrain elimination scheme is required to solve this problem. The details of method will be discussed in detail in the following section.

4.4. Solution Approach:

The model in Eq. 4.1 is a difficult MILP model and it can be solved exactly or approximately. In this study, the exact approach will be chosen as a solution technique and Branch and Cut method is used to solve it. However because, the model itself is difficult to solve, some techniques are developed to make the system computationally tractable. The details of the algorithm are shown in Fig 4.3.

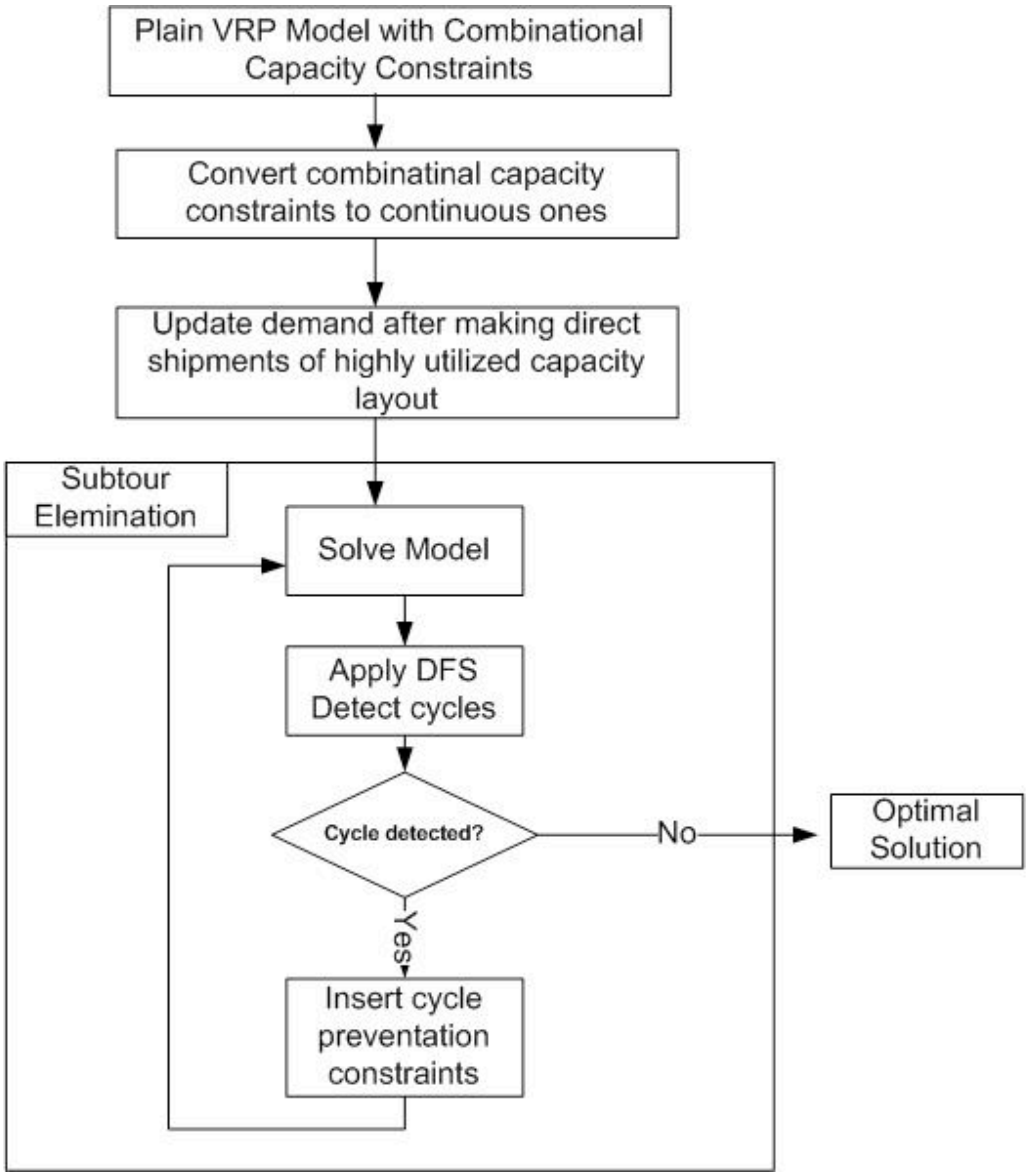


Figure 4.3: The VRP algorithm used for corresponding problem.

4.4.1. Constraint Elimination Scheme:

The model represented by Eq. 4.1. is an extended version of standard VRP model. There are constraints for routing of the trucks and there are constraints for the capacity limits. For the constraints of the routing, it is nearly impossible to change or relax constraints to obtain a computationally easy problem without disturbing the quality of the result. Therefore, in this study we are focusing on the constraints of the combinational capacities. As seen the in model represented by Eq. 4.1, a binary variable is introduced into the system for each tour and brand. The difficulty of our case is the combinatorial nature of the capacity constraints. Therefore to decrease the computational difficulty of the problem, instead of using combinational capacity constraints, continuous constraints should be used without disturbing the quality of the optimal solution. In this section, the method that converts the combinational constraints to continuous constraints is discussed in detail. To make this conversion the following linear model is developed.

$$\min Z \quad (4.2.1)$$

$$s.t.$$

$$\sum_{b=1}^{B} comb_{mb} v_b \leq C \quad \forall m\,[1 \dots M] \quad (4.2.2)$$

$$\sum_{b=1}^{B} comb_{mb} v_b + v_s \geq C \quad \forall m\,[1 \dots M] \quad (4.2.3)$$

$$Z \geq C - \sum_{b=1}^{B} comb_{mb} v_b \quad \forall m\ [1 \dots M] \quad (4.2.4)$$

The aim of the model represented by Eq. 4.2.1-4 is the finding capacity usage values for each model of product. With respect to Eq. 4.2.2, it assured that total capacity usage for each combination of the carriage should be less than total capacity. In Eq. 4.2.3, it is said that, the capacity usage of sum of the smallest product and each combination of carriage should be greater than the total capacity. By this way it is assured that, it is impossible carry even one more product with the combination of each carriage. The objective of this model is the minimization of the maximum space available for each combination of carriage. The success of this model is highly depends on the combinations given for the product layout in the truck. If all of the possible combinations are given, the model gives the applicable capacity usage values. If we do not have all of the combinations, the VRP model can still includes some of the combinational capacity constraints, but the total number of combinations the model included will certainly be decreased. In other words, if we have all of the combinations, then the developed method gives exact result. However on the lack of even 1 combination out of whole set of combination, the exactness of the method cannot be guaranteed.

In addition, the model presented in Eq. 4.2.1-4 cannot be solved optimally every time. Using all of the carriage constraints sometimes makes the model infeasible. Because sometimes it is impossible to find capacity usage values that satisfy all of the combinational constraints. Finding a unique number that shows capacity utilization of each model cannot be reachable every time. Therefore the following model is developed to determine which one of the combinational capacity constraints cannot be eliminated.

$$\min \sum_{m=1}^{M} y_m \quad (4.2.5)$$

$s.t.$

$$\sum_{b=1}^{B} comb_{mb} v_b \leq C + M y_m \quad \forall m\ (1 \dots M] \quad (4.2.6)$$

$$\sum_{b=1}^{B} comb_{mb} v_b + v_s \geq C - M y_m \quad \forall m\ (1 \dots M] \quad (4.2.7)$$

where $y \in \{0,1\}$.

After the model in Eq 4.2.5 - 7 is solved some of the values of *y* will be equal to 1. The combinations corresponds to those variables cannot be represented by unique capacity usage values. Therefore only the model represented by Eq. 4.2.5-7 is enough for the elimination of the combinational constraints. The variables in model are the v_m and y_m. Any combination whose y value is equal to 1 cannot be represented by the value of v_m. Those constraints should be included in the model by using the constraint 4.1.14. However for the remaining the combinational constraints can be represented as in Eq. 4.2.8 by using the v_m values that comes from Eq. 4.2.5-7.

After solving this model, the capacity usage for each model of product and the capacity of each truck is obtained. As soon as these capacity usage values are obtained, we do not need the combinational capacity constraints. Therefore instead of using the constraints Eq. 4.1-13 and 4.1-14 to represent the capacity of trucks, inclusion of Eq. 4.2.8 will be enough.

$$\sum_{i=1}^{N}\sum_{b=1}^{B} s_{ikb} v_b \leq C \quad \forall k\ [1 \dots T] \quad (4.2.8)$$

As seen in Eq. 4.2.8, unlike Eq. 4.1.13 and 4.1.14, it does not include any binary variables. By this way, the combinational capacity constraints are relaxed without disturbing the quality of the optimal solution. With this relaxation, the problem becomes more tractable. Because adding extra integer variables and combinatorial constraints make the VRP even harder. With this developed method, the extra difficulty occurred because of combinational constraints are removed. The computational results will be discussed in detail in the results section. Besides this relaxation, to make the problem more tractable future relaxation can be done. In the next section, the relaxation done by using the triangular in equality and combinational constraints are discussed in detail. By this way, some of the routes are eliminated from the VRP scheme.

As defined before, this method does not guarantee elimination of all capacity related binary variables from system. However as the number of the combinations given increases, without disturbing exactness of the solution, the number of eliminated binary variables increases and the problem becomes more computationally tractable. For all of the combinations given, the model represented by Eq. 4.2.5 – 7 is solved, and in the first look up, the combinations that can be represented without using integer variables are determined. If all of the combinations can be represented by one unique set of capacity utilization values, in other words all of the y_m values are zero in the first run, then without using any integer variables. However if in the first run, only some of the combinations can be eliminated, then in the second run the combinations that are determined by first run is used as a data set to the model represented by Eq. 4.2.5 – 7. This process continues like that till all of the combinations are assigned to one set for representation. Therefore we can accept that, the method decreases the number integer variables, and even some times we can model the capacity part of the VRP without using any integer variables.

This method does not disturb the optimality of the system. The usage reason of the Eq. 4.2.2 and 4.2.3 is guarantying the uniqueness of capacity utilization values of each product type. Therefore at the end of these constraints unique capacity utilization values of the each product that also satisfies the capacity constraints are obtained. However, if we have unique capacity utilization values, and these values do not disturb the capacity constraints, then we are assured that they will not disturb the optimality. In other words the new model whose formation is occurred after removal of the integer variables and addition of new linear constraints for capacity is equal to model with integer combinational constraints. These two models are equal to each other, and they are producing same result after optimization.

4.4.2. Route Elimination Scheme:

The aim of the VRP is the satisfaction of the demand of the nodes in the network by travelling minimum amount of distance. Therefore as the, as the demand that should be satisfied is increased, the difficulty of the problem also increases. One of the main reasons of this is the characteristic of the demand. As seen in the model represented by Eq. 4.1, the demand and the satisfaction of demand by each route is integer variables. The reason of this is the nature of the product that is carried. Because volume and the specs of the products are very different from each other, the amount of demand will be in the form of the integer values. Therefore the demand satisfaction and the product carriage values will also be integer. Any decrease in the dimension of these values during the routing process will affect the computational complexity of the problem dramatically. For instance, the satisfaction variable s_{ikb} is an integer variable with 3 index. The index i represents the demand node, k represents routes or the trucks and b is the brand of the products. If values of the one of this index decrease, the solution time of the problem will decrease. In this section, our main aim is developing a method that decreases these values. Too succeed this; the triangular condition of the distances is used. This means that if there is possibility of direct shipment, it is always better than the carriage of products with indirect routes. As much as direct shipments are done, the computational complexity of the routing process will decrease. The problem in this scheme is the shipment layout of trucks. In another words, one truck can be loaded in many combinations, but to not to disturb the optimality, which product layout should be chosen. To choose which of the layout should be chosen, it is decided that if one truck is traveling with the highest possible capacity utilization, then this is the solution closest to optimal one. Therefore, the following algorithm is applied to demand set:

For All Nodes

1. Chose the layout combination that has a highest rate of capacity utilization with respect to capacity usage values determined in Section 3.2.2.
2. Search for the demand nodes that have demand which is equal to layout determined in 1.
3. If a node is returned from 2, subtract the layout in section from the demand of selected node, terminate inner loop and try 1-2-3 for all nodes again.

End

Figure 4.4: Direct shipment algorithm

After applying this algorithm to the all of the demand nodes, if possible, some of them is eliminated from the routing system and this make the problem more tractable in terms of computational complexity.

Indeed the result of the algorithm is trivial. The layout with the highest occupancy rate is the, one with has a full of the smallest products. Therefore if there is a demand in one node with full of the smallest product, send this without adding this demand into the main VRP problem. By this way, number of one route will decrease. Also sending this layout with a truck without making any routing does not disturb the optimality if a direct shipment is done with a full capacity. In other words, there is not any other routing scheme for these products better. However if a layout other than full of smallest products is chosen for a direct shipment, the optimality can be disturbed. However if time limit to obtain a solution is very hard, in the expense of optimality, other layouts can be sent without explicitly routing them. This algorithm still gives feasible and acceptable solution.

The algorithm in Fig 4.4 does not guarantee the optimality in all conditions. As stated before, the total capacity utilization of each combination is found after capacity usage of each product type is determined in the previous part. In the previous part in addition to capacity utilization values of each product type, the minimum capacity value of a truck is also determined. With this derived values, some a combination can 100% utilizes the capacity or it can utilize less than 100%. However we know that, one more, even the smallest one, product cannot be carried in addition to given product combinations. Therefore it can be accepted that in combinatorial view, every combination is actually full truckload. In other words, one combination is not more full than the other. For that reason, the algorithm shown in Fig. 4.4 does not guarantee optimality. However in practice this method gives result, both in terms of optimality and computational time. Even if method does not guarantee the optimality, experiments shows that most of the time, this method gives optimal result. Also because the number of the routes optimized is decreased by this way, total computational time decrease. The details of the result are discussed in detail in the result section.

Only one more question is remaining for the VRP that is solved. After applying the model in Eq. 4.1, some cycles or sub-tours can be occurred. To eliminate these sub-tours a different scheme is applied to the model. The details of the sub-tour elimination scheme will be discussed in the following section.

4.5 Sub-tour Elimination:

The VRP model represented by Eq. 4.1 and Eq. 4.2 cannot prevent the occurrence of the cycles. In the VRP model, each tour should start at depot or factory and should end at factory too. Between depots the tours should not include any cycles. However the constraints sets do not ensure this. To prevent the occurrence of cycles (sub-tours) in a holistic way, exponential amount of constraints should be added into VRP model. However this method is not efficient, because the logic behind adding exponential amount of constraints is the insertion of all of constraints that prevent the occurrence of all possible sub-tours in the network [62]. However, in the most of the VRP problems, only small portion of the possible sub-tour candidates occurs. Therefore using all possible cycle prevention constraints only for small proportion of cycles is not effective. Instead of this, in this study we have used iterative cycle prevention scheme. Different from holistic approach, in the iterative approach, the cycle prevention constraints are added as far as, corresponding cycles occurred.

The sub-tour elimination scheme has 3 steps:

- Cycle detection
- Insertion of cycle prevention constraints

- Re-solution of the model.

4.5.1. Cycle Detection: Cycles are occurred once the plain VRP model is solved. After the solution of the model the model gives total K amount of routes. Some of these routes include cycle and some of them do not. To detect the cycles, Depth First Search (DFS) is applied each of the routes. After applying DFS to each route, if there exist a cycle, it is returned by DFS algorithm [63].

4.5.2. Insertion of Cycle Prevention Constraints: Suppose that in route *k* a cycle occurs. The nodes that the route *k* visits are : D-1-3-4-6-3-D, where D is the depot and the number shows the name of the nodes, the route visits in order. As seen from the route, there is a cycle between nodes 3-4-6-3. To prevent occurrence of this cycle, the following set of constraints are inserted into the system:

$x_{34k} + x_{46k} + x_{63k} \leq 3$ (4.3.1)

$x_{36k} + x_{64k} + x_{43k} \leq 3$ (4.3.2)

Addition of the Eq. 4.3, prevent occurrence of the link between the nodes that form cycle simultaneously. The assumption in here is that, if all of the edges take place in a route simultaneously, than they form a cycle, which is correct. Addition of Eq. 4.3 to the VRP model as a constraint prevents the occurrence of the corresponding cycle [64].

4.5.3. Re-Solution of Model:

As all of the cycles are detected and all of the cycle prevention equations are created, these equations are inserted into the VRP model as a constraint, and it is solved again. This 3 steps process is iteratively applied to the model until no cycle is detected.

4.5.4 The VRP Algorithm:

The algorithm is demonstrated in the Fig. 4.3. In this figure, all of the steps of the algorithm can be seen. At the first steps we have only the problem with the parameters.

With respect to these parameters, first VRP model is formed. However this model includes all of the demand without post-process and the capacity constraints are written in terms of the combination of the layouts of trucks. First the method that is developed to convert combinational constraints to continuous constraints. During the application of this method, the capacity usage of the each product is determined. Then to decrease the computational complexity problem, if direct shipment can be done from depot to any node with the full truck of highly utilized layout combination, it is done, and the product sent is subtracted from total demand. Then with the remaining demand and the continuous capacity constraints the first VRP model is solved. However, because any constraint that prevents the occurrence of the cycle is not inserted into the plain VRP model, occurrence of cycle is probable. If any of them is detected, the corresponding cycle prevention constraint is added into system and it is solved again. This process goes until no cycle is detected. In the occurrence of no cycle, the solution obtained is the optimal solution.

4.6 Results:

After the model and the algorithm are developed, it is applied to data set that is provided from real life example. As said before, because of the nature of the sector, the distribution or shipments are made twice a day and sometimes more. Therefore, the result of the VRP model should be obtained in at most 1 hour. This one-hour limit is informed by the sector managers. The result are obtained after the optimization of an instance of real life data . In this case, total 3 different products are distributed to the 81 regions of Turkey in everyday. The demand is totally random, but real (taken from real customer demand). Total demand of 30 days is taken and runs are made for these 30 days. These 30 days are a period of peak season. With respect to this data, the results are shown in Table 4.2.

Data Set	VRP	VRP + Cont. Const	VRP+Cont. Const.+Dir. Ship.
D1	29%	9%	3%
D2	21%	9%	3%
D3	32%	11%	4%
D4	19%	8%	2%
D5	24%	7%	1%
D6	21%	12%	1%
D7	21%	10%	3%
D8	28%	9%	3%
D9	26%	9%	4%
D10	27%	9%	1%
D11	34%	11%	1%
D12	18%	12%	3%
D13	22%	7%	4%
D14	25%	9%	3%
D15	21%	9%	2%
D16	25%	13%	6%
D17	17%	9%	2%
D18	26%	7%	2%
D19	27%	9%	3%
D20	22%	12%	1%
D21	19%	12%	4%

D22	21%	11%	9%
D23	21%	9%	1%
D24	19%	11%	3%
D25	36%	9%	3%
D26	29%	9%	2%
D27	54%	7%	3%
D28	19%	8%	5%
D29	27%	12%	4%
D30	24%	9%	3%

Table 4.2: Results of run in terms of optimality gap after 1h run time

In the Table 4.2, the optimality gap of the model after 1 hour passed is shown of for the data of 30 days. In the first row, the optimality gap for the plain VRP model is shown, in the second column, optimality gap after the combinational capacity constraints are converted to continuous ones is shown. In the third column the results after both capacity is converted to continuous and direct shipments are done; is shown. As seen from the values, both method developed to decrease the run time is working. We can derive this result from the average values of each column. The average of the first column is 25% that of second column is 10%, and that of third column is 3%. By looking at these values, it can be said that, with the development and usage of these 2 methods, the intractable VRP problem becomes more tractable.

Also it should be reported again that after direct shipment, the optimality is not guaranteed. However for the 30 instances given in this thesis, the solutions obtained after direct shipment are still optimal. These optimality shows that, the method shows empirically good results in terms of solution quality.

In the previous experimental study, the effect of the algorithm for limited time span is shown. However, these results cannot show the exact difficulty of the problems. For the second trial, to show the exact difficulty of the problem and success of the algorithms developed the solution time of 7 instances will be reported. These 7 instances are selected randomly out of 30 instances give. The solution times of plain problems and solution times after application of each algorithm are reported in the Table 4.3.

Data Set	VRP	VRP + Cont. Const	VRP+Cont. Const.+Dir. Ship.
D1	9.4h	4.2h	3.5h
D2	8.1h	5.1h	2.0h
D3	12.h	4.h	1.1h
D4	7.8h	4.7h	3.1h
D5	7.9h	3.5h	1.9h
D6	8.3h	3.4h	2.1h
D7	9.1h	3.3h	1.7h

Table 4.3: Solution times of the problems

As seen from the Table 4.3, the solution time of random plain VRP problems are quite high. However after application of each algorithm the solution times are decreased. In some data set, the decrease is not excessive. The reason of this is the characteristics of the problem and data set. Because there is not a shipment to the all of the Turkey everyday, and some times the shipment car types are heterogeneous. In those case, the run times can be higher. However by just looking at the solution times, it can be said that the algorithms developed are quite useful to decrease the solution time. However out of the solution algorithms, the problem itself is also contribution because the combinationaly constrained VRP has never been studied in the literature.

4.7 Summary:

The vehicle routing problem (VRP) is one of the most difficult combinatorial problems. Besides, its application can be very frequently seen in the real life. In this study we have developed model and solution algorithm for the specific version of the VRP. In this version both demand and capacity usage is integer and the capacities of the trucks are represented the combination of products that a truck can carry at most. Because of these two factors, this specific version of the VRP becomes even more computationally difficult. To deal with this difficulty first a method that converts combinational constraints to continuous ones is developed. Then the direct shipments are done because of the validity of the triangle inequality and the dimension of demand is decreased. After these two changes, the model becomes more tractable in its application area.

4.8 Nomenclature:

G…	The graph
V…	Vertices of a graph
A…	Edges of a graph
N…	Number of vertices or that of demand node
CVRP…	Capacitated vehicle routing problem
C…	Capacity of truck
A,B,D…	The brands of products

SKU…	Stock keeping unit
Xijk…	binary variable that shows whether route k goes from node i to node j.
T…	number of routes
cij…	cost of transportation from node I to node j.
dib…	demand of brand b requested by node i.
B…	number of brands
i,j,m…	indices for nodes
k…	indices for route
sjkb…	the satisfaction amount of brand b of node j by route k.
crkb…	amount of brand b carried by route k.
zjk…	binary variable shows whether route k visits node j.
combmb…	number of brand b exists in the combinational carriage constraint
M…	number combinational carriage constraint.
ykm…	binary variables shows whether the route k carries the load of combinational carriage constraint.
vb…	capacity usage of brand b.
vs…	capacity usage of smallest brand (can be assumed 1 as a default)
L…	a large number

Chapter 5

CONCLUSION

The main focus of this the supply chain management problems or the supply chain network problems. We used optimization approach for the supply chain modeling and management problem. Because the supply chain context is very broad, we divided it into 3 parts: Modeling, solution and applications.

In the modeling part a new modeling technique is developed. With this new modeling approach the system topology is considered as data and it is separated from the model to make the model leaner, more modular, understandable and extendable. In the solution part we have focused on the batch production constraints and their non-integer representations. To succeed this a bi-level optimization problem solver algorithm is developed by using the law of strong duality. Also a linear programming model is developed to solve modular arithmetic problems. Also the mixed integer problem is converted into bi-linear model. The bi-linear model is solved by using approximations techniques. Then finally an original hybrid genetic algorithm based algorithm is developed to solve the bi-linear system. In the application part, a unique Vehicle Routing Problem, VRP with combinational capacity constraint is developed. This problem is more computationally complex and difficult than standard VRP. To make the problem computationally tractable a new exact algorithm is developed. This algorithm converts combinatorial combinational capacity constraint to the continuous ones without disturbing the optimality. Then to decrease the run time more an approximation rule is developed. Even if this rule does not guarantee the optimality, the empirical results have a high quality in terms of the optimality.

In the modeling part, a new modeling methodology is developed and it is applied to sample supply chain system. In the conventional modeling practices, for each node and for each flow and even sometimes for each time step different sets of constraints are used. These constraints are generally used to model the topology of the system or network. However, in the best practice of the modeling, we know that data and model should be separated from each other. In our idea, the topology is also data, therefore model of different topologies should not be different. Only the data part of them should be different. However they should have same model.

To separate the topology of the model, or to separate topology related behavior from the model, we have used stoichiometric matrix to hold the properties of the system topology. In conventional supply chain, the behavior of the system is at least depending on the supply chain state and the rate of change of state of each entity of the system. Therefore at first a 2-dimensional matrix is used to hold the system topology. However some times more parameters are added into system to determine the behavior, such as time. If that is the case, a more dimensional matrix is used to capture topology.

The reason of use of this modeling technique is the applying the best practice of modeling methodology. In the best practice, the model and data should be separated so-called model-data independence. The reason of this independence is to make the model maintenance, re-modeling and model-to-model integration easier. Because with this new modeling technique, without changing the model, by just manipulating data set, a new extended model can be obtained.

For the solution part, we are interested in the solution strategies in supply chain management models. The model that we are focused on is basically multi-echelon, multi-product supply chain network with batch production and procurement constraints. First of all we try to solve the system without using integer variables to represent the batch production and procurement behaviors. First of all, by using the low of strong duality we modeled 2 level and 3 level systems without integer variables. We solved the 2 level system. As a result of it, we obtained linear time solution of modular arithmetic and result of the step-wise system used in the representation of the batch production. However to get full scale supply chain model we have to superimpose this two level system into the main supply chain model which we called macro model. After this superimposition we obtained a 3 level system without integer variables, but this model still behaves in the same way as the MIP model behaves. To solve 3 level system we have applied the same method that is derived from low of the duality. However because of the lack of the dynamic behavior of primal-dual relationship, we get premature result.

Than the integer model is converted into the bi-linear system. This bi-linear system gives the same result with the MIP system. However bi-linear systems are non-convex and there is not an easy way of solving them. Therefore first of all we applied existing commercial solvers (Baron-Dicopt) to the bi-linear system to see whether there is an advantage of passing from MIP domain to bi-linear domain. However it is seen that the solvers for MIP is more successful than non-linear solvers.

After underestimator schemes are developed to solve the bi-linear system. Because the underestimators are the branch of the approximation algorithms and the results are highly dependent on the bounds of the variables, good enough results cannot be obtained.

As a result a hybrid genetic algorithm is developed and applied to the bi-linear system. Even if still the MIP solver (Cplex) is the best, the heuristic algorithm becomes more successful than the commercial non-linear solvers.

For the application part, we have studied on a unique Vehicle Routing Problem. The vehicle routing problem (VRP) is one of the most difficult combinatorial problems. Besides, its application can be very frequently seen in the real life. In this study we have developed model and solution algorithm for the specific version of the VRP. In this version both demand and capacity usage is integer and the capacities of the trucks are represented the combination of products that a truck can carry at most. Because of these two factors, this specific version of the VRP becomes even more computationally difficult. To deal with this difficulty first a method that converts combinational constraints to continuous ones is developed. Then the direct shipments are done because of the validity of the triangle inequality and the dimension of demand is decreased. After these two changes, the model becomes more tractable in its application area.

BIBLIOGRAPHY

[1] Meixell, M.J., Gargeya, V.B. , "Global supply chain design: a literature review and critique", Transportation Research Part E, 41-6 (2005), pp.531- 550

[2] Mestan E., Turkay M., Arkun Y. Optimization of supply chain systems using hybrid system approach and model predictive control, Ind. Eng. Chem. Res. 45 (2006) 6493-6503

[3] Perea E., Grossmann I. E., Ydstie E., Tahmassebi T., Dynamic modeling and classical control theory for supply chain management. Computers and Chemical Engineering. 24 (2000) 1143-1149.

[4] Fleischemann M., Bloemhof-Ruwaard, J. M., Dekker, R., Van der Laan E., Van Nunen, A.E.E, Van Wassenhove, L.N. Quantitative models for reverse logistics: A review. European Journal of Operational Research (1997) 103, 1-17

[5] Pohlen, T.L. Farris, H.M. Reverse logistics in plastics recycling. International Journal of Physical Distribution & Logistics Management. (1992) 22(7), 35-47.

[6] Krikke, H., Recovery strategies and reverse logistics network design. Thesis, University of Twente, Enschede, Netherland (1998).

[7] Thierry, M., Salomon, M. Nunen, J., Wassenhove, L. Strategic Issues in Product Recovery Management. California Management Review. (1995) 37(2), 114-135.

[8] Rogers D.S., Tibben-Lembke R.S. Going Backwars: Reverse logistics trends and practices., Reverse Logistics Executive Council. (1998)

[9] Daniel, V., Vaidyanatanhan Jayarama, Jr., Srivastava, R. Benton, W.C , Supply-Chain management for recoverable manufacturing systems. Interface. (2000) 30:3(125-142)

[10] Lund, R.T. Remanufacturing. Technology Review.(1984) 87(2) 19-29.

[11] Law A. M., Simulation Modeling and Analysis, Mcgraw Hill Higher Education (2006).

[12] Nahmias S. Production and Operations Analysis. McGraw-Hill. (2001)

[13] Goetschalckx M., Vidal C. J., Dogan K., Modeling and Design of Global Logistics Systems: A Review of Integrated Strategic and Tactical Models and Design Algorithms, Europen Journal of Operational Research. (2002) 143-1

[14] Grossmann I. Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. AIChe Journal. (2005), 51-7

[15] Mele F.D., Espuna A., Puigjaner L., Supply Chain Management Through Dynamic Model Parameters Optimization. Ind. Eng. Chem. Res. (2006) 45, 1708-1721

[16] Dawlatshahi, S., Developing a theory of Reverse Logistics. Interfaces. (2000) 30-, 143-155.

[17] Gabaix X, Laibson D, Guillermo M., Weinberg S., Costly Information Acquisition: Experimental Analysis of a Boundedly Rational Model. The American Economic Review. (2006) 96:4, 1043-1068

[18] Shapiro J. S., Modeling the Supply Chain. Duxbury Thomson Learning (2001)

[19] Hiller F. S., Lieberman G. J., Introduction to Operations Research, McGraw Hill (2005)

[20] Biegler L. T., Grossmann I.E., Retrospective on Optimization. Computers & Chemical Engineering (2004) 28:8, 1169-1192

[21] Welch W. J., Algorithmic Complexity: Three NP-Hard Problems in Computational Statistics. Journal of Statistical Computation and Simulation.(1982) 15:1

[22] Croom S., Romano P., Giannakis M., Supply Chain Management: An Analytical Framework for Critical Literature Review. European Journal of Purchasing and Supply Management.(2001). 6:1, 67-83.

[23] Stadtler H. Supply Chain Management and Advanced Planning- Basics, Overview and Challenges. European Journal of Operations Research.(2005) 163:3 575-588

[24] Christofides N., Eilon S., An Algorithm for the Vehicle-Dispatching Problem. Operations Research. (1969) 20:3. 309-318.

[25] Dantzig G.B., Ramser J.H. The Truck Dispatching Problem. Management Science (1959) 6:80.

[26] Larman C. Applying UML and Patterns. Prentice Hall NJ. (2004)

[27] Liberti L., Pantelides C.C. An Exact Reformulation Algorithm for Large Nonconvex NLPs Involving Bilinear Terms. Journal of Global Optimization.(2006) 36. 161-189

[28] Bergamini M. L., Grossmann I., Scenna N., Aguirre P. An Improved Piecewise Outer-approximation Algorithm for the Global Optimization of MINLP Models Involving Concave and Bilinear Terms. Computers and Chemical Engineering. (2008) 32, 477-493

[29] Lee K. S., Geem Z.W. A New Meta-heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice, Computer Methods in Applied Mechanics and Enginering (2004) 194:36, 3902-3933.

[30] ILOG Cplex Reference Manual and Concert Technology User Manual (2010)

[31] Vollman T.E., Berry W. L., Whybark D.C. Manufacturing Planning and Control Systems. Newyork.(1997)

[32] Zipkin P. Z. Models for Design and Control of Stochastic, Multi-Item Batch Production Systems. Operations Research (1984) 34:1, 91-104

[33] Tomlin J. A., An Improved Branch and Bound Method for Integer Programming. Scientific Control Systems Ltd. Technical Report (1969)

[34] Bracken J., McGill J. T., Mathematical Programs with Optimization Problems in the Constraints. Operations Research (1973) 21:1, 37-44.

[35] Bracken J., McGill J. T., Defense Mathematical Programs with Optimization Problems in the Constraints. Operations Research(1974) 22:5 1086-1096.

[36] Bracken J., McGill J. T., A Method for Solving Mathematical Programs with Nonlinear Programs in Constraints. Operations Research (1973) 22:5 1097-1101.

[37] Jeroslow R. G., The Polynomial Hierarchy and a Simple Model for Competitive Analysis. Mathematical Programming(1985) 32:2, 146-164.

[38] Gams-BARON Reference Manual

[39] Gams-Dicopt Reference Manula

[40] McCormic, G.P. , 1976. Computability of global solutions to factorable nonconvex programs: Part 1-Convex Underestimating Problems. Mathematical Programming, 10, 147-175.

[41] Duran M. A., Grossmann I. E., A Mixed-integer Nonlinear Programming Algorithm for Process System Synthesis. AIChE Journal (2004) 32:4, 592-606.

[42] Yang Y.W., Xu J. F., Soh C. K., An evolutionary programming algorithm for continuous global optimization, European Journal of Operational Research 168 (2006) 354-369

[43] Hedar A., Fukushima M., Tabu Search directed by direct search methods for nonlinear global optimization, European Journal of Operational Research 170 (2006) 329-349.

[44] Nelder J. A., Mead R., A Simplex Method for Function Minimization. Computer Journal.(1965) 7, 308-313.

[45] Miettinen K, Makela M. M., Maaranen H., Efficient hybrid methods for global continuous optimization based on simulated annealing, Computers & Operations Research 33 (2006) 1102-1116.

[46] Kumar C., Combining Mutation Operators in Evolutionary Programming, IEEE Transactions on Evolutionary Programming Vol:2, No:3, September 1998.

[47] Hedar A., Fukushima M., Tabu Search directed by direct search methods for nonlinear global optimization, European Journal of Operational Research 170 (2006) 329-349.

[48] Chopra S., Meindl P., Supply Chain Management: Strategy, Planning, and Operations. (2003) Prentice Hall.

[49] Perea-Lopez E., Ydstie E., Grossmann I.E., A Model Predictive Control Strategy for Supply Chain Optimization. Computers and Chemical Engineering (2003) 27:1, 1201-1218

[50] Weiss M. A., Data Structures and Problem Solving using Java. Addison Wesly(2002)

[51] Yimer A. D., Demirli K., A Genetic Approach to two-phase Optimization of Dynamic Supply Chain Scheduling. Computers and Industrial Enginnering. 58 (2002) 411-422

[52] Chatfield D. C., Harrison T. P., Hayya J. C., SISCO: An Object-Oriented Supply Chain Simulation System. Decisions Support Systems. 42 (2006) 422-434

[53] B. Pallson, Systems Biology: Properties of reconstructed networks. Cambridge University Press, New York, 2006.

[54] A. U. Raghunathan, J. R. Perez-Correa, E. Agosin, L. T. Biegler, Parameter estimation in metabolic flux balance models for batch fermentation-formulation & solution using differential variational inequalities. Annals of Operations Research. 148 (2006) 251-270.

[55] C. Martabit, C. Varela, J. Sainz, E. Agosin, A model combining sugar uptake kinetics and metabolic networks for predicting wine fermentations technical report, Department of Chemical and Bioprocess Engineering, Pontifica Universiad Catolica de Chile .(2004)

[56] Clarke, G., J.W. Wright., Scheduling of vehicles from a central depot to a number of Delivery Points. Operations Research. (1964) 12, 568-581.

[57] Desrochers, M., J.K. Lenstra, M.W.P. Savelsbergh A classification scheme for vehicle routing and scheduling problems. European Journal of Operational Research (1990) 46, 322-332.

[58] Laporte G., and Y. Nobert, Exact algorithms for the vehicle routing problem, in S. Martello, G. Laporte, M. Minoux and C. Riberiro(eds), Survey in Combinatorial Optimization, North-Holland, Amsterdam, (1987) 147-184.

[59] Ribeiro G. M., Laporte G., An Adaptive Large Neighborhood Search Heuristic for the Cumulative Capacitated Vehicle Routing Problem. (2011) CIRRELT-2011-02.

[60] Perboli G., Tadei R., Vigo D., The Two-Echelon Capacitated Vehicle Routing Problem: Models and Math-Based Heuristics.(2008), CIRRELT -2008-55

[61] Zhang, X. , Tang, L., Disruption management for the vehicle routing problem with time windows. In D.S. Huang, L. Heutte and M. Loog. Eds. Advanced Intelligent Computing Techniques. (2007)

[62] Ahuja R. K., T.L Magnanti, J.B. Orlin. Network Flows. Prentice Hall, Upper Saddle River, New-Jersey (1993).

[63] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., Introduction to Algorithms. Introduction to Algorithms, MIT Press, Cambridge, Massachusetts. (2003).

[64] Nemhauser G.L, L.A. Wolsey. Integer and Combinatorial Optimization. Wiley, Newyork. (1988)

[65] Solomon M., M.W.P. Savelsbergh A brach-and-price algorithm for the pickup and delivery problem with the time window. (1994) Memorandum COSOR 94-22.

[66] Dantzig, G.B, Fulkerson and Johnson. On linear programming combinatorial approach to the traveling salesman problem. (1959) JORSA, (7) 1.

[67] Bard J.F., G. Kontoravdis, G. Yu. Branch and Cut for the Vehicle Routing Problem. Transportation Science. (2002) 36:2. 250-269

VITA

Ugur Kaplan was born in Elazig, Turkey, on September 9, 1982. He graduated from Malatya Fen Lisesi in 2000. He received his BS from in both Industrial Engineering and Computer Engineering from Koc University, Istanbul in 2005. He received his MS degree in Industrial Engineering on 2007. Same year he joined the PhD program in Industrial Engineering and Operations Management at Koc University.