LOT SIZING WITH NONLINEAR PRODUCTION COST FUNCTIONS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

By
Esra Koca
July, 2015

LOT SIZING WITH NONLINEAR PRODUCTION COST FUNCTIONS
By Esra Koca
July, 2015

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Prof. Hande Yaman(Advisor)

Prof. M. Selim Aktürk(Co-Advisor)

Assoc. Prof. Oya Ekin Karaşan

Assoc. Prof. Sinan Gürel

Prof. Haldun Süral

Assist. Prof. Özlem Çavuş

Approved for the Graduate School of Engineering and Science:

Prof. Levent Onural
Director of the Graduate School

ABSTRACT

LOT SIZING WITH NONLINEAR PRODUCTION COST FUNCTIONS

Esra Koca
Ph.D. in Industrial Engineering
Advisor: Prof. Hande Yaman
Co-Advisor: Prof. M. Selim Aktürk
July, 2015

In this study, we consider different variations of the lot sizing problem encountered in many real life production, procurement and transportation systems. First, we consider the deterministic lot sizing problem with piecewise concave production cost functions. A piecewise concave function can represent quantity discounts, subcontracting, overloading, minimum order quantities, and capacities. Computational complexity of this problem was an open question in the literature. We develop a dynamic programming (DP) algorithm to solve the problem and show that the problem is polynomially solvable when number of breakpoints of the production cost function is fixed and the breakpoints are time-invariant. We observe that the time complexity of our algorithm is as good as the complexity of existing algorithms in the literature for the special cases with capacities, minimum order quantities, and subcontracting. Our algorithm performs quite well for small and medium sized instances. For larger instances, we propose a DP based heuristic to find a good quality solution in reasonable time.

Next, we consider the stochastic lot sizing problem with controllable processing times where processing times can be reduced in return for extra compression cost. We assume that the compression cost function is a convex function in order to reflect the increasing marginal cost of larger reductions in processing times. We formulate the problem as a second-order cone mixed integer program, strengthen the formulation and solve it by a commercial solver. Moreover, we obtain some convex hull and computational complexity results. We conduct an extensive computational study to see the effect of controllable processing times in solution quality and observe that even with small reductions in processing times, it is possible to obtain a less costly production plan.

As a final problem, we study the multistage stochastic lot sizing problem with nervousness considerations and controllable processing times. System nervousness is one of the main problems of dynamic solution strategies developed for stochastic lot sizing problems. We formulate the problem so that the nervousness of the system is restricted by some additional constraints and parameters. Mixing and continuous mixing set structures are observed as relaxations of our formulation. We develop valid inequalities for the problem based on these structures and computationally test these inequalities.

Keywords: Lot sizing, piecewise concave cost function, convex cost function, controllable processing times, nervousness.

ÖZET

DOĞRUSAL OLMAYAN ÜRETİM MALİYETİ FONKSİYONLARI OLAN KAFİLE BÜYÜKLÜĞÜ PROBLEMİ

Esra Koca
Endüstri Mühendisliği, Doktora
Tez Danışmanı: Prof. Dr. Hande Yaman
İkinci Tez Danışmanı: Prof. Dr. M. Selim Aktürk
Temmuz, 2015

Kafile büyüklüğü problemi gerçek hayattaki birçok üretim, tedarik ve taşımacılık sisteminde karşılaşılabilen bir problemdir. Bu çalışmada, kafile büyüklüğü probleminin farklı versiyonları incelenmiştir. Öncelikle, üretim maliyetleri parçalı içbükey fonksiyon olan deterministik kafile büyüklüğü problemi çalışılmıştır. Parçalı içbükey bir fonksiyon ile indirimler, taşeron kullanma, aşırı yükleme, minimum üretim kısıtları ve kapasiteler modellenebilir. Parçalı içbükey üretim maliyeti fonksiyonu olan kafile büyüklüğü probleminin hesaplama karmaşıklığına dair literatürde herhangi bir sonuç yoktu. Bu çalışmada, bu problemin çözümü için bir devingen programlama (DP) algoritması geliştirilmiştir. Üretim maliyeti fonksiyonunun kırılma noktaları sayısı sabit ve kırılma noktaları dönemlerden bağımsız iken, problemin polinom sürede çözülebildiği gösterilmiştir. Problemin bazı özel durumları (kapasiteli, minimum üretim kısıtlı ve taşeron kullanılan gibi) için DP algoritmasının zaman karmaşıklığının literatürdekiler kadar iyi olduğu görülmüştür. Algoritmamız küçük ve orta büyüklükteki problem örnekleri için kabul edilebilir sürelerde çözüm vermektedir. Büyük problem örnekleri için ise DP algoritmasından faydalanılarak sezgisel bir çözüm yöntemi geliştirilmiştir.

Daha sonra, işlem süreleri belirli bir maliyet karşılığında (azaltma maliyeti) azaltılabilen rassal kafile büyüklüğü problemi üzerine çalışılmıştır. Azaltma maliyetinin dışbükey bir fonksiyon olduğu varsayılmıştır. Bu problem, ikinci dereceden konik karmaşık tamsayılı program olarak formüle edilmiş, formülasyon güçlendirilmiş ve bir çözücü ile çözülmüştür. Ayrıca, bazı dışbükey örtü ve hesaplama karmaşıklığı sonuçları elde edilmiştir. Kapsamlı bir sayısal çalışma ile işlem sürelerinin azaltılmasının sonuçlar üzerindeki etkisi incelenmiş ve işlem sürelerindeki çok az bir değişikliğin dahi daha iyi sonuçlar elde edilmesini sağladığı görülmüştür.

Son olarak, sistem gerginliğinin göz önünde bulundurulduğu çok aşamalı rassal kafile büyüklüğü problemi üzerine çalışılmıştır. Sistem gerginliği, rassal kafile büyüklüğü problemleri için geliştirilen dinamik çözüm yöntemlerinin ana problemlerinden birisidir. Bu çalışmada, çok aşamalı rassal kafile büyüklüğü problemi için daha az sistem gerginliğine neden olacak bir formülasyon geliştirilmiş ve bu formülasyonun bazı karma set yapılarını gevşetme olarak içerdiği görülmüştür. Bu yapılar göz önünde bulundurularak formülasyon için geçerli eşitsizlikler önerilmiş ve bu eşitsizlikler sayısal olarak test edilmiştir.

Anahtar sözcükler: Kafile büyüklügü problemi, parçalı içbükey maliyet fonksiyonu, dışbükey maliyet fonksiyonu, kontrol edilebilir işlem süreleri, sistem gerginliği.

Acknowledgement

I would like to express my deepest gratitute to my advisors Prof. Hande Yaman and Prof. M. Selim Aktürk for their invaluable guidance, support, patience and encouragement during my graduate study. I am extremely lucky to have such great advisors and mentors.

I am grateful to Assoc. Prof. Oya Ekin Karaşan and Assoc. Prof. Sinan Gürel for devoting their valuable time to read each part of my work and providing precious suggestions. I also want to express my gratitude to Prof. Haldun Süral and Asst. Prof. Özlem Çavuş for accepting to be a member of my examination committee and for their valuable suggestions.

I would like to acknowledge that this research is supported by grant 112M220 of Program 1001 of TUBITAK, The Scientific and Technological Research Council of Turkey.

It was an honor to be a member of Bilkent University Department of Industrial Engineering, and I would like to thank each member of the department.

My friends Ece Demirci, Burak Paç, Gizem Özbaygın, Hatice Çalık, Merve Meraklı, Nihal Berktaş, Ramez Kian and Sinan Bayraktar deserve special thanks for their friendship and support. I want to thank each of them for making my life more beautiful. They are like family, and distances do not matter for families. I keep my special thanks to İrfan Mahmutoğulları, Okan Dükkancı and Kamyar Kargar for being such nice friends. I feel very lucky to have so many great people around me. Life would be cheerless and gloomy without them.

Finally, I would like to express my deepest gratitude to my family for their endless love and trust.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Lot sizing problems arise in production, procurement and transportation systems under different cost and capacity settings. Given a planning horizon, demand, production (or procurement/shipment) and inventory holding costs, the aim of the lot sizing problem is to propose a minimum cost production plan to satisfy the demand. This problem is applicable to many industries like injection molding [1], paper production [2], textile industry [3], tire production [4], bottling [5], etc.

In production planning context, lot sizing is a medium term planning process which directly affects the system performance. Thus, it is very important to make the right lot sizing decisions for a manufacturing firm to be competitive in the market [6].

Single item lot sizing problem is one of the most important and most studied versions of the lot sizing problem since it is a sub-problem of more “difficult” lot sizing problems and solution methods developed for this problem can be used to solve more complex problems [7].

In this thesis, three lot sizing problems are considered. First, in Chapter 3, we study the lot sizing problem where the inventory holding cost function is concave and the production cost function is a piecewise concave function. A piecewise concave function can be used to represent quantity discounts, subcontracting, overloading, minimum order quantities and capacities. Moreover, one can use a piecewise concave function to represent combinations of these options. Although lot sizing problem is studied since the seminal paper of Wagner and Whitin [8], computational complexity of this problem was an open question in the literature. We first develop a dynamic programming (DP) algorithm to solve the problem and show that the problem is polynomially solvable when number of breakpoints of the production cost function is fixed and the breakpoints are time-invariant. Moreover, we observe that computational complexity of the DP algorithm is as good as the existing algorithms developed for solving special cases of our problem (e.g. Florian and Klein [9], Atamtürk and Hochbaum [10], Hellion et al. [11]). Results of the computational experiments indicate that the DP performs quite well for small and medium sized instances, but (as it is expected) its solution time for larger instances gets larger. For these instances, we propose a DP based heuristic to find a good quality solution in reasonable time.

Next, we consider the stochastic lot sizing problem where demand follows a stochastic process. In the classical lot sizing problem, it is assumed that the demand of each period is known with certainty although this is not the case for most of the production and inventory systems and approximating the demand precisely may be very difficult. In the stochastic lot sizing problem, this assumption is relaxed but the probability distribution of the demand is assumed as known.

In Chapter 4, we consider the stochastic lot sizing problem with controllable processing times where processing times of jobs can be controlled in return for extra cost (compression cost). Processing time of a job can be controlled (and reduced) by changing the machine speed, allocating extra manpower, subcontracting, overloading, consuming additional money or energy. Although these options are available in many real life production and inventory systems, in the traditional studies on the lot sizing problem, processing times of jobs are assumed as constant. Thus, this is the first study that considers the stochastic lot sizing problem with controllable processing times. We consider this problem under static uncertainty strategy and α service level constraints where α represents probability of no stock out in any period. In the static uncertainty strategy, all the decisions are taken at the beginning of the planning horizon and production plan is implemented without any revision.

If the production and compression cost functions are concave, then the DP algorithm developed in Chapter 3 can be used to solve the problem studied in Chapter 4. However, a concave function represents economies of scale and in the controllable processing times context, it should be harder to reduce processing times in larger amounts. Thus, it should cost more. Therefore, in Chapter 3, we assume that the compression cost function is a convex function since a convex function can represent increasing marginal costs. But, when the total cost function contains convex terms, nice optimal solution properties of the problem do not hold anymore [12] and the inclusion of a nonlinear cost component complicates the problem formulation significantly. We utilize the recent advances in second order cone programming to alleviate this difficulty. In this study, we formulate the problem as a second-order cone mixed integer program (SOCMIP), strengthen the formulation (conic strengthening) and solve it by a commercial solver. Moreover, we obtain some convex hull and computational complexity results. We conduct an extensive computational study to see the effect of controllable processing times in solution quality and observe that even with small changes in processing times, it is possible to obtain a better solution; thus, constant capacity assumption of the existing studies may lead to worse solutions.

Dynamic uncertainty strategy is another strategy to solve the stochastic lot sizing problems. As it can be understood from the name, in this strategy, production decisions are taken dynamically as response to demand realizations. It is possible to find less costly production plans with dynamic uncertainty strategies since more information is obtained until the time of the decision. On the other hand, under this strategy, production plan is not known in advance, or there may be a production plan on hand but revisions should be done in it periodically. This makes it harder to manage the system. For example, frequent revisions in the production schedule may cause problematic buyer-supplier relations. Moreover, it is possible to run out of raw materials, or not to have enough capacity for the decided production amount. This situation is called “system nervousness”. Nervousness is one of the most important performance measures in the inventory control theory [13].

As a final problem, we study the stochastic lot sizing problem with nervousness considerations in Chapter 5. When demand follows a finite discrete probability distribution, it can be represented by a scenario tree. In the classical scenario tree formulation of the stochastic lot sizing problem, setup and production decisions are taken for each scenario separately. Thus, this solution method contains both setup and quantity oriented nervousness. In Chapter 5, we propose a formulation in which setup oriented nervousness is eliminated and quantity oriented nervousness is reduced by some additional parameters and constraints. To do this, we restrict the production amounts under different scenarios to a range which is defined by new decision variables called *promised* production amounts. Promised production amounts are decided for periods and the production amount for any period in any scenario should be in some certain range of the promised production amount of the corresponding period. Since promised production amounts are independent from demand realizations, their values can be obtained when the problem is solved. One can use these values in arrangements before the production, purchasing or transportation starts. For example, a producer can make arrangements for the production like material planning, or a buyer can inform its supplier or transporter and reserve capacities for future periods by using these values.

In Chapter 5, we again assume that the processing times are controllable. We formulate this problem as a SOCMIP and show that continuous mixing set is a relaxation of the lot sizing problem with controllable processing times. We develop two classes of valid inequalities based on mixing and continuous mixing set substructures of our formulation and computationally test these inequalities.

Chapter 2

Literature Review

In the classical lot sizing problem, we would like to find a minimum cost production plan over a planning horizon of T periods. The demand d_t, the production cost function p_t and the inventory holding cost function h_t are given for each period t. Let x_t be the amount produced in period t and s_t be the stock on hand at the end of period t. Using these variables, the lot sizing problem can be modeled as

LS

$$\min \quad \sum_{t=1}^{T} p_t(x_t) + \sum_{t=0}^{T} h_t(s_t) \tag{2.1}$$

$$\text{s.t.} \quad s_{t-1} + x_t = d_t + s_t \qquad t = 1, \ldots, T, \tag{2.2}$$

$$s_0 = 0, \tag{2.3}$$

$$s, x \geq 0. \tag{2.4}$$

Constraints (2.2) are inventory balance constraints. The assumption on the initial inventory being zero is imposed by constraint (2.3) and is made without loss of generality. Constraints (2.4) are variable restrictions. The objective function (2.1) is the sum of production and inventory holding costs.

In this chapter, we first review the related literature on the deterministic and stochastic lot sizing problems. Next, we will give a brief summary of the studies on the controllable processing times, system nervousness, and mixing sets.

2.1 Deterministic Lot Sizing Problem

Since the seminal paper by Wagner and Whitin [8] a lot of research has been done on lot sizing problems. Wagner and Whitin [8] study the uncapacitated lot sizing problem with linear cost functions and develop an $O(T^2)$ time algorithm for solving the problem where T is the length of the planning horizon. Veinott [14] shows that Wagner-Whitin algorithm still works in $O(T^2)$ time when the production and inventory holding costs are arbitrary concave functions. Zangwill [15] studies the same problem in which backlogging is allowed and shows that this problem is also solvable in $O(T^2)$ time when all the cost functions are concave.

Following two definitions are mostly made use of in optimal policies for variations of the lot sizing problem: *regeneration interval* and *fractional period*. Time interval $[j, l]$ is called a regeneration interval if the initial inventory of period j and the final inventory of period l are zero, and final inventory of any period between j and l is positive. A period in this interval $i \in [j, l]$ is called a fractional period if the production amount in this period is positive but not equal to the capacity. Note that these two definitions may be modified according to the problem setting, i.e, cost function, upper/lower bounds on production or inventory amounts, etc.

Florian and Klein [9] find an optimal policy for the capacitated lot sizing problem with concave production and inventory holding cost functions. The authors show that there exists an optimal solution for the problem that is composed of a sequence of regeneration intervals such that at most one fractional period exists in each regeneration interval. They develop an $O(T^4)$ time algorithm for solving the problem when the capacity is constant over the planning horizon.

Florian et al. [12] and Bitran and Yanasse [16] show that numerous special cases of the lot sizing problem are NP-Hard and develop polynomial time algorithms for some other special cases of the problem, i.e., nondecreasing/nonincreasing costs/capacities. Florian et al. [12] show that the lot sizing problem remains NP-Hard in each of the following cases:

- arbitrary cost functions, no setup costs, no capacity limits;
- concave cost functions, no setup costs, arbitrary capacity limits;
- convex cost functions, unit setup costs, no capacity limits.

Since even the uncapacitated lot sizing problem with unit setup costs and convex production cost functions is NP-Hard, most of the studies on the lot sizing problem with convex cost functions assume there exist no setup costs [17, 14, 18, 19, 20, 21, 22]. Note that when setup costs are zero, the problem can be modeled as a minimum cost network flow problem [23]. Erenguc and Aksoy [24] study the lot sizing problem where the production cost function contains a fixed set up cost and a piecewise linear convex production cost function which is composed of two segments. The first segment of the cost function corresponds to the regular production cost and the second segment represents using overtime in the production. The authors develop a branch and bound algorithm for solving the problem.

Shaw and Wagelmans [25] study the capacitated lot sizing problem with piecewise linear production cost functions and arbitrary inventory holding costs, and develop a pseudo-polynomial time dynamic programming algorithm for solving the problem. Van Hoesel and Wagelmans [26] show that there exists a fully polynomial approximation scheme for the lot sizing problem with piecewise concave production cost functions and nondecreasing inventory holding costs, when the number of pieces of the production cost function is polynomially bounded in the size of the problem.

In Chapter 3, we consider the lot sizing problem with piecewise concave production cost functions. A function p is piecewise concave with breakpoints at $b^0 < b^1 < \ldots < b^m$, if p is concave in each of the m intervals $[b^{j-1}, b^j]$ for

$j = 1, \ldots, m$. Note that concavity of p in each of the intervals implies that it is lower semi-continuous [27]. Swoveland [28] presents characteristics of an optimal solution for the lot sizing problem with piecewise concave inventory holding and production cost functions. He proposes a pseudo-polynomial time dynamic programming algorithm to solve this problem.

Examples of piecewise concave production costs are depicted in Figures 2.1 and 2.2. In Figure 2.1, the first two functions represent common quantity discounts known as incremental discount and all units discount. There are many studies in the literature that consider the lot sizing problem with these quantity discounts. But in most of the studies on the lot sizing problems with quantity discounts, it is assumed that there is no capacity restriction. Federgruen and Lee [29] consider the lot sizing problem with both all units and incremental discounts. The authors assume that the production cost function has two pieces and propose dynamic programming algorithms of complexity $O(T^3)$ and $O(T^2)$ for the problems with all units discount and incremental discount, respectively. Li et al. [30] study the lot sizing problem with all units discount and resales under the assumptions that the breakpoints of the cost function are time-invariant, the number of breakpoints is fixed and there is no capacity constraint. They develop an $O(T^{m+3})$ time algorithm to solve this problem, where m is the number of breakpoints.

Cost function given in Figure 2.1c is the modified all units discount which is studied by Chan et al. [31]. Chan et al. [31] prove that the lot sizing problem with this cost structure is NP-hard when either the production cost functions vary from period to period or the number of breakpoints is not bounded by a constant. Archetti et al. [32] present polynomial time algorithms to solve special cases of the lot sizing problem with modified all units discount and incremental discount when the cost functions are time-invariant.

A special case of the function given in Figure 2.1d, in which the second segment has no end point is considered by Atamtürk and Hochbaum [10]. Atamtürk and Hochbaum [10] study the lot sizing problem with subcontracting where the production and subcontracting costs are concave nondecreasing functions and the inventory holding cost is a linear function. The overall production cost function is

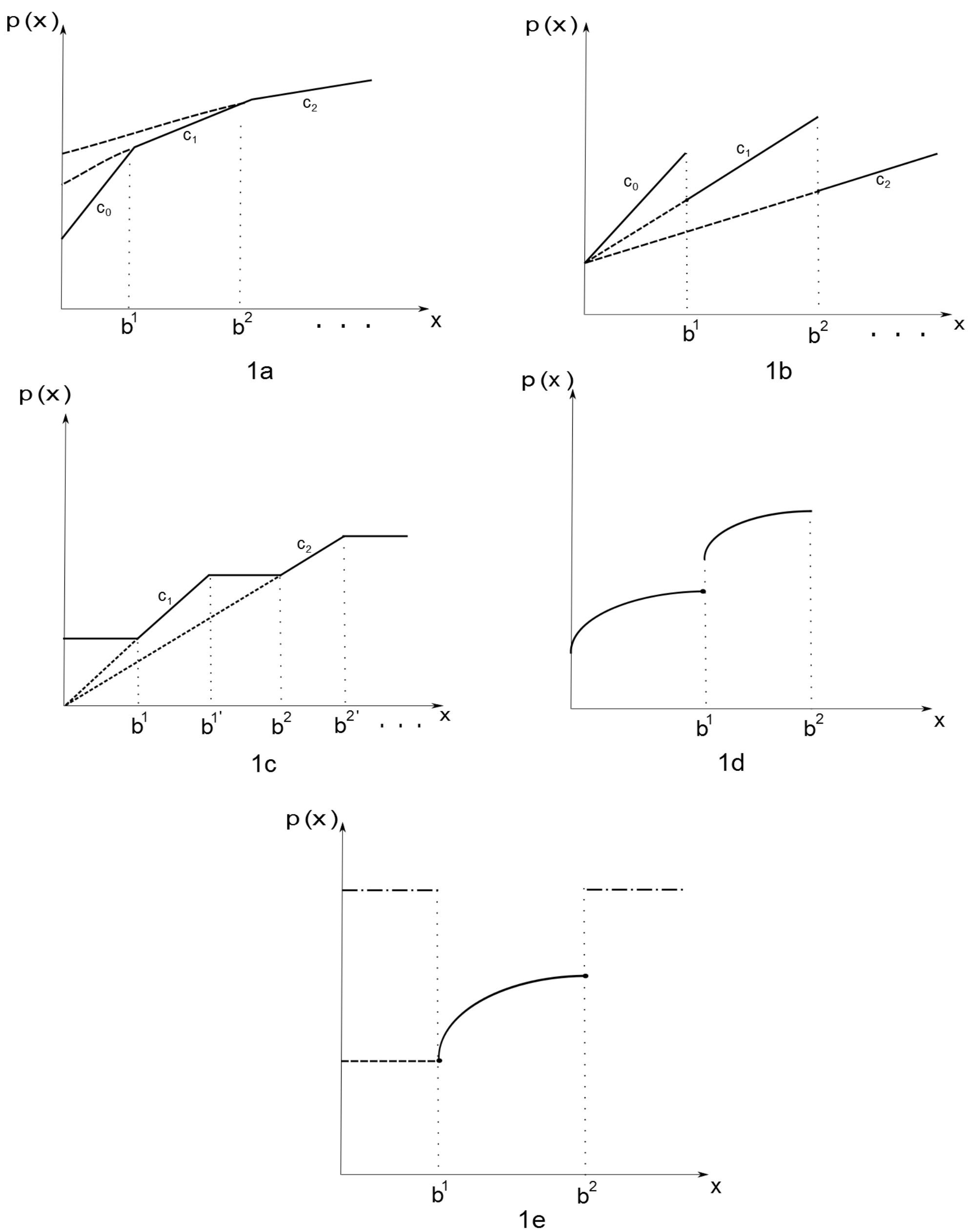

Figure 2.1: Some special cases of piecewise concave functions

piecewise concave with two segments: the first piece of the function corresponds to regular production and the second piece corresponds to subcontracting or overloading. The authors develop an $O(T^5)$ time dynamic programming algorithm for the case where the regular production capacities (the breakpoint of the cost function) are the same for all periods.

Finally, the production cost function given in Figure 2.1e represents constraints on minimum production (order) quantities as studied by Hellion et al. [11]. In this setting, if there is a production at a given period, then the production amount should not be less than a minimum level b^1 and should not exceed the capacity b^2. The authors assume that the production and inventory holding cost functions are concave and propose a dynamic programming algorithm for this problem. The time complexity reported in Hellion et al. [11] was corrected and reported as $O(T^6)$ [33]. A special case of this problem in which production and inventory holding costs are linear is studied by Okhrin and Richter [34]. They assume that there is no setup cost and unit production and inventory holding costs are constant over the planning horizon. The authors develop a polynomial time algorithm to solve this problem.

As seen above, piecewise concave functions can be used to represent discounts, subcontracting, capacity acquisition, overloading, as well as minimum quantity requirements and capacities. In addition, one can represent any combination of these using piecewise concave functions. In Figure 2.2a, we model a setting with discounts and overloading. The unit cost, c_0, up to the first breakpoint b^1 can be viewed as the regular unit purchasing cost. Then a quantity discount applies and the unit cost becomes $c_1 < c_0$ up to the second breakpoint b^2, which is the capacity of the supplier. Afterwards, the supplier requires use of overtime (or subcontracting) in order to fulfill the additional orders and hence the unit cost is $c_2 > c_0$. Note that the resulting cost function is neither convex nor concave, and it is a piecewise linear function.

Now consider the case where several suppliers give offers (possibly with discounts) for a product and the company purchases its products from at most one

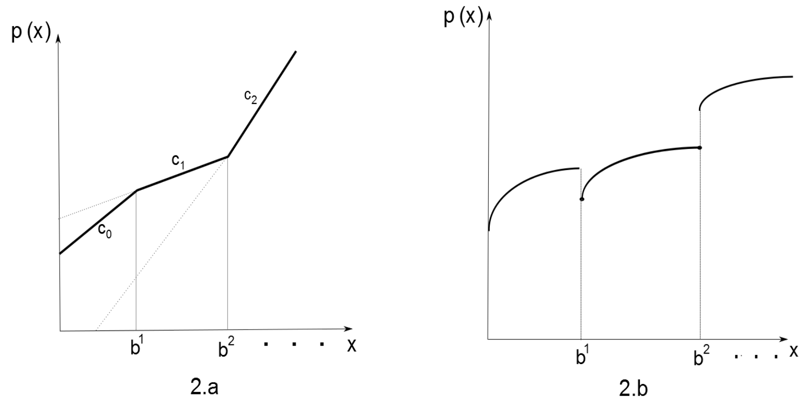

Figure 2.2: Examples of piecewise concave functions

supplier in each period. Then the production cost is the minimum of the purchasing costs over all suppliers and is a piecewise concave function if the cost function of each supplier is concave. An example is given in Figure 2.2b in which each segment of the cost function represents a supplier. The second supplier offers the most attractive price but has a lower bound for procurement, b^1 units, and has a capacity of b^2 units. It is more beneficial to buy from the first supplier up to b^1 units and from the third supplier after b^2 units. Accordingly, decisions on the purchasing amounts in each period will also determine the supplier of each period. Therefore, this problem can be seen as a supplier selection and lot sizing problem.

To sum up, special cases of lot sizing problem with cost functions depicted in Figure 2.1 are polynomially solvable. However, to the best of our knowledge, there is no polynomial time algorithm to solve the problem with cost functions like those in Figure 2.2. Indeed, the complexity of the problem is open for the case where the number of breakpoints is fixed and the breakpoints are time-invariant. In Chapter 3, we prove that the lot sizing problem with piecewise concave production cost functions can be solved in polynomial time under these assumptions.

2.2 Stochastic Lot Sizing Problem

In this section, we review the studies on stochastic lot sizing problems.

Silver [35] suggests a heuristic solution procedure for solving the stochastic lot sizing problem. Laserre et al. [36] consider the stochastic capacitated lot sizing problem with inventory bounds and chance constraints on inventory. They show that solving this problem is equivalent to solving a deterministic lot sizing problem. Bookbinder and Tan [37] study the stochastic uncapacitated lot sizing problem with α-service level constraints under three different strategies (static uncertainty, dynamic uncertainty and static-dynamic uncertainty). Service level α represents the probability that inventory will not be negative. In other words, it means that with probability α, the demand of any period will be satisfied on time. Under the static uncertainty decision rule, which is the strategy that will be used in our study, all the decisions (production and inventory decisions) are taken at the beginning of the planning horizon (frozen schedule). The authors formulate the problem and show that their model is equivalent to the deterministic problem by showing the correspondence between the terms of these two formulations.

Service level constraints are mostly used in place of shortage or backlogging costs in the stochastic lot sizing problems. Since shortages may lead to loss of customer goodwill or delays on the other parts of the system, it may be hard to estimate the backlogging or shortage costs in the real life production and inventory systems. Rather than considering the backlogging cost as a part of the total cost function, a specified level of service (in terms of availability of stock) can be assured by service level constraints and when the desired service level is high, backlogging costs can be omitted. This situation makes the usage of service level constraints more popular in the real life systems [37, 38, 39]. A detailed investigation of different service level constraints can be found in Chen and Krass [39].

Vargas [40] studies (the uncapacitated version of) the problem of Bookbinder and Tan [37] but rather than using service level constraints he assumes that there is a penalty cost for backlogging, the cost components are time varying and there is a fixed lead time. He develops a stochastic dynamic programming algorithm, which is tractable when the demand follows a normal distribution. Sox [41] studies the uncapacitated lot sizing problem with random demand and non-stationary costs. He assumes that the distribution of demand is known for each period and considers the static uncertainty model, but uses penalty costs instead of service level constraints. He formulates the problem as an MIP with nonlinear objective (cost) function and develops an algorithm that resembles the Wagner-Whitin algorithm.

In the static-dynamic uncertainty strategy of Bookbinder and Tan [37], the replenishment periods are determined first, and then replenishment amounts are decided at the beginning of these periods. They also suggest a heuristic two-stage solution method for solving this problem. Tarım and Kingsman [42] consider the same problem and formulate it as MIP. Moreover, Özen at al. [43] develop a non-polynomial dynamic programming algorithm to solve the same problem. Recently, Tunç et al. [44] reformulate the problem as MIP by using alternative decision variables and Rossi et al. [45] propose an MIP formulation based on the piecewise linear approximation of the total cost function, for different variants of this problem.

In the dynamic uncertainty strategy, production decision for any period is made at the beginning of that period. Dynamic and static-dynamic strategies are criticized due to the system nervousness they cause; supply chain coordination may be problematic under these strategies since the production decision for each period is not known until the beginning of the period [46, 47].

There are studies in the literature, in which instead of α service level, fill rate criterion (β service level) is used. Fill rate can be defined as the proportion of demand that is filled from available stock on hand. Thus, this measure also includes information about the backordering size. Tempelmeier [48] proposed a heuristic approach to solve the multi-item capacitated stochastic lot sizing problem under fill rate constraint. Helber et al. [49] consider the multi-item stochastic capacitated lot sizing problem under a new service level measure, called as δ-service-level. This service level reflects both the size of the backorders and waiting time of the customers and can be defined as the expected percentage of the maximum possible demand-weighted waiting time that a customer is protected against. The authors assume that the cost components are time-invariant and there is an overtime choice with linear costs for each period. They develop a nonlinear model and approximate it by two different linear models.

The stochastic lot sizing problem can be formulated using multistage stochastic programming if the demand follows a finite discrete probability distribution. In multistage stochastic programming, a scenario tree is constructed and any path from the root node to a leaf node represents a scenario. Note that, by a scenario tree dependency of demand of different periods can be formulated. However, the tree size grows exponentially with the number of possible demand realizations; for example, for 10 periods ($T = 10$) and 2 possible demand realizations for each period, number of nodes is 2047.

Escudero and Kamesam [50] consider the multistage stochastic lot sizing problem with two suppliers ((capacitated) in-house production and (uncapacitated) vendor supply) and different recourse options: simple (production decisions are the same for all scenarios), partial (in-house production decisions are the same for all scenarios) and full (all decisions may be different for different scenarios). The authors assume there is no setup cost and propose a heuristic solution method by clustering the time horizon into three stages: first two stages are periods 1 and 2, respectively, and the remaining time periods are assumed as stage 3.

Ahmed et al. [51] consider the multistage stochastic capacity expansion model and draw an equivalence between this problem and the multistage stochastic uncapacitated lot sizing problem. The authors formulate the stochastic uncapacitated lot sizing problem as a facility location problem and show that the Wagner-Whitin property does not hold this problem. Brandimarte [52] formulates the multi-item capacitated stochastic lot sizing problem as a facility location problem, and develops a fix and relax heuristic by partitioning the setup variables according to the time index.

Guan et al. [53] develop a family of valid inequalities, called (Q, S_Q), for the stochastic uncapacitated lot sizing problem and Guan et al. [54] show these inequalities are sufficient for describing the convex hull of the set of solutions for two period problem. Then, DiSumma and Wolsey [55] prove that these inequalities are mixing inequalities and extend these results to the constant capacity case. Guan et al. [56] introduce a scheme called "pairing" in order to derive valid inequalities for mixed integer sets. Guan et al. [57] develop valid inequalities for multistage stochastic programs by applying the pairing scheme and test their branch-and-cut method on the (capacitated) stochastic lot sizing problem instances.

Halman et al. [58] show that multistage stochastic lot sizing problem is NP-hard even the problem is uncapacitated, production and inventory costs are linear and there are two possible demand scenarios for each period. In the study by Guan [59], polynomial time (in the tree size) dynamic programming algorithms are developed for the problem when backlogging is possible and/or capacities are varying between periods.

Recently, Luedtke [60] proposes a new branch-and-cut decomposition algorithm for mathematical programs with probabilistic constraints, and tests the algorithm on probabilistic resource planning problem which is the problem of choosing and allocating the resources to customers in order to minimize the total cost, while respecting the resource capacities and satisfying service level constraints on customers' demand satisfaction.

2.3 Controllable Processing Times

Controllable processing times is well studied in the context of scheduling. One of the earliest studies on scheduling with controllable processing times is conducted by Vickson [61].

Earlier studies on this subject assume linear compression costs as adding non-linear terms to the objective (total cost) function may make the problem more difficult [62]. However, as it is stated in recent studies, reducing processing times gets harder (and more expensive) as the compression amount increases in many applications [62, 63]. For example, by increasing machine speed, processing times can be reduced, but this also decreases life of the tool and an additional tooling cost is incurred. Moreover, increasing the machine speed may also increase the energy consumption of the facility. Another example is a transportation system in which trucks may be overloaded or their speeds could be increased in return for extra cost due to increasing fuel consumption or limiting the carbon emission. Thus, considering a convex compression cost function is realistic since a convex function represents increasing marginal costs and may limit higher usage of the resource due to environmental issues. Kayan and Aktürk [62] and Aktürk et al. [63] consider a CNC machine scheduling problem with controllable processing times and convex compression costs. Jansen and Mastrolilli [64] develop approximation schemes, Gürel et al. [65] use an anticipative approach to form an initial solution, Türkcan et al. [66] use a linear relaxation based algorithm for the scheduling problem with controllable processing times. Shabtay and Kaspi [67, 68] and Shabtay and Steiner [69] study the scheduling problem with convex resource consumption functions. A detailed review on scheduling with controllable processing times can be found in Shabtay and Steiner [70].

As reducing processing time of a job is equivalent to increasing production capacity, subcontracting, overloading or capacity acquisition can be seen as special cases of the controllable processing times. There are studies in the literature that consider the lot sizing problem with subcontracting (or outsourcing) [71, 49, 72] or capacity acquisition (or expansion) [51, 73, 74]. However, in all of these studies costs of these options are assumed as linear or concave. This assumption makes it possible to extend the classical extreme point or optimal solution properties for these cases. In Chapter 4, we assume that the compression cost is a convex function of the compression amount, and as it is stated in the previous section, even the uncapacitated (deterministic) lot sizing problem with unit setup costs and convex production cost functions is NP-hard. On the other hand, with the recent advances in convex programming techniques, many commercial solvers (like IBM ILOG CPLEX) can now solve second order cone programs (SOCP). We make use of this technique and formulate the problem as SOCMIP so that it can be solved by a commercial solver.

2.4 System Nervousness

System nervousness is caused by uncertainty of the production plans. Nervousness in an upper level of the supply chain affects all the supply chain and it causes lack of coordination in the production systems. If the system is not flexible, i.e. revisions cannot be handled easily, nervousness becomes a bigger problem and in these systems, it may be more appropriate to look for a more stable production plan, in which revisions are not needed [75].

There are two types of system nervousness considered in the literature: setup oriented nervousness and quantity oriented nervousness. Setup oriented nervousness is caused by changes in the production periods (cancellation of a production decision or deciding to produce in a period that is not considered as a production period before). Quantity oriented nervousness is related to modifications (increase or decrease) in the decided production amounts. In most of the studies in the literature, it is assumed that setup oriented nervousness is more critical.

There are few studies in the literature that consider nervousness. In early studies on this subject, simulation of the systems is used to test different strategies and investigate the impact of parameter settings [76, 77, 78, 79, 80]. Kropp et al. [81] and Kropp and Carlson [82] incorporated nervousness to the total cost function by assigning cost parameters to nervousness and solve the problems heuristically.

Inderfurth [83], De Kok and Inderfurth [13], and Heisig [84, 75] consider system nervousness caused by inventory policies like (s, S), (s, nQ), (R, S), and develop different measures for nervousness. However, in all of these studies, the systems are assumed as stationary. According to these studies, (s, S) policy, which is the optimal policy for stationary systems [85], performs worst in terms of system nervousness, since production decision for each period is taken at the beginning of that period according to the revised inventory level.

In recent studies, new nervousness measures are considered. Kılıç and Tarım [86] develop a method for measuring cost of system nervousness in non-stationary systems under (s, S) and (R, S) policies and conclude that (R, S) policy performs better in terms of nervousness. Tunç et al. [46] develop another method to calculate cost of nervousness for a system in which (s, S) and (R, S) inventory policies are considered and Tunç et al. [47] propose a method for evaluating costs of setup and quantity oriented nervousness by using static, dynamic and static-dynamic uncertainty strategies. Note that static uncertainty strategy is nervousness free since all the production decisions are taken at the beginning of the planning horizon. On the other hand, dynamic uncertainty strategy causes both setup oriented and quantity oriented nervousness. As a combination of these two strategies, static-dynamic uncertainty strategy causes only quantity oriented nervousness. The authors conclude that, setup oriented nervousness can be avoided by a small cost increase in the system whereas it is harder to avoid quantity oriented nervousness.

Rolling horizon is frequently applied to the systems when it is not possible to have an accurate forecast for the demand of further periods [87]. Determining the length of the planning interval is one of the problems of the rolling horizon approach. As production plan for a given period may change as much as this interval length, this method may cause system nervousness [83]. Kazan et al. [88] analyze different algorithms under the rolling horizon when there exist cost terms associated with setup and quantity oriented nervousness. The authors perform a detailed computational experiment for comparing the algorithms under different environments.

In the problem studied in Chapter 5, we restrict quantity oriented nervousness by some additional constraints on the production decisions. The idea is very similar to the *restricted recourse* concept of Vladimirou and Zenios [89]. Vladimirou and Zenios [89] search for recourse *robust* solutions for two-stage stochastic linear programs by investigating different formulations in which variability of the second stage decisions is restricted via some additional constraints. The authors develop solution procedures for these formulations by using the primal-dual interior point method.

2.5 The Mixing and Continuous Mixing Sets

Mixing and continuous mixing sets arise as relaxations of lot sizing problems. For example, mixing set is a submodel of the constant capacity lot sizing problem and the two period stochastic lot sizing problem with constant capacities. Continuous mixing set arises as relaxations of these problems when backlogging is allowed [90]. In Chapter 5, we will show that the continuous mixing set is also a relaxation of the lot sizing problem with controllable processing times.

In this section, we introduce the mixing and continuous mixing sets, and valid inequalities developed for these sets.

The simplest possible MIP set is called the basic MIP set:

$$X^{MI} = \{(s, y) \in \mathbb{R}^1_+ \times \mathbb{Z}^1 : s + y \geq b\}.$$

The simple mixed integer rounding (MIR) inequality

$$s \;\geq\; f\left(\lceil b \rceil - y\right)$$

is valid for the set X^{MI}, and suffices to give the convex hull of X^{MI}, where $f = b - \lfloor b \rfloor$.

The mixing set is an intersection of K basic MIP sets with the same continuous variable:

$$X^{MIX} = \{(s, y) \in \mathbb{R}^1_+ \times \mathbb{Z}^K : s + y_k \geq b_k,\; k = 1, \ldots, K\}.$$

Note that K simple MIR inequalities $s + f_k s_k \geq f_k \lceil b_k \rceil$ are valid for X^{MIX} where $f_k = b_k - \lfloor b_k \rfloor$, $k = 1, \ldots, K$. But these inequalities do not suffice to describe the convex hull of X^{MIX} when $K > 1$. Günlük and Pochet [91] develop strong valid inequalities for the set X^{MIX} by a procedure called "mixing". The valid inequalities obtained by the mixing procedure are called "mixing inequalties". The mixing procedure can be described as follows.

Let $R = \{i_1, \ldots, i_r\} \subseteq \{1, \ldots, K\}$ be an ordered set such that $0 = f_{i_0} \leq f_{i_1} \leq f_{i_2} \leq \ldots \leq f_{i_r} < 1$. Mixing inequalities for the set R are given by

$$s \;\geq\; \sum_{j=1}^{r} \left(f_{i_j} - f_{i_{j-1}}\right)\left(\lfloor b_{i_j} \rfloor + 1 - y_{i_j}\right) \tag{2.5}$$

$$s \;\geq\; \sum_{j=1}^{r} \left(f_{i_j} - f_{i_{j-1}}\right)\left(\lfloor b_{i_j} \rfloor + 1 - y_{i_j}\right) + (1 - f_{i_r})\left(\lfloor b_{i_1} \rfloor - y_{i_1}\right). \tag{2.6}$$

The mixing inequalities (2.5) and (2.6), and the inequalities $s + y_k \geq b_k$, $s \geq 0$ describe the convex hull of X^{MIX} [91].

Separation of mixing inequalities can be carried out in $O(K\, log\, K)$ time by the following procedure [92]. Order the variables and inequalities $k = 1, \ldots, K$ so that $f_1 \leq f_2 \leq \ldots, \leq f_K$. Let $(s^*, y^*) \in \mathbb{R}^1_+ \times \mathbb{R}^K$, and $\beta = \max_{j=1,\ldots,K}\left(\lfloor b_i \rfloor + 1 - y^*_j\right)$. If $\beta \leq 0$, then left hand sides of inequalities (2.5) and (2.6) are both nonpositive; thus all the mixing inequalities are satisfied. Otherwise, find a subsequence $i_1, \ldots, i_r$ of $\{1, \ldots, K\}$ so that

$$\begin{aligned} i_j &= \arg\max_{i > i_{j-1}} \{\lfloor b_i \rfloor + 1 - y^*_i : \lfloor b_i \rfloor + 1 - y^*_i > [\beta - 1]^+\} \quad \text{for } j = 1, \ldots, r \\ \lfloor b_i \rfloor &+ 1 - y^*_i \leq [\beta - 1]^+ \qquad\qquad \text{for } i > i_r. \end{aligned}$$

This can be done in linear time by working backwards. Let

$$\gamma = \sum_{j=1}^{r} \left(f_{i_j} - f_{i_{j-1}}\right)\left(\lfloor b_{i_j} \rfloor + 1 - y^*_{i_j}\right).$$

If $\beta \leq 1$ and $\gamma > s^*$, the mixing inequality (2.5) is the most violated inequality. If $\beta \leq 1$, but $\gamma \leq s^*$, then no inequality is violated. If $\beta > 1$ and $\gamma + (\beta - 1) > s^*$, then the mixing inequality (2.6) is most violated, and if $\gamma + (\beta - 1) \leq s^*$ no inequality is violated.

The continuous mixing set

$$X^{CMIX} = \{(s, r, y) \in \mathbb{R}^1_+ \times \mathbb{R}^K_+ \times \mathbb{Z}^K : s + r_k + y_k \geq b_k,\ k = 1, \ldots, K\}$$

is first studied by Miller and Wolsey [93]. The authors generalize the mixing inequalities (2.5) and (2.6) for this set, but show that these inequalities are not sufficient for describing the convex hull of the set X^{CMIX}. Moreover, the authors introduce an extended formulation of $O(K^2)$ variables and constraints. Van Vyve [90] studies a generalization of the continuous mixing set where the nonnegativity restriction of the continuous variable s is dropped. The author develops valid inequalities called "cycle inequalities" and an extended formulation of size $O(K) \times O(K^2)$ variables and constraints. We will explore the generation and separation of the cycle inequalities in Chapter 5.

2.6 Summary

As it is stated in Section 2.1, special cases of the lot sizing problem with piecewise concave production cost functions were studied in the literature before. However, to the best of our knowledge, computational complexity of this problem was an open question in the literature. In Chapter 3, we will consider this problem and develop a dynamic programing algorithm to solve this problem in polynomial time.

Sections 2.1 and 2.2 reveal that there is not so much work on the lot sizing problem with convex cost functions. In Chapter 4, we will study the stochastic lot sizing problem with controllable processing times and convex compression cost functions. This problem was not studied in the literature before. We will utilize the recent advances in second order cone programming to solve this problem.

As it can be observed from Section 2.2, it is possible to obtain less costly production plans by using dynamic strategies. On the other hand, dynamic strategies cause system nervousness. In Chapter 5, we will consider the stochastic lot sizing problem with controllable processing times under a dynamic strategy. We will assume that the demand can be represented by a scenario tree, and formulate the problem so that the nervousness of the system is reduced.

Chapter 3

Lot Sizing with Piecewise Concave Production Cost Functions

In this chapter, we study the lot sizing problem with piecewise concave production cost functions and concave inventory holding cost functions. We call this problem the "lot sizing problem with piecewise concave production costs" and abbreviate it with LS-PC.

In Section 3.1, we formally define the problem LS-PC and state some important properties of an optimal solution to the problem. In Section 3.2, we present a polynomial time dynamic programming algorithm for solving the problem when the number of breakpoints is fixed and the breakpoints are time-invariant and show that the complexity of the DP is as good as the complexity of algorithms available in the literature for some special cases of the problem. We then report our computational experiments in Section 3.3, and propose a state space reduction based heuristic algorithm for large instances in Section 3.4. Finally in Section 3.5 we present some concluding remarks.

The results of this chapter are published in INFORMS Journal on Computing

[94].

3.1 Problem Definition and Properties of Optimal Solutions

Given a planning horizon of T periods, demand of each period d_t, and production and inventory holding costs, our problem is to find a minimum cost production plan to satisfy the demand. We assume that the inventory holding cost function $h_t(.)$ is a concave function on $[0, \infty)$ and $p_t(.)$ is a piecewise concave function on $[0, \infty)$ with m_t finite breakpoints $b_t^1, \dots, b_t^{m_t}$ such that $b_t^0 = 0$ and $b_t^{i-1} < b_t^i$ for $i = 1, \dots, m_t$.

As typically done in the lot sizing literature (see [92]), we will use the concepts of regeneration intervals and fractional periods in analyzing the structure of optimal solutions. An interval $[j, l]$ with $1 \leq j \leq l \leq T$, $s_{j-1} = s_l = 0$ and $s_t > 0$ for $j \leq t < l$ is referred to as a *regeneration interval* and a period i whose production level is not equal to any of the breakpoints of the production cost function, i.e., $x_i \in [b_i^0, \infty) \setminus \{b_i^0, \dots, b_i^{m_i}\}$ is referred to as a *fractional period*. We define $b_i^{m_i+1} = \infty$ for all i.

If the production cost function is not monotone (see Figures 2.1e and 2.2b), we may have positive ending inventory in all optimal solutions. Therefore, contrary to the case with the classical lot sizing problems, we cannot say that there exists an optimal solution that is composed of a series of successive regeneration intervals. However, for our problem, there exists an optimal solution that is composed of a series of regeneration intervals that cover the interval $[1, j-1]$ plus an interval $[j, T]$ for some $1 \leq j \leq T+1$. We know the following properties for these intervals.

Theorem 3.1.1. *[28] There exists an optimal solution to the problem LS-PC such that in each regeneration interval $[j, l]$, there exists at most one fractional period.*

Theorem 3.1.1 is a generalization of the "fractional period property" for the capacitated lot sizing problem [9]. Note that if $x_i > b_i^{m_i}$, then period i is a fractional period.

Theorem 3.1.2. *Suppose that the ending inventory is positive in all optimal solutions. Then, there exists an optimal solution to the problem in which the last interval* $[j, T]$ *with* $s_{j-1} = 0$ *and* $s_t > 0$ *for* $j \leq t \leq T$ *does not contain any fractional periods. In other words, there exists an optimal solution to the problem that is composed of a series of regeneration intervals that cover the interval* $[1, j-1]$ *plus an interval* $[j, T]$ *for some* $1 \leq j \leq T$ *with no fractional period in the last interval* $[j, T]$.

Proof. Suppose that at all optimal solutions we have $s_T > 0$. Let (x, s) be an optimal solution with the largest j value such that $s_{j-1} = 0$ and $s_t > 0$ for $t = j, \ldots, T$. Suppose that there exists a fractional period with $i \in [j, T]$ such that $b_i^k < x_i < b_i^{k+1}$ for some $k \in \{0, \ldots, m_i\}$. Define $\alpha = \min\{\min_{t=i}^{T} s_t, x_i - b_i^k\}$ and $\beta = b_i^{k+1} - x_i$ if b_i^{k+1} is finite and $\beta = \alpha$ otherwise. Clearly, α and β are positive.

Now consider the two solutions (x^1, s^1) and (x^2, s^2) that are the same as (x, s) except that $x_i^1 = x_i - \alpha$, $s_t^1 = s_t - \alpha$ for $t = i, \ldots, T$, $x_i^2 = x_i + \beta$, and $s_t^2 = s_t + \beta$ for $t = i, \ldots, T$. Both solutions are feasible. Optimality of (x, s) implies that

$$p_i(x_i - \alpha) + \sum_{t=i}^{T} h_t(s_t - \alpha) - p_i(x_i) - \sum_{t=i}^{T} h_t(s_t) \geq 0$$

and

$$p_i(x_i + \beta) + \sum_{t=i}^{T} h_t(s_t + \beta) - p_i(x_i) - \sum_{t=i}^{T} h_t(s_t) \geq 0.$$

Since p_i is concave on $[b_i^k, b_i^{k+1}]$ and h_t is concave on $[0, \infty)$ for each $t = i, \ldots, T$, we also have

$$\frac{\beta}{\alpha + \beta} p_i(x_i - \alpha) + \frac{\alpha}{\alpha + \beta} p_i(x_i + \beta) \leq p_i(x_i)$$

and

$$\frac{\beta}{\alpha + \beta} h_t(s_t - \alpha) + \frac{\alpha}{\alpha + \beta} h_t(s_t + \beta) \leq h_t(s_t)$$

for $t = i, \ldots, T$. Therefore, both (x^1, s^1) and (x^2, s^2) are also optimal. Either b_i^{k+1} is finite and (x^2, s^2) is an optimal solution where the fractional period i is eliminated. Or $k = m_i$ and as (x, s) is an optimal solution with the largest j value such that $s_{j-1} = 0$ (implying that $s_t^1 > 0$ for $t = i, \ldots, T$), (x^1, s^1) is an optimal solution in which i is not a fractional period anymore. □

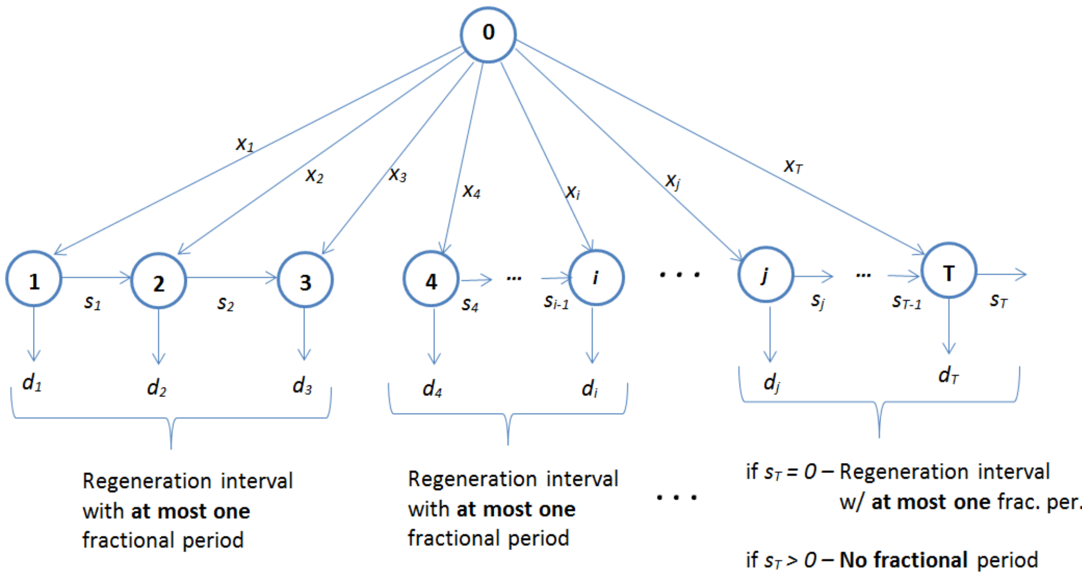

Figure 3.1: Optimal solution properties

Figure 3.1 illustrates the network flow representation of an optimal solution for LS-PC. In this graph, node i represents period i for $i = 1, \ldots, T$, node 0 is the dummy node and it represents production. Flow on the arc $(0, i)$ is given by the production amount in period i (x_i) and flow on the arc $(i, i+1)$ is the inventory at the end of period i (s_i). In this example, since $[1, 3]$ is a regeneration interval it may contain at most one fractional period due to Theorem 3.1.1. For example, if x_1 is fractional, both of x_2 and x_3 should be equal to breakpoints of the production cost function. If there exists an optimal solution such that $s_T = 0$, then there exists an optimal solution for the problem that is composed of a series of regeneration intervals. But if $s_T > 0$ in all optimal solutions, then there exists an optimal solution such that the last interval $[j, T]$ does not contain fractional period due to Theorem 3.1.2.

Remark 1. *If we assume that the inventory holding cost function, h_t, is also piecewise concave with q_t finite breakpoints $r_t^1, \ldots, r_t^{q_t}$ such that $r_t^i < r_t^{i+1}$ for*

$i = 1, \ldots, q_t - 1$ and $t = 1, \ldots, T$, then with small modifications Theorems 3.1.1 and 3.1.2 still remain valid. In this case, an interval $[j, l]$ with $1 \leq j \leq l \leq T$ is called as a regeneration interval if $s_{j-1} \in \{r^1_{j-1}, \ldots, r^{q_{j-1}}_{j-1}\}$, $s_l \in \{r^1_l, \ldots, r^{q_l}_l\}$ and $s_t \notin \{r^1_t, \ldots, r^{q_t}_t\}$ for $j \leq t < l$ [28]. Theorem 3.1.1 still holds true for this definition [28]. However, we need to restate Theorem 3.1.2 as the following:

Suppose that the ending inventory is not at a breakpoint level of the inventory holding cost, i.e. $s_T \notin \{r^1_T, \ldots, r^{q_T}_T\}$, in all optimal solutions. Then, there exists an optimal solution to the problem in which the last interval $[j, T]$ with $s_{j-1} \in \{r^1_{j-1}, \ldots, r^{q_{j-1}}_{j-1}\}$ and $s_t \notin \{r^1_t, \ldots, r^{q_t}_t\}$ for $j \leq t \leq T$ does not contain any fractional periods.

Due to Theorem 3.1.1, as it is done in the classical lot sizing problems, we can find the minimum cost solution for each regeneration interval $[j, l]$ by assuming that it consists at most one fractional period. However, it is not sufficient for finding a minimum cost solution for the problem since for the intervals $[j, T]$ we need to consider the case where it is not a regeneration interval. In this case, for the intervals $[j, T]$, due to Theorem 3.1.2, we can search for a minimum cost solution by assuming that it does not consist any fractional period. Consequently, we can find a minimum cost solution for each interval $[j, T]$ by picking the least cost solution among the cases that it is a regeneration interval or not. In the next section, we develop a dynamic programming algorithm for finding an optimal solution for LS-PC by using these results.

3.2 Dynamic Programming Algorithm

In this section, we propose a dynamic programming algorithm for the special case where the breakpoints of the production cost function are time-invariant and the number of breakpoints is fixed, i.e., $b^i_t = b^i$ for all $t = 1, \ldots, T$ and $i = 0, \ldots, m$ where $m_t = m$ for all $t = 1, \ldots, T$ and $m(\geq 1)$ is fixed.

This algorithm is a generalization of the algorithm given by Florian and Klein [95] for the constant capacity lot sizing problem with concave production cost functions.

Let e_i be a unit vector of size m in which the i^{th} component is one and the other components are zero for $i = 1, \ldots, m$ and e_0 be a zero vector of size m.

3.2.1 Minimum cost for an interval $[j, l]$ with no fractional period

First, we compute the minimum cost for a regeneration interval $[j, l]$ with $1 \leq j \leq l \leq T-1$ and for an interval $[j, T]$ for $1 \leq j \leq T$ when there is no fractional period. To this end, we define the following function. Let $\tau \in \mathbb{Z}_+^m$ and $t \in \{j, \ldots, l\}$. If $l \leq T-1$, let $F_{jl}(t, \tau)$ be the minimum cost for periods j up to t during which τ_i times b^i, for $i = 1, \ldots, m$, units are produced, no fractional production is done, given that $s_{j-1} = s_l = 0$ and $s_u > 0$ for $u \in \{j, \ldots, \min\{t, l-1\}\}$. If $l = T$, then we define the same function by dropping the requirement that $s_l = 0$. For $j \leq t$, we let $d_{jt} = \sum_{i=j}^{t} d_i$.

Note that the amount of production between periods j and t is equal to $\sum_{i=1}^{m} \tau_i b^i$ and the number of periods in which production takes place is $\sum_{i=1}^{m} \tau_i$. If $t < l$ and $\sum_{i=1}^{m} \tau_i b^i \leq d_{jt}$, then we cannot have $s_t > 0$. Also, if $t = l$ and $\sum_{i=1}^{m} \tau_i b^i \neq d_{jl}$, then $s_l = 0$ is not possible. If $\sum_{i=1}^{m} \tau_i > t - j + 1$, the production schedule is infeasible.

For $i = 0, \ldots, m$, we let

$$F_{jl}(j, e_i) = \begin{cases} p_j(b^i) + h_j(b^i - d_j) & \text{if } d_j < b^i \text{ and } (j < l \text{ or } l = T), \\ p_j(b^i) & \text{if } d_j = b^i \text{ and } j = T, \\ \infty & \text{otherwise,} \end{cases}$$

and $F_{jl}(j, \tau) = \infty$ if $\sum_{i=1}^{m} \tau_i \geq 2$.

Let $t \in \{j+1, \ldots, l\}$, and $\tau \in \mathbb{Z}_+^m$. If we produce b^i units for some $i \in \{0, \ldots, m\}$ in period t, then the minimum cost for periods j to $t-1$ is $F_{jl}(t-$

$1, \tau - e_i)$. Therefore, we compute $F_{jl}(t, \tau)$ as

$$F_{jl}(t,\tau) = \begin{cases} \infty & \text{if } \sum_{i=1}^{m} \tau_i > t - j + 1 \text{ or} \\ & (\sum_{i=1}^{m} \tau_i b^i \leq d_{jt} \text{ and } t < l) \text{ or} \\ & (\sum_{i=1}^{m} \tau_i b^i \neq d_{jl} \text{ and } t = l \text{ and } l < T) \text{ or} \\ & (\sum_{i=1}^{m} \tau_i b^i < d_{jl} \text{ and } t = l = T), \\ \min_{i=0,\ldots,m:\tau \geq e_i} \{F_{jl}(t-1, \tau - e_i) + p_t(b^i) + h_t(\sum_{i=1}^{m} \tau_i b^i - d_{jt})\} & \\ & \text{otherwise.} \end{cases}$$

We evaluate the recursion for increasing values of t and all possible values of τ. For given t and τ, $F_{jl}(t, \tau)$ can be computed in constant time since we assume that m is fixed. As $\tau_i \leq T$ for $i = 1, \ldots, m$, we have $O(T^m)$ possible τ vectors. As a result, the function F_{jl} can be evaluated in $O(T^{m+1})$ time for a given interval $[j, l]$.

3.2.2 Minimum cost for an interval $[j, l]$ with a fractional period

Next, we compute the minimum cost for a regeneration interval $[j, l]$ with $1 \leq j \leq T$ when the interval contains a fractional period. Note that for an interval $[j, T]$ that is part of an optimal solution, when the interval contains a fractional period, there exists an optimal solution with $s_T = 0$. Hence, we only consider regeneration intervals in this computation.

The minimum cost when a fractional period exists is computed for two separate cases:

Case a. The fractional production amount is less than b^m.

As we are interested in solutions with one fractional period, we know that there is no production greater than b^m.

Let $\tau \in \mathbb{Z}_+^m$, $\pi \in \mathbb{Z}_+^{m-1}$ and $t \in \{j, \ldots, l\}$. If τ_i times b^i, for $i = 1, \ldots, m$, units are produced in periods j up to $t - 1$ and π_i times b^i, for $i = 1, \ldots, m - 1$, and

$\left\lfloor \frac{d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor$ times b^m units are produced in periods $t+1$ to l, then the production amount in period t is equal to

$$\rho_{jl}\left(\tau,\pi\right) = d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i - \left\lfloor \frac{d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor b^m.$$

Now let $G_{jl}\left(t,\tau,\pi\right)$ be the minimum cost for periods j up to t during which τ_i times b^i units for $i = 1,\ldots,m$, are produced and one time a fractional production is done given that π_i times b^i, for $i = 1,\ldots,m-1$, and $\left\lfloor \frac{d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor$ times b^m units are produced after period t, $s_{j-1} = s_l = 0$ and $s_u > 0$ for $u \in \{j,\ldots,\min\{t,l-1\}\}$.

Let $\tau \in \mathbb{Z}_+^m$ and $\pi \in \mathbb{Z}_+^{m-1}$. If $\sum_{i=1}^{m} \tau_i \geq 1$ or $d_{jl} \leq \sum_{i=1}^{m-1} \pi_i b^i$ or $\sum_{i=1}^{m-1} \pi_i + \left\lfloor \frac{d_{jl} - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor > l-j$ or $\rho_{jl}\left(e_0,\pi\right) \in \{0, b^1, \ldots, b^m\} \cup (b^m, \infty)$, we set $G_{jl}(j,\tau,\pi) = \infty$. For other values, we compute

$$G_{jl}(j,e_0,\pi) = \begin{cases} p_j\left(\rho_{jl}\left(e_0,\pi\right)\right) + h_j\left(\rho_{jl}\left(e_0,\pi\right) - d_j\right) & \text{if } \rho_{jl}\left(e_0,\pi\right) > d_j \text{ and } j < l, \\ p_j\left(\rho_{jl}\left(e_0,\pi\right)\right) & \text{if } \rho_{jl}\left(e_0,\pi\right) = d_j \text{ and } j = l, \\ \infty & \text{otherwise.} \end{cases}$$

Now let $t \in \{j+1,\ldots,l\}$, $\tau \in \mathbb{Z}_+^m$ and $\pi \in \mathbb{Z}_+^{m-1}$. If $\sum_{i=1}^{m} \tau_i > t-j$ or $\sum_{i=1}^{m-1} \pi_i + \left\lfloor \frac{d_{jl} - \sum_{i=1}^{m} \tau_i b^i - \sum_{i=1}^{m-1} \pi_i b^i}{b^m} \right\rfloor > l-t$, then we set $G_{jl}\left(t,\tau,\pi\right) = \infty$. If $\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}\left(\tau,\pi\right) \leq d_{jt}$ and $t < l$, then $s_t \leq 0$ and if $\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}\left(\tau,\pi\right) \neq d_{jl}$ and $t = l$, then $s_l \neq 0$. If $d_{jl} < \sum_{i=1}^{m} \tau_i b^i + \sum_{i=1}^{m-1} \pi_i b^i$, then s_l cannot be zero. Moreover, we do not want to have $\rho_{jl}\left(\tau,\pi\right) \in \{0, b^1, \ldots, b^m\} \cup (b^m, \infty)$. Hence, we set $G_{jl}\left(t,\tau,\pi\right) = \infty$ in these cases. For the remaining values, we compute

$$G_{jl}\left(t,\tau,\pi\right) = h_t\left(\sum_{i=1}^{m} \tau_i b^i + \rho_{jl}\left(\tau,\pi\right) - d_{jt}\right) + \min\Big\{F_{jl}\left(t-1,\tau\right) + p_t\left(\rho_{jl}\left(\tau,\pi\right)\right),$$
$$\min_{i=0,\ldots,m:\tau\geq e_i} \left\{G_{jl}\left(t-1,\tau-e_i,\pi+\bar{e}_i\right) + p_t\left(b^i\right)\right\}\Big\},$$

where $\bar{e}_i$ is the restriction of e_i to the first $m-1$ entries. Here, we first add the inventory holding cost. If the fractional production takes place at period t, then the production cost is $p_t(\rho_{jl}(\tau,\pi))$ and the minimum cost for periods j to $t-1$ is $F_{jl}(t-1,\tau)$. If we produce b^i units in period t for some $i \in \{0,\ldots,m\}$, then the production cost is $p_t(b^i)$ and the minimum cost for periods j to $t-1$ is $G_{jl}(t-1, \tau - e_i, \pi + \bar{e}_i)$ since the fractional period is before period t.

For given t, τ and π, $G_{jl}(t, \tau, \pi)$ can be computed in constant time. Hence G_{jl} can be evaluated in $O(T^{2m})$ time.

Case b. The fractional production amount is greater than b^m.

Let $\tau \in \mathbb{Z}_+^m$, $\hat{\pi} \in \mathbb{Z}_+^m$, $t \in \{j, \ldots, l\}$ and $\hat{G}_{jl}(t, \tau, \hat{\pi})$ be the minimum cost for periods j up to t during which τ_i times b^i units, for $i = 1, \ldots, m$, are produced and one time a fractional production $\hat{\rho}_{jl}(\tau, \hat{\pi}) = d_{jl} - \sum_{i=1}^m \tau_i b^i - \sum_{i=1}^m \hat{\pi}_i b^i > b^m$ is done given that $\hat{\pi}_i$ times b^i, for $i = 1, \ldots, m$, units are produced after period t, $s_{j-1} = s_l = 0$ and $s_u > 0$ for $u \in \{j, \ldots, \min\{t, l-1\}\}$. The function $\hat{G}_{jl}$ can be computed in a similar way to G_{jl}. As the dimension of the vector $\hat{\pi}$ is one more than the one of π, computing $\hat{G}_{jl}$ requires $O(T^{2m+1})$ time.

3.2.3 Time complexity

Overall, we can find the minimum cost for interval $[j, l]$ as

$$\mu_{jl} = \min_{\tau \in \{0, \ldots, T\}^m} \left\{ F_{jl}(l, \tau), G_{jl}(l, \tau, \bar{e}_0), \hat{G}_{jl}(l, \tau, e_0) \right\}.$$

Theorem 3.2.1. *The lot sizing problem with piecewise concave production costs is polynomially solvable when the breakpoints of the production cost function are time-invariant and when the number of breakpoints is fixed.*

Proof. For an interval $[j, l]$ with $1 \leq j \leq l \leq T$, as evaluating the functions F_{jl}, G_{jl} and $\hat{G}_{jl}$ take $O(T^{m+1})$, $O(T^{2m})$ and $O(T^{2m+1})$ time, respectively, the minimum cost μ_{jl} can be computed in $O(T^{2m+1})$ time. Once these costs are computed, we can solve the problem by solving a shortest path problem as done for the classical lot sizing problem. Let $G = (V, A)$ be a directed graph for $V = \{1, \ldots, T+1\}$ and $A = \{(j, l+1) : 1 \leq j \leq l \leq T\}$. The shortest path problem from node 1 to node $T+1$ in the graph G with cost μ_{jl} on arc $(j, l+1)$ with $d_{jl} > 0$ and cost 0 on arc $(j, l+1)$ with $d_{jl} = 0$, solves our problem (see figure 3.2

for an example). As μ_{jl} can be computed in $O(T^{2m+1})$ time and there are $O(T^2)$ intervals, we require $O(T^{2m+3})$ time to construct the graph. This dominates the time to compute a shortest path. Therefore, the overall complexity is $O(T^{2m+3})$ and is polynomial for fixed m. □

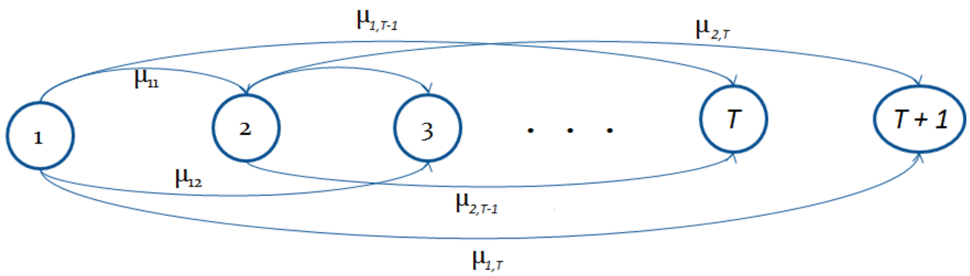

Figure 3.2: Shortest path problem

3.2.4 Special cases

Now we discuss some special cases. Suppose that the production amount in any period cannot exceed a given capacity C. This can be modeled by setting $b^m = C$ and $p_t(x) = \infty$ for $x \in (b^m, \infty)$ and $t = 1, \ldots, T$. In this case $\hat{G}_{jl} = \infty$ for all intervals $[j, l]$. Then the overall complexity of the algorithm decreases to $O(T^{2m+2})$. The constant capacity lot sizing problem is the special case with $m = 1$. For this special case our algorithm runs in $O(T^4)$ time, and hence has the same time complexity as the one of Florian and Klein [95].

Hellion et al. [11] study the capacitated lot sizing problem with concave costs, minimum order quantities (L) and constant capacities (C). In order to model this special case, we let $p_t(x) = \infty$ if $x \in (0, L) \cup (C, \infty)$, so we assume that $m = 2$. In this case, again, $\hat{G}_{jl} = \infty$ for all intervals $[j, l]$. Therefore, our DP algorithm can solve this special case of the problem in $O(T^6)$ time, which is equal to the computational complexity of the algorithm of Hellion et al. [33].

Atamtürk and Hochbaum [10] propose an $O(T^5)$ time algorithm for the special case where the production cost function has two pieces; the first piece corresponds to regular work and the second piece represents subcontracting. As $m = 1$, our

DP algorithm can also solve this problem in $O(T^5)$ time.

If we assume that backordering is allowed, we can redefine $h_t(s_t)$ as the cost of holding s_t units of inventory during period t if $s_t > 0$ and as the cost of backordering s_t units during period t if $s_t < 0$. We assume that $h_t(.)$ is a concave function on both $(-\infty, 0]$ and $[0, \infty)$, and consequently $h_t(.)$ is a piecewise concave function on $\mathbb{R}$. If we change the condition $s_t > 0$ to $s_t \neq 0$ in the definition of regeneration intervals, Theorem 3.1.1 and Theorem 3.1.2 still hold true in the case of backlogging. We can use the DP given in this section in order to solve this problem with some small modifications without changing the computational complexity.

In conclusion, for the special cases discussed above, our algorithm's performance is as good as the performance of algorithms in the literature.

Finally, note that the case where the inventory holding cost is a piecewise concave function can be handled similarly if we assume that the breakpoints of the holding cost function are also time-invariant and the number of breakpoints is fixed, i.e., $r_t^i = r^i$ for all $t = 1, \ldots, T$ and $i = 1, \ldots, q$ where $q_t = q$ for all $t = 1, \ldots, T$ and $q(\geq 1)$ is fixed. According to the redefinition of regeneration interval given in Remark 1, for each regeneration interval $[j, l]$ now we need to know the starting and ending inventories, $s_{j-1}, s_l \in \{r^1, \ldots, r^q\}$. Therefore, for each function defined for (regeneration) interval $[j, l]$ in the dynamic programming algorithm, additional initial (and final) inventory levels should be appended. For example, $F_{jl\alpha\beta}(t, \tau)$ will give the minimum cost for periods j up to t in a regeneration interval $[j, l]$ with initial inventory $s_{j-1} = r^\alpha$ and final inventory $s_l = r^\beta$ given that τ_i times b^i, for $i = 1, \ldots, m$, units are produced and no fractional production is done until period t. Moreover, for the last interval $[j, T]$ (which may not be a regeneration interval) as the final inventory level may not be equal to a breakpoint level, we need to define $\hat{F}_{j\alpha}(t, \tau)$ as the minimum cost for periods j up to t in the interval $[j, T]$ with $s_{j-1} = r^\alpha$ and $s_t \notin \{r^1, \ldots, r^q\}$ for $t = j, \ldots, T$ given that τ_i times b^i, for $i = 1, \ldots, m$, units are produced and no fractional production is done until period t. Similarly, functions $G_{jl\alpha\beta}$ and $\hat{G}_{jl\alpha\beta}$ will give the minimum costs for the regeneration interval $[j, l]$ with initial inventory $s_{j-1} = r^\alpha$

and final inventory $s_l = r^\beta$ when there exists exactly one fractional period.

3.3 Computational Results

In this section, we will examine the computational efficiency of our algorithm. Although our algorithm can solve the lot sizing problem with any piecewise concave function, in order to compare the algorithm's performance with a mixed-integer programming (MIP) solver, we use piecewise linear production cost functions and linear holding costs in our computational study.

We tested three well known linearizations of piecewise linear functions: multiple choice, incremental and convex combination formulations (see, e.g., Croxton et al. [96]). For the capacitated lot sizing problem, multiple choice linearization is as follows.

MC

$$\min \quad \sum_{t=1}^{T}\sum_{j=1}^{m}(f_t^j y_t^j + c_t^j x_t^j) + \sum_{t=1}^{T} h_t s_t \tag{3.1}$$

$$\text{s.t.} \quad s_{t-1} + \textstyle\sum_{j=1}^{m} x_t^j = d_t + s_t \qquad t = 1,\dots,T, \tag{3.2}$$

$$b^{j-1} y_t^j \le x_t^j \le b^j y_t^j \qquad t = 1,\dots,T, j = 1,\dots,m, \tag{3.3}$$

$$\textstyle\sum_{j=1}^{m} y_t^j \le 1 \qquad t = 1,\dots,T, \tag{3.4}$$

$$s_0 = 0, \tag{3.5}$$

$$s, x \ge 0, y \text{ binary}. \tag{3.6}$$

In this formulation, if the production amount is in the j^{th} piece of the cost function, then there is a fixed cost f_t^j and a variable cost c_t^j (see Figure 3.3). We assume that the production cost function is lower semicontinuous. The inventory holding cost function is a linear function and h_t is the cost of holding one unit of inventory during period t. The variable y_t^j is equal to one if the production amount in period t lies in the segment $[b^{j-1}, b^j]$. Constraints (3.4) ensure that at most one of the y_t^j variables is one in period t. Consequently, constraints (3.3) guarantee that x_t^j should be in the segment $[b^{j-1}, b^j]$ if $y_t^j = 1$, and at most one of the production variables x_t^j will be nonzero for t. Constraints (3.2) are inventory balance constraints and the objective function (3.1) is the sum of production and inventory holding costs. By constraints (3.5), we impose the requirement that the initial inventory is zero.

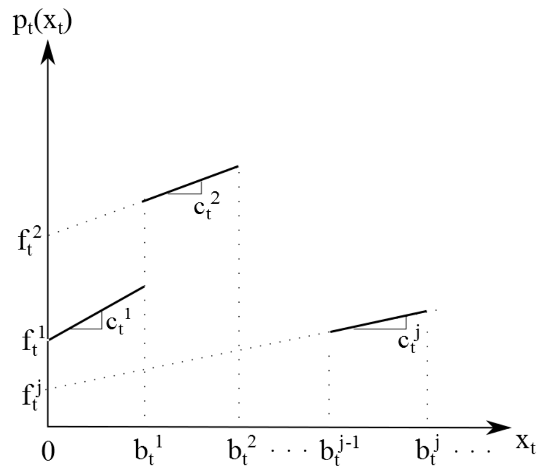

Figure 3.3: Production cost function for MC

The incremental formulation for the capacitated lot sizing problem is as follows.

$$\begin{aligned}
\min \quad & \sum_{t=1}^{T}\sum_{j=1}^{m}(\hat{f}_t^j y_t^j + c_t^j x_t^j) + \sum_{t=1}^{T} h_t s_t & \\
\text{s.t.} \quad & s_{t-1} + \textstyle\sum_{j=1}^{m} x_t^j = d_t + s_t & t = 1, \ldots, T, \\
& (b^j - b^{j-1}) y_t^{j+1} \le x_t^j \le (b^j - b^{j-1}) y_t^j & t = 1, \ldots, T,\ j = 1, \ldots, m \quad (3.7) \\
& s_0 = 0, & \\
& s, x \ge 0, y \text{ binary.} &
\end{aligned}$$

In the incremental formulation, production amount for each period is represented incrementally. In this formulation, x_t^j is the production amount for period t produced from the cost segment j. Thus, the total production for period t is given by $\sum_{j=1}^{m} x_t^j$. Note that if $y_t^{j+1} = 1$ then $x_t^{j+1} > 0$ and $x_t^j = b^j - b^{j-1}$. This is ensured by Constraints (3.7). In this formulation, production cost for period t is given by $\sum_{j=1}^{m}\left(\hat{f}_t^j y_t^j + c_t^j x_t^j\right)$ where $\hat{f}_t^j = f_t^j + c_t^j b^{j-1} - \left(f_t^{j-1} + c_t^{j-1} b^{j-1}\right)$.

In the convex combination formulation, if the production amount for period t lies on the cost segment j, then this amount is expressed as a convex combination of the breakpoints b^{j-1} and b^j. The convex combination formulation for the capacitated lot sizing problem is as follows.

$$
\begin{aligned}
\min\ & \sum_{t=1}^{T}\sum_{j=1}^{m}\left(a_t^j(f_t^j + c_t^j b^{j-1}) + d_t^j(f_t^j + c_t^j b^j)\right) + \sum_{t=1}^{T} h_t s_t \\
\text{s.t. } & s_{t-1} + \sum_{j=1}^{m}(a_t^j b^{j-1} + d_t^j b^j) = d_t + s_t \qquad t = 1,\ldots,T, \\
& a_t^j + d_t^j = y_t^j \qquad t = 1,\ldots,T,\ j = 1,\ldots,m \\
& \sum_{j=1}^{m} y_t^j \leq 1 \qquad t = 1,\ldots,T, \\
& s_0 = 0, \\
& s, a, d \geq 0, y \text{ binary.}
\end{aligned}
\tag{3.8}
$$

In this formulation, $y_t^j = 1$ if the production amount for period t lies in the cost segment j. If $y_t^j = 1$, then the production amount for period t is the convex combination of the breakpoints b^j and b^{j-1}: $a_t^j b^{j-1} + d_t^j b^j$ is the production amount for period t where $a_t^j + d_t^j = 1$. This is ensured by constraints (3.8). In this case, total production cost for period t is given by the convex combination of the costs of the breakpoints: $\sum_{j=1}^{m} a_t^j(f_t^j + c_t^j b^{j-1}) + d_t^j(f_t^j + c_t^j b^j)$.

Our preliminary tests showed that the multiple choice linearization outperforms the other two linearizations. We implemented the formulation MC in Xpress 1.22 and the DP in Java (JDK 7) and run them on a 2.53 GHz Intel Core 2 Duo Machine with 4GB memory running Windows 7. We let the solver run for 1000 seconds.

In our computational study, we only consider the capacitated problem and ignore the last piece of the cost function since we assume that it has a very large cost. We first analyze two cost segment instances, i.e., $m = 2$, and create randomly generated problems with different cost parameters, all time-invariant, as summarized in Table 3.1. Furthermore, for 40 and 50 period cases we assume that the demand has the same distribution and the holding cost is the same such that the inventory holding cost to be 0.05 and the demand to be an integer drawn from a uniform distribution, $U[400, 500]$. Consequently, for each case there are

Table 3.1: Experimental factors when $m = 2$

Factors		# levels	Experimental Settings			
Fixed Costs	$f = (f^1, f^2)$	3	1 (3000,6000)	2 (3000,4000)	3 (3000,7500)	
Variable Costs	$c = (c^1, c^2)$	4	1 (0,0)	2 (0.5,1)	3 (1,0.5)	4 (1,1)
Breakpoints	$b = (b^1, b^2)$	3	1 (800,1600)	2 (900,1800)	3 (1000,2000)	

Table 3.2: Experimental factors when $m = 3$

Factors	# levels	Experimental Settings			
f	3	1 (3000,6000,9000)	2 (3000,5000,6500)	3 (3000,3500,5000)	
b	2	1 (500,1000,1500)	2 (600,1200,1800)		
c	7	1 (0,0,0)	2 (1.3,1.5,1.8)	3 (1.3,1.8,1.5)	4 (1.5,1.3,1.8)
		5 (1.5,1.8,1.3)	6 (1.8,1.3,1.5)	7 (1.8,1.5,1.3)	

36 randomly generated test problems. We also generated instances as described in Hellion et al. [11]. However, we do not report the results for these instances as all were solved in less than one second by a commercial solver for $n = 40$.

For 20 periods and 3 cost segments instances, we consider different cost structures as summarized in Table 3.2. For example, increasing unit costs $(1.3, 1.5, 1.8)$ may represent a system with subcontracting, or decreasing unit costs $(1.8, 1.5, 1.3)$ may represent quantity discounts. Also, note that unit costs $(1.5, 1.3, 1.8)$ can be seen as a combination of these two systems (Figure 2.2a). We now generate 42 randomly generated problems for which we assume that the inventory holding cost to be 0.05 and the demand is an integer drawn from a uniform distribution, $U[500, 600]$.

We also consider instances with 15 periods ans 4 cost segments. We generate 24 instances where the inventory holding cost is 0.05 and the demand is an integer from a uniform distribution, $U[400, 500]$. Other experimental settings for these instances are given in Table 3.3.

Table 3.3: Experimental factors when $m = 4$

Factors	# levels	**Experimental Settings**	
f	2	1 (3000,6000,9000,12000)	2 (3000,5500,8000,10000)
b	3	1 (450,900,1350,1800)	2 (600,1200,1800,2400)
		3 (750,1500,2250,3000)	
c	4	1 (0.6,0.8,1.0,1.3)	2 (0.8,0.6,1.3,1.0)
		3 (1.0,0.6,0.8,1.3)	4 (1.3,0.6,0.8,1.0)

In order to improve the bounds obtained from the formulation MC, we use the valid inequalities developed by Sanjeevi and Kianfar [97] for the multi-module lot sizing problem. These inequalities are based on mixing set relaxations. We briefly describe these inequalities. Let $k < l$ be two periods and $S \subseteq \{k, \dots, l\}$. For each $i \in S$, define $S_i = S \cap \{k, \dots, i\}$ and compute

$$n_i = \begin{cases} \min\{t : t \in S \setminus S_i\} & \text{if } S \setminus S_i \neq \emptyset, \\ l + 1 & \text{if } S \setminus S_i = \emptyset. \end{cases}$$

Adding up the equations (3.2) from $t = k$ to $t = n_i - 1$, we obtain

$$s_{k-1} + \sum_{t=k}^{n_i-1} \sum_{j=1}^{m} x_t^j = d_{k,n_i-1} + s_{n_i-1}.$$

As $s_{n_i-1} \geq 0$ and $x_t^j \leq b^j y_t^j$, we have

$$s_{k-1} + \sum_{t \in \{k,\dots,n_i-1\} \setminus S_i} \sum_{j=1}^{m} x_t^j + \sum_{j=1}^{m} \sum_{t \in S_i} b^j y_t^j \geq d_{k,n_i-1}.$$

Now, let $I \subseteq S$. Then $s_{k-1} + \sum_{t \in \{k,\dots,n_{|I|}-1\} \setminus S} \sum_{j=1}^{m} x_t^j \geq s_{k-1} + \sum_{t \in \{k,\dots,n_i-1\} \setminus S_i} \sum_{j=1}^{m} x_t^j$ for all $i \in I$. By letting $z_i^j = \sum_{t \in S_i} y_t^j$, one obtains the relaxation

$$s_{k-1} + \sum_{t \in \{k,\dots,n_{|I|}-1\} \setminus S} \sum_{j=1}^{m} x_t^j + \sum_{j=1}^{m} b^j z_i^j \geq d_{k,n_i-1} \qquad i \in I.$$

Note that $s_{k-1} + \sum_{t \in \{k,\dots,n_{|I|}-1\} \setminus S} \sum_{j=1}^{m} x_t^j \in \mathbb{R}_+$ and $z_i^j \in \mathbb{Z}_+$ for all $i \in I$ and $j = 1, \dots, m$. Sanjeevi and Kianfar [97] generate mixed n-step MIR inequalities based on this relaxation when the coefficients satisfy some conditions. Like Sanjeevi and Kianfar [97], we consider all possible pairs k and l. We let $S = \{k, \ldots, l\}$, $S = \{t \in \{k, \ldots, l\} : \bar{y}_t^j > 0 \text{ for some } j \in \{1, \ldots, m\}\}$ and $S = \{t \in \{k, \ldots, l\} : \bar{y}_t^j < 1 \text{ for some } j \in \{1, \ldots, m\}\}$ where $(\bar{x}, \bar{y}, \bar{s})$ is the LP optimum. For these choices of S, we consider all possible two-element subsets I, i.e., $|I| = 2$, and add the resulting inequality if it is violated. We apply this cutting phase at the root node. Then we drop the inactive cuts and give the strengthened formulation to the solver.

In Tables 3.4-3.6, we report the results for the formulation MC, the formulation MC with valid inequalities (MC-CUTS) and our dynamic programming algorithm (DP). Columns BUB, LPGap, FGap correspond to the best upper bound obtained by the solver within the time limit, the percentage gap between the optimal value of the LP relaxation and the optimal value of the integer problem and the percentage gap between the best lower and upper bounds attained at the end of the time limit, respectively. Some instances are solved to optimality by MC or MC-CUTS; in this case we report the time spent to solve the formulation in parentheses in column (Time). Columns OPT and Time under DP correspond to the optimal value of the problem and the solution time of the dynamic programming algorithm.

We observe that none of the instances is solved to optimality using MC in 1000 seconds for 40 and 50 periods and 2 pieces instances and only 11 of the 42 instances of the 20 periods and 3 pieces instances are solved to optimality. As expected, the performance varies from one instance to another: the LPGap between 1 to 10% and the final gap between 0 to 5%. MC-CUTS can solve some instances in a second, whereas for others the final gap can be as large as 3-4%. Clearly, the DP has a stable solution time. Moreover, as shown in Tables 3.4 and 3.6, the proposed DP can handle all of these different cost functions and solves the problems to optimality whereas the MC formulation in Xpress may end up with an optimality gap of 3% at the end of the time limit of 1000 seconds.

It can be observed from Table 3.4 that for each setting of fixed and variable costs, increasing breakpoint levels increases the final gaps of MC. However, we

Table 3.4: Results for $T = 40$ and $m = 2$

instance			MC			MC-CUTS			DP	
f	c	b	BUB	LPGap	FGap	BUB	LPGap	FGap (Time)	OPT	Time
1	1	1	69645	2.61	1.6	69737	1.37	1.22	69620	160
		2	63556	5.05	3.96	63779	3.64	3.63	63475	161
		3	57745	5.91	4.35	57888	4.02	3.90	57652	153
	2	1	78676	2.31	1.29	78763	1.15	0.61	78661	163
		2	72589	4.42	3.30	73286	3.10	1.92	72515	158
		3	66835	5.11	3.72	66802	3.44	1.33	66692	158
	3	1	79678	3.51	1.29	79707	1.51	0.76	79642	162
		2	73610	5.71	3.22	73605	3.00	2.38	73505	151
		3	67908	6.79	3.47	67968	3.54	2.64	67891	146
	4	1	87701	2.07	1.24	87781	1.09	0.53	87701	162
		2	81676	3.93	3.13	81787	2.83	2.72	81556	155
		3	75873	4.50	3.37	75797	3.06	2.76	75733	148
2	1	1	48332	6.34	2.79	48260	1.29	(35)	48260	162
		2	44262	9.22	4.67	44280	2.43	0.73	44262	153
		3	40524	10.74	5.06	40519	2.97	0.81	40515	146
	2	1	65963	4.03	1.31	65941	1.19	(315)	65941	161
		2	61905	5.87	2.64	61924	2.35	1.41	61893	151
		3	58119	6.64	2.72	58142	2.84	1.58	58098	145
	3	1	57755	5.90	2.93	57642	0.78	(4)	57642	161
		2	53595	8.01	4.27	53505	1.17	(3)	53505	154
		3	49914	9.40	4.70	49891	1.48	(6)	49891	150
	4	1	66363	4.61	1.91	66341	0.94	(13)	66341	162
		2	62360	6.55	3.36	62347	1.73	0.43	62343	153
		3	58616	7.43	3.53	58609	2.06	0.53	58596	146
3	1	1	69620	2.61	1.45	69620	1.30	(1)	69620	161
		2	63492	5.05	3.61	63553	3.50	2.58	63475	150
		3	57730	5.91	4.21	57811	3.98	3.42	57652	148
	2	1	78661	2.31	1.22	78661	1.15	(1)	78661	159
		2	72567	4.42	3.29	72626	3.06	2.50	72515	151
		3	66722	5.11	3.57	66700	3.44	2.74	66692	146
	3	1	87701	2.07	1.12	87701	1.03	(1)	87701	161
		2	81618	3.93	2.93	82461	2.78	2.64	81556	152
		3	75817	4.50	3.24	76413	3.04	2.56	75733	145
	4	1	87723	2.07	1.18	87849	1.03	0.51	87701	162
		2	81573	3.93	2.78	81679	2.72	2.30	81556	151
		3	75831	4.50	3.25	75827	3.03	2.50	75733	145

Table 3.5: Results for $T = 50$ and $m = 2$

instance			MC			MC-CUTS			DP	
f	c	b	BUB	LPGap	FGap	BUB	LPGap	FGap (Time)	OPT	Time
1	1	1	87850	3.97	3.17	87831	2.77	2.69	87727	720
		2	76621	1.88	0.99	76314	0.24	(6)	76314	677
		3	70052	3.65	2.20	70300	1.64	1.74	69944	653
	2	1	99074	3.52	2.76	99045	2.40	1.26	98959	745
		2	87718	1.64	0.50	87546	0.19	(2)	87546	689
		3	81293	3.14	1.86	81254	1.40	0.95	81176	659
	3	1	100021	4.52	2.57	100401	2.73	2.39	99990	742
		2	89028	3.27	0.42	89026	0.96	(10)	89026	696
		3	82740	4.92	1.76	82696	2.21	0.42	82695	658
	1	1	110270	3.16	2.49	110246	2.21	2.04	110191	720
		2	99265	1.45	0.98	98778	0.19	(11)	98778	686
		3	92474	2.76	1.56	92575	1.24	1.06	92408	653
2	1	1	60592	7.23	4.01	60599	2.80	1.16	60538	737
		2	53386	6.43	1.88	53349	1.49	(5)	53349	689
		3	49068	8.38	3.02	49036	2.81	(114)	49036	661
	2	1	82666	4.82	2.46	82684	2.31	1.31	82602	736
		2	75490	3.95	0.76	75363	1.08	(8)	75363	682
		3	71028	5.08	1.37	71000	2.13	(735)	71000	648
	3	1	72015	6.39	3.61	71990	2.45	(523)	71990	744
		2	64886	5.72	1.94	64863	1.26	(7)	64863	672
		3	60749	7.47	3.05	60695	2.49	(12)	60695	642
	4	1	83055	5.27	2.92	83002	2.04	0.77	83002	737
		2	75850	4.52	1.29	75813	1.05	(8)	75813	674
		3	71511	5.74	1.98	71500	1.88	(84)	71500	651
3	1	1	87802	3.97	3.04	87782	2.68	0.59	87727	727
		2	76440	1.88	0.48	76314	0.22	(1)	76314	676
		3	70020	3.65	2.06	70044	1.63	1.07	69944	643
	2	1	98989	3.52	2.62	98959	2.37	(24)	98959	732
		2	87726	1.64	0.52	87546	0.19	(2)	87546	680
		3	81270	3.14	1.76	81391	1.40	1.07	81176	637
	3	1	110247	3.16	2.41	111021	2.16	1.99	110191	721
		2	98905	1.45	0.40	98778	0.17	(2)	98778	682
		3	92543	2.76	1.66	92529	1.23	0.84	92408	640
	4	1	110341	3.16	2.46	111866	2.13	2.02	110191	726
		2	98868	1.45	0.35	98778	0.17	(2)	98778	678
		3	92483	2.76	1.55	92589	1.23	0.94	92408	646

Table 3.6: Results for $T = 20$ and $m = 3$

instance			**MC**			**MC-CUTS**			**DP**	
f	c	b	BUB	LPGap	FGap (Time)	BUB	LPGap	FGap (Time)	OPT	Time
1	1	1	69160	3.85	3.40	69160	3.81	3.42	69160	173
		2	57167	3.01	2.37	57164	2.99	2.40	57137	150
	2	1	84084	3.26	2.35	84084	3.12	2.37	84084	174
		2	71545	2.41	0.94	71545	2.39	0.86	71545	150
	3	1	84216	3.55	2.26	84216	3.18	2.27	84216	172
		2	71545	2.41	0.42	71545	2.39	0.46	71545	151
	4	1	83842	3.50	2.62	83867	3.37	2.65	83836	171
		2	71903	2.89	1.62	71929	2.75	1.65	71903	151
	5	1	84107	3.81	2.57	84133	3.02	2.49	84107	171
		2	72194	3.28	1.47	72194	2.66	1.46	72194	151
	6	1	83914	3.57	2.56	83908	3.44	2.53	83901	177
		2	72018	3.05	1.52	72018	2.90	1.56	72018	155
	7	1	84153	3.86	2.45	84152	3.01	2.38	84152	173
		2	72307	3.44	1.31	72307	2.54	1.15	72307	154
2	1	1	51063	5.95	2.64	51064	2.87	1.38	51063	175
		2	42680	6.23	(182)	42680	3.00	(33)	42680	154
	2	1	70712	3.87	1.85	70712	2.42	1.57	70712	172
		2	62330	3.78	0.84	62330	2.36	0.57	62330	151
	3	1	67942	4.84	1.93	67942	1.72	(993)	67942	173
		2	59185	4.29	(28)	59185	2.21	(6)	59185	152
	4	1	70528	3.60	1.95	70512	2.67	1.89	70512	174
		2	61894	3.11	1.13	61900	2.59	1.18	61894	151
	5	1	65875	5.21	1.91	65866	1.43	(42)	65866	172
		1	57194	4.83	(20)	57194	2.16	(3)	57194	153
	6	1	67488	4.20	1.72	67488	2.42	1.25	67487	173
		2	59282	4.45	(984)	59282	2.36	(350)	59282	152
	7	1	65653	4.90	1.37	65652	1.95	0.33	65652	173
		2	57307	5.02	(116)	57307	1.95	(3)	57307	155
3	1	1	39064	5.43	2.22	39063	3.75	2.09	39063	174
		2	32633	5.66	(718)	32633	3.89	(410)	32633	150
	2	1	57393	3.45	1.80	57395	3.26	1.70	57393	173
		2	50557	3.18	(382)	50557	2.97	(624)	50557	153
	3	1	55942	4.24	0.76	55943	2.09	0.53	55942	172
		2	49927	5.04	(127)	49927	2.05	(14)	49927	154
	4	1	55410	3.99	1.88	55410	3.80	1.70	55410	172
		2	48651	3.94	(311)	48651	3.73	(360)	48651	151
	5	1	53866	4.67	(516)	53866	1.75	(40)	53866	172
		2	48133	6.11	(145)	48133	1.84	(3)	48133	151
	6	1	55051	3.36	1.96	55064	3.17	1.99	55051	175
		2	48190	3.01	0.50	48182	2.79	0.41	48182	150
	7	1	53652	4.29	1.09	53652	2.38	0.78	53652	172
		2	47493	4.84	(95)	47493	2.34	(9)	47493	151

cannot generalize this result since according to Table 3.5 for each combination of the fixed and variable costs, MC ends up with the largest gap when $(b^1, b^2) = (800, 1600)$, and with the minimum one when $(b^1, b^2) = (900, 1800)$. Note that, all of the instances with $(b^1, b^2) = (900, 1800)$ in Table 3.5 are solved to optimality by MC-CUTS. Moreover, according to the final gaps of MC given in Table 3.4, the instances with $(f^1, f^2) = (3000, 4000)$ seem like the hardest ones. However, interestingly, when the valid inequalities are added, 50% of these instances are solved to optimality and for the other ones with positive gaps the final gaps are relatively small compared to the other instances. A similar result can be obtained from Table 3.5.

Addition of valid inequalities to MC improves the LP gap in all of the instances. This improvement depends on the instance: LP gap may be decreased from 10.7% to 2.9% in one instance (in Table 3.4), but for another one the improvement may be quite small, like from 2.5% to 2% (in Table 3.5). Moreover, in Tables 3.4 - 3.5, (for the instances with positive final gap) none of the solutions found by MC-CUT at the end of time limit is optimal since the value of the best solution (BUB) is greater than the optimal value found by the DP.

For the instances with 3 pieces (Table 3.6), we can see that the difference between LP gaps of MC and MC-CUTS may be negligible as the improvement may be from 3.01% to 2.99% or from 2.41% to 2.39%. It can be observed from Table 3.6 that in 50% of the instances with positive final gaps MC finds optimal solutions but it cannot prove the optimality.

Similar results are obtained for the instances with 4 pieces but we only report the summary of the results for these instances in Table 3.7.

From Tables 3.4 - 3.6, we can see that for a given breakpoint level, for instances with different fixed and variable cost settings the solution time of the DP is stable. For different breakpoint levels the difference between solution times is also very small (less than 100 seconds). On the other hand, it is hard to obtain a general result for the MIP approach. The performance of the MIP approach strongly depends on the data instance; by small changes in instances, we may come up

Table 3.7: Summary of the results

(T,m)	MC						MC-CUTS					
	LPGap			FGap			LPGap			FGap		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
(40,2)	2.07	5.08	10.74	1.12	2.94	5.06	0.78	2.31	4.02	0.00	1.43	3.90
(50,2)	1.45	4.04	8.38	0.35	1.91	4.01	0.17	1.60	2.81	0.00	0.68	2.69
(20,3)	2.41	4.08	6.23	0.00	1.27	3.40	1.43	2.70	3.89	0.00	1.10	3.42
(15,4)	5.41	6.88	9.26	0.00	2.76	5.60	5.01	6.39	8.41	0.00	2.09	5.02

(T,m)	DP		
	Time		
	min	avg	max
(40,2)	145	154	163
(50,2)	637	688	745
(20,3)	150	162	177
(15,4)	400	421	446

with easier or harder instances.

A summary of the results is given in Table 3.7. In Table 3.7, columns named min, avg, max show the maximum, average and minimum values of the corresponding columns. As it can be observed from Table 3.7, when n or m increases, as expected, the solution time of DP gets larger. On the other hand, the DP solves all of the instances in less than 1000 seconds whereas Xpress may end up with positive optimality gaps even for the strengthened formulation.

We can conclude that for the small or medium sized instances, the DP outperforms the MIP approach. Furthermore, for solving larger instances of the problem we can easily modify the DP for getting good quality solutions in reasonable computation times as discussed below.

3.4 Heuristic for Solving Larger Instances

The computational complexity of our dynamic programming algorithm strongly depends on the number of different τ, π and $\hat{\pi}$ vectors since we need to evaluate the functions F_{jl}, G_{jl} and $\hat{G}_{jl}$ for all possible τ, π and $\hat{\pi}$ vectors. As there are $O(T^m)$ possible τ and $\hat{\pi}$ and $O(T^{m-1})$ π vectors, for larger T and m it may not

Table 3.8: Experimental factors for the heuristic solution approach when $m = 3$

Factors	# levels	**Experimental Settings**		
f	2	1 (3000,6000,9000)	2 (3000,5000,6000)	
b	2	1 (800,1600,2400)	2 (1000,2000,3000)	
c	3	1 (0,0,0)	2 (1,0.5, 0.7)	3 (1,0.5, 1)

be a good choice to use the DP directly. Moreover, as Xpress could not solve some medium sized instances in our experiments, we expect its performance to get worse for larger instances.

In order to get a good solution for larger instances in a reasonable time, we develop a heuristic method based on our dynamic programming algorithm. We heuristically restrict the length of any regeneration interval (and also the final interval which may not be a regeneration interval) of a solution. Let ν $(1 \leq \nu \leq T)$ be a given upper bound on the length of any regeneration interval. We consider the interval $[j, l]$, $1 \leq j \leq l \leq T$, and find the minimum cost μ_{jl} if $l - j + 1 \leq \nu$. Consequently, we reduce the number of intervals to be considered to $O(\nu T)$. Moreover, for a given interval $[j, l]$ the number of possible τ, π and $\hat{\pi}$ vectors become $O(\nu^m)$, $O(\nu^{m-1})$ and $O(\nu^m)$, respectively. Therefore, with this restriction we reduce the state space and consequently, the time complexity of the DP.

Note that when $\nu = T$, the restriction becomes redundant and the heuristic is the same as the exact DP. If $\nu = 1$, then the (trivial) solution is to produce in every period as much as the demand of that period. Moreover, if we know the maximum regeneration interval length in an optimal solution, say ν^*, then we can set $\nu = \nu^*$ and obtain an optimal solution to the problem with the heuristic. The performance of this heuristic depends on ν; we may obtain a better quality solution with larger ν but in longer computation time.

In order to test this solution method, we consider different ν values and compare the total cost of the solution obtained by this method with the lower bound obtained from MC-CUTS. We use larger instances that are created the same way as the instances used in the previous section. We have selected a representative set of instances to test the solution quality of the proposed heuristic. The experimental factors are listed in Table 3.8. For all of the instances, we assume that the inventory holding cost is 0.05 and the demand is an integer drawn from a uniform distribution, $U[400, 500]$ for all periods.

Tables 3.9 and 3.10 summarize the results of this experiment for $m = 2$ and $m = 3$, respectively. Columns under MC-CUTS represent the results for the formulation MC with valid inequalities, and the columns under DP-HEUR represent the results of our heuristic method. For each instance, we consider different ν values in order to see the trade-off between the solution quality and the solution time and we sign the rows with the minimum optimality gap by bold. With MC-CUTS, we let Xpress run 1000 and 2000 seconds and we calculate the gap of the heuristic solution using the best lower bound obtained in 1000 seconds. We also report the CPU times at which the best upper bound and the best optimality gap are attained, in parenthesis under the corresponding values.

As it can be seen from Tables 3.9 and 3.10, letting Xpress run for an additional 1000 seconds results in very little improvement in the final gaps. When the cost function has two pieces (Table 3.9), in all of the test instances, the heuristic finds better solutions than MC-CUTS in less than 50 seconds. Moreover, as it can be revealed from the table, when ν increases, the computation time increases (as expected) but the increase is not too fast. Therefore, the user can select a higher ν value and may obtain better solutions in reasonable computation times.

In Table 3.10, we report the results for the instances with 3 pieces. For 50 and 80 periods, the heuristic finds better solutions than MC-CUTS in very short computation times. For 100 periods, we again find better solutions by the heuristic algorithm but the computation time of the algorithm is about 2000 seconds. But note that for the second instance of 100 periods solution found for $\nu = 18$ (in less than 1000 seconds of time) is also a better solution than that of MC-CUTS. Moreover, we believe that by letting Xpress run for more than 2000 seconds we can only obtain slightly better optimality gaps. Thus, when $m = 3$ the heuristic algorithm still reports better solutions compared to the MIP approach in less

Table 3.9: Results of the heuristic for $m = 2$

T	instance			MC-CUTS				DP-HEUR			
				1000 seconds		**2000 seconds**					
T	f	c	b	BUB	Gap	BUB	Gap	ν	BUB	Gap	Time
80	1	1	3	112909	2.20	112909	2.19	10	118081	6.48	1
				(609)	(643)		(1350)	12	112493	1.84	2
								14	**112488**	**1.83**	**3**
								16	112488	1.83	6
								18	112488	1.83	11
								20	112488	1.83	20
								22	112488	1.83	32
	1	4	1	175495	0.36	175495	0.35	10	175376	0.29	1
				(931)	(931)		(1226)	12	175351	0.28	2
								14	175339	0.27	3
								16	175339	0.27	7
								18	175339	0.27	13
								20	**175324**	**0.26**	**21**
								22	175324	0.26	35
100	1	1	2	154876	1.22	154672	1.08	10	157515	2.87	1
				(44)	(735)	(1313)	(1313)	12	157298	2.74	2
								14	157277	2.73	4
								16	157251	2.71	8
								18	157251	2.71	16
								20	154558	1.02	27
								22	**154498**	**0.98**	**44**
	1	4	1	219163	1.05	218739	0.85	10	220697	1.73	1
				(781)	(781)	(1641)	(1641)	12	220694	1.73	2
								14	220687	1.73	4
								16	220687	1.73	8
								18	220687	1.73	16
								20	**217919**	**0.48**	**27**
								22	217913	0.48	46

Table 3.10: Results of the heuristic for $m = 3$

T	**instance** *f*	*c*	*b*	**MC-CUTS** **1000 seconds** BUB	Gap	**2000 seconds** BUB	Gap	**DP-HEUR** ν	BUB	Gap	Time
50	1	1	2	70146	2.33	70146	2.31	10	72742	5.81	9
				(355)	(971)		(1742)	12	69948	2.05	27
								14	**69944**	**2.04**	**74**
								16	69944	2.04	176
								18	69944	2.04	380
								20	69944	2.04	763
								22	69944	2.04	1448
	2	3	1	83143	1.73	83126	1.68	10	83433	2.07	9
				(310)	(885)	(1495)	(1737)	12	83433.3	2.07	27.6
								14	83431	2.07	75
								16	**83041**	**1.61**	**180**
								18	83041	1.61	394
								20	83041	1.61	806
								22	83041	1.61	1508
80	1	1	2	105500	1.12	105473	1.06	10	110781	5.83	14
				(253)	(903)	(1475)	(1779)	12	108170	3.56	46
								14	105433	1.06	125
								16	105433	1.06	306
								18	**105430**	**1.05**	**676**
								20	105430	1.05	1397
								22	105430	1.05	2701
	2	2	2	112940	2.67	112940	2.66	10	118081	6.90	14
				(670)	(903)		(1032)	12	112493	2.28	49
								14	**112488**	**2.27**	**125**
								16	112488	2.27	307
								18	112488	2.27	682
								20	112488	2.27	1402
								22	112488	2.27	2729
100	1	1	1	173543	1.43	173543	1.42	10	175486	2.52	19
				(424)	(424)		(1620)	12	175483	2.52	60
								14	175476	2.51	166
								16	175476	2.51	414
								18	175476	2.51	919
								20	**172708**	**0.95**	**1916**
								22	172702	0.95	3760
	2	2	2	131320	0.93	131320	0.91	10	137835	5.61	18
				(442)	(892)		(1595)	12	135444	3.95	58
								14	132401	1.74	160
								16	132401	1.74	394
								18	131275	0.90	883
								20	**131235**	**0.87**	**1827**
								22	131235	0.87	3576

computation times. Furthermore, according to Tables 3.9 and 3.10, similar to the exact DP, for given T, m and ν values, the computation time of the heuristic algorithm is stable.

3.5 Conclusions

In this chapter, we studied the lot sizing problem with piecewise concave production costs. A piecewise concave function can represent economies of scale, discounts, subcontracting, overloading, minimum order quantities and capacities. The computational complexity of this problem was an open question in the literature. We developed a dynamic programming algorithm and showed that the problem is polynomially solvable when the number of breakpoints of the production cost function is fixed and the breakpoints are time invariant. The algorithm performs well for small and medium sized instances and can easily be modified to be used as a heuristic for larger instances.

As it is expected and it can be observed from our computational study, when m increases the solution time of the DP increases rapidly. For example, when $m = 5$, the solution time of the (exact) DP is about 100 seconds for $T = 6$, 350 seconds for $T = 6$, and 1000 seconds for $T = 8$. Therefore, a different approach is required to solve problems with larger number of breakpoints.

It may also be interesting to consider the problem when one of the pieces of the production cost function is convex (but not linear), which means that the function is not piecewise concave. A convex function can indicate increasing marginal costs, therefore convex part of this function may represent overloading or cost of extra usage of a resource. This will be the topic of our next chapter.

Chapter 4

Stochastic Lot Sizing Problem with Controllable Processing Times

In this chapter, we consider the lot sizing problem with controllable processing times where demand follows a stochastic process and processing times of jobs can be controlled in return for extra cost (compression cost). This study is published in OMEGA The International Journal of Management Science [98].

In our study, we consider the following convex compression cost function for period t: $\gamma_t(k_t) = \kappa_t k_t^{a/b}$ where $k_t > 0$ is the total compression amount in period t, $\kappa_t \geq 0$ and $a \geq b > 0$, $a, b \in Z_+$. Note that, for $a > b$ and $\kappa_t > 0$, γ_t is strictly convex. This function can represent increasing marginal cost of compressing processing times in larger amounts. Moreover, this function can be related to a (convex) resource consumption function ([67], [69]). Suppose that one additional unit of the resource costs κ_t and for compressing the processing time by k_t units, additional $k_t^{a/b}$ units of resource should be allocated. Thus, in this context, compression cost represents resource consumption cost and the resource may be a continuous nonrenewable resource such as energy, fuel or catalyzer.

The rest of this chapter is organized as follows. In the next section, we briefly review the related literature. In Section 4.2, we formulate the problem and in Section 4.3, we strengthen the formulation using the second order conic strengthening. In Section 4.4, we present the results of our computational experiments. We first compare alternative conic formulations presented in Section 4.3, afterwards we investigate the impact of controllable processing times on production costs. In Section 4.5, conclusions and future research directions are discussed.

4.1 Problem Definition and Formulations

We consider the stochastic capacitated lot sizing problem with service level constraints and controllable processing times. We assume that the demand of each period is independent from each other and normally distributed with mean θ_t and standard deviation σ_t for period $t = 1, \ldots, T$, where T is the length of the planning horizon. We denote the demand of period t by d_t. We allow backlogging but assume that all the shortages are satisfied as soon as a supply is available. We restrict this case by using α service level constraints, where α corresponds to the probability of no stock out in a period. We assume that the resource is capacitated and capacity of period t in terms of time units is indicated by C_t. Processing time of an item is v_t time units, but we can reduce (compress) it in return for extra cost (compression cost). The processing time of an item can be reduced by at most u_t ($< v_t$) time units. We assume that all the production decisions are made at the beginning of the planning horizon. The problem is to find a production plan that satisfies the minimum service level constraints and minimizes the total production, compression and inventory costs.

Let x_t be the production amount in period t, $y_t = 1$ if there is a setup in period t and 0 otherwise, and s_t be the inventory on hand at the end of period t. We define $\gamma_t : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ as the compression cost function and k_t as the total compression amount (reduction in processing time) in period t. We assume that γ_t is a convex function, and the production cost function is composed of setup cost f_t and unit production cost c_t. Let h_t be the unit inventory holding costs for

period t. The problem can be formulated as the following:
SLS-I

$$\min \quad \sum_{t=1}^{T} \left(f_t y_t + c_t x_t + h_t \mathbb{E}\left[\max\{s_t, 0\}\right] + \gamma_t\left(k_t\right)\right) \tag{4.1}$$

$$\text{s.t.} \quad s_t = \sum_{i=1}^{t} x_i - \sum_{i=1}^{t} d_i \qquad t = 1, \ldots, T, \tag{4.2}$$

$$\Pr\left\{s_t \geq 0\right\} \geq \alpha \qquad t = 1, \ldots, T, \tag{4.3}$$

$$v_t x_t - k_t \leq C_t y_t \qquad t = 1, \ldots, T, \tag{4.4}$$

$$k_t \leq u_t x_t \qquad t = 1, \ldots, T, \tag{4.5}$$

$$x_t, k_t \geq 0 \qquad t = 1, \ldots, T, \tag{4.6}$$

$$y_t \in \{0, 1\} \qquad t = 1, \ldots, T. \tag{4.7}$$

In constraints (4.2), inventory at the end of each period is expressed. Note that we assume the initial inventory is zero. If this is not the case, we can easily add s_0 to the right hand side of constraint (4.2). The probability expressed in constraint (4.3) is the probability that no stock-out occurs in period t and this should be greater than or equal to α. Constraint (4.4) is the capacity constraint: if x_t units are produced in period t, $v_t x_t$ time units are necessary for production without any compression, but if this is larger than the capacity C_t, then we need to reduce the processing times by $k_t = C_t - v_t x_t$ in total. Since processing time of a unit cannot be reduced more than u_t time units and x_t units are produced in period t, total compression amount k_t should be less than or equal to $u_t x_t$, and this is ensured by (4.5).

In our problem, since d_t is a random variable (with known distribution), s_t is also a random variable. Therefore, from constraint (4.2), expected inventory at the end of each period can be obtained as $\mathbb{E}\left[s_t\right] = \sum_{i=1}^{t} x_i - \sum_{i=1}^{t} \mathbb{E}\left[d_i\right]$, $t = 1, \ldots, T$.

Let $G_{d_{1t}}$ be the cumulative probability distribution of the cumulative demand up to period t, which is denoted by $d_{1t} = \sum_{i=1}^{t} d_i$. Since demand of each period is independent from each other, d_{1t} is normally distributed with mean $\theta_{1t} = \sum_{i=1}^{t} \theta_i$ and standard deviation $\sigma_{1t} = \sqrt{\sum_{i=1}^{t} \sigma_i^2}$. Therefore, we can rewrite the α service level constraint (4.3) as

$$
\begin{aligned}
\Pr\left\{s_t \geq 0\right\} &= \Pr\left\{\sum_{i=1}^{t} x_i \geq \sum_{i=1}^{t} d_i\right\} \\
&= G_{d_{1t}}\left(\sum_{i=1}^{t} x_i\right) \geq \alpha \Leftrightarrow \sum_{i=1}^{t} x_i \geq G_{d_{1t}}^{-1}(\alpha) \\
&\qquad\qquad\qquad\qquad \Leftrightarrow \sum_{i=1}^{t} x_i \geq Z_\alpha \sigma_{1t} + \theta_{1t} \qquad (4.8)
\end{aligned}
$$

since the inverse cumulative probability of d_{1t} is $G_{d_{1t}}^{-1} = Z_\alpha \sigma_{1t} + \theta_{1t}$ where Z_α represents the α-quantile of the standard normal distribution [37]. Note that inequality (4.8) is similar to demand satisfaction constraint of the classical lot sizing problem. Let $\hat{d}_1 = Z_\alpha \sigma_{11} + \theta_1$ and $\hat{d}_t = Z_\alpha\left(\sigma_{1t} - \sigma_{1(t-1)}\right) + \theta_t$ for $t = 2, \ldots, T$ be the new demand parameters and suppose $\hat{s}$ denotes the new stock variables. Then, (4.8) can be expressed as

$$
\begin{aligned}
\hat{s}_{t-1} + x_t &= \hat{d}_t + \hat{s}_t && t = 1, \ldots, T, && (4.9) \\
\hat{s}_0 &= 0, && && (4.10) \\
\hat{s}_t &\geq 0 && t = 1, \ldots, T. && (4.11)
\end{aligned}
$$

Finally, as we assume α is sufficiently large and shortages are fulfilled as soon as a supply is available, we can approximate the expected total inventory cost as done in Bookbinder and Tan [37]:

$$
\begin{aligned}
\sum_{t=1}^{T} h_t \mathbb{E}\left[\max\{s_t, 0\}\right] &\approx \sum_{t=1}^{T} h_t \left(\sum_{i=1}^{t} x_i - \sum_{i=1}^{t} \mathbb{E}\left[d_i\right]\right) \\
&= \sum_{t=1}^{T} \overline{h}_t x_t - h_t \theta_{1t}
\end{aligned}
$$

where $\overline{h}_t = \sum_{j=t}^{T} h_j$. Let $\overline{c}_t = c_t + \overline{h}_t$, then we can remove the original inventory variables s_t from the formulation SLS-I and rewrite the objective function (4.1) as

$$
\sum_{t=1}^{T} \left(f_t y_t + \overline{c}_t x_t + \gamma_t\left(k_t\right)\right). \qquad (4.12)
$$

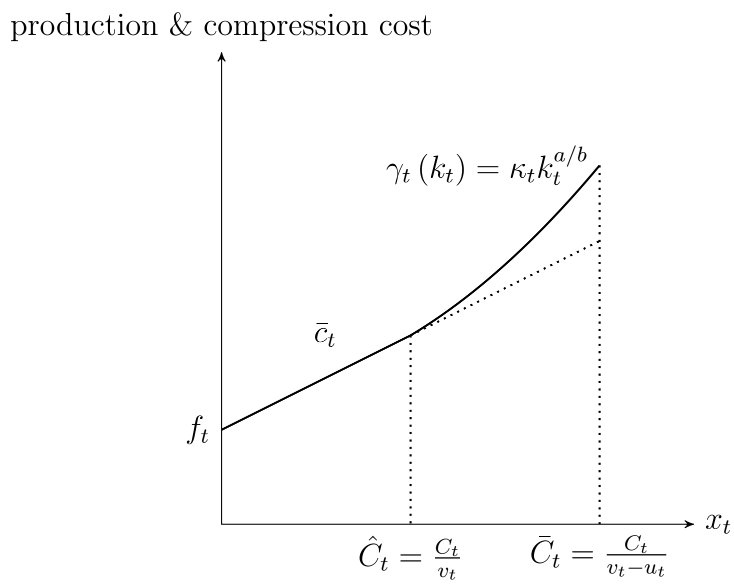

Figure 4.1: Total production and compression cost function

Now consider the capacitated deterministic lot sizing problem. It is known that when the production and inventory holding cost functions are concave, the lot sizing problem has an optimal solution that is composed of consecutive regeneration intervals and in each of these intervals there exists at most one fractional period [9]. Most of the dynamic programming algorithms developed for variations of the lot sizing problem use modifications of this property. The reader is referred to Pochet and Wolsey [99] for more details. As it can be observed from the following example, this property does not hold for our problem as our production cost function is not concave.

Example 4.1.1. *Consider the following problem instance:* $T = 3$, $f_t = 100$, $c_t = 0$, $h_t = 1$, $\kappa_t = 0.25$, $C_t = 20$, $v_t = 1$, $u_t = 0.5$ *for* $t = 1, \ldots, T$, $a/b = 2$ *and* $\hat{d} = (10, 20, 10)$. *Optimal solution to the problem is* $x^* = (18, 22, 0)$, $s^* = (8, 10, 0)$ *and* $k^* = (0, 2, 0)$ *with total cost 219. This solution is composed of one regeneration interval* $[1, 3]$ *and both of the production periods in this interval are fractional if the capacity is assumed as 20/(1-0.5) = 40. Thus, the regeneration interval property of the classical lot sizing problem does not hold for this problem.*

Note that, the total production and compression cost function for each period has two breakpoints $\hat{C}_t = \frac{C_t}{v_t}$ and $\overline{C}_t = \frac{C_t}{v_t - u_t}$. The first segment $[0, \hat{C}_t]$ corresponds to the regular production cost and the second segment $[\hat{C}_t, \overline{C}_t]$ corresponds to the cost of production with compression (see Figure 4.1). If $\hat{C}_t$ are time dependent then the problem is NP-Hard, since the classical lot sizing problem with arbitrary capacities is a special case of our problem (the case with $u_t = 0$ for all t). If $\hat{C}_t = C^1$ and $\overline{C}_t = C^2$ for $t = 1, \ldots, T$, and $a/b = 1$, then the problem is a lot sizing problem with piecewise linear production costs and it can be solved in polynomial time due to the dynamic programming algorithm developed in Chapter 3. When $a/b > 1$, as the compression cost function is convex and there exist setup costs, it is unlikely to find a polynomial time algorithm for solving the problem since even the uncapacitated lot sizing problem with convex production cost functions and unit setup costs is NP-Hard [12]. Besides, if the compression cost function is piecewise linear and convex, then the total production cost function is also piecewise linear and any formulation for the piecewise linear functions (multiple choice, incremental, convex combination (see, e.g., [100]) or the (pseudo-polynomial time) algorithm of Shaw and Wagelmans [25] can be used. Moreover, as it is stated above, if the breakpoints of the total production cost function is time invariant and the number of breakpoints is fixed, then the problem is polynomially solvable due to the dynamic programming algorithm developed in Chapter 3.

4.2 Reformulations

Now we need to examine the compression cost function $\gamma_t(.)$. There is not much done on this class of lot sizing problems with convex production cost functions, since most of the optimality properties are not valid for this particular case as demonstrated in Example 4.1.1. Still, as it is shown in this section, the problem we study has some nice structure that we could use to strengthen the formulation.

Assume that compression cost function for period t is given by $\gamma_t(k_t) = \kappa_t k_t^{a/b}$ where $k_t > 0$ is the total compression amount in period t, $\kappa_t \geq 0$ and $a \geq b > 0$, $a, b \in Z_+$ ($k_t = \max\{0, v_t x_t - C_t\}$). In order to formulate this case, as done in Aktürk et al. [63], we introduce auxiliary variables r_t, add the following inequalities

$$k_t^{a/b} \leq r_t \qquad t = 1, \ldots, T, \tag{4.13}$$

and replace $\gamma_t(k_t)$ with $\kappa_t r_t$ in the objective function (4.12). As $b > 0$, we can rewrite (4.13) as

$$k_t^a \leq r_t^b \qquad t = 1, \ldots, T.$$

Therefore, we could reformulate the problem as follows:

SLS-II

$$\begin{aligned}
\min \quad & \sum_{t=1}^{T} (f_t y_t + \bar{c}_t x_t + \kappa_t r_t) & \\
\text{s.t.} \quad & \hat{s}_{t-1} + x_t = \hat{d}_t + \hat{s}_t & t = 1, \ldots, T, \\
& v_t x_t - k_t \leq C_t y_t & t = 1, \ldots, T, \\
& k_t \leq u_t x_t & t = 1, \ldots, T, \\
& k_t^a \leq r_t^b & t = 1, \ldots, T, \\
& \hat{s}_0 = 0, & \\
& x_t, k_t, r_t, \hat{s}_t \geq 0 & t = 1, \ldots, T, \\
& y_t \in \{0, 1\} & t = 1, \ldots, T.
\end{aligned} \tag{4.14}$$

Moreover, as it is done in Aktürk et al. [63], we can strengthen inequality (4.14) as

$$k_t^a \leq r_t^b y_t^{a-b} \qquad t = 1, \ldots, T. \tag{4.15}$$

Note that if there is no production in period t, then $y_t = 0$ and there will be no need for compression; thus, $k_t = 0$. On the other hand, if $y_t = 1$, then inequality (4.15) reduces to (4.14).

We will refer to the strengthened version of SLS-II (the set of constraints (4.14) is replaced with the set of constraints (4.15)) as SLS-III. Now we will show that this strengthening gives the convex hull of the set

$$S = \{(x, k, r, y) \in \mathbb{R}_+^3 \times \{0, 1\} : \ k^{a/b} \leq r, \ k \leq ux, \ vx - k \leq Cy\},$$

where the subscripts are dropped for ease of presentation. Set S can be seen as a single period relaxation that involves only the production, setup and compression variables associated with a given period. Our hope is that having a strong formulation for set S may be useful in solving the overall problem. The computational results presented in the next section show that this strengthening is indeed useful.

Let

$$S' = \{(x,k,r,y) \in \mathbb{R}^4_+ : \ k^a \le r^b y^{a-b}, \ k \le ux, \ vx - k \le Cy, \ 0 \le y \le 1\}.$$

Proposition 4.2.1. *S' is the convex hull of S, i.e., $conv(S) = S'$.*

Proof. First, we will show that $conv(S) \subseteq S'$. Consider $(x_1,k_1,r_1,y_1),\ (x_2,k_2,r_2,y_2) \in S$. Note that, if $y_1 = y_2$, then convex combination of these points is in $S \subseteq S'$. Thus, suppose that $y_1 = 0$ (and consequently, $x_1 = k_1 = 0$) and $y_2 = 1$. Consider the convex combination of these points:

$$\begin{aligned}(x,k,r,y) &= (1-\lambda)(0,0,r_1,0) + \lambda(x_2,k_2,r_2,1) \\ &= (\lambda x_2, \lambda k_2, (1-\lambda)r_1 + \lambda r_2, \lambda)\end{aligned}$$

for $\lambda \in [0,1]$. Note that, $0 \le y = \lambda \le 1$, $vx - k = \lambda(vx_2 - k_2) \le \lambda C = Cy$, and $k = \lambda k_2 \le \lambda u x_2 = ux$. Finally,

$$\begin{aligned}k^a = (\lambda k_2)^a = \lambda^b k_2^a \lambda^{a-b} &= ((1-\lambda)0 + \lambda k_2^{a/b})^b \lambda^{a-b} \\ &\le ((1-\lambda)r_1 + \lambda r_2)^b \lambda^{a-b} = r^b y^{a-b}.\end{aligned}$$

Thus, $(x,k,r,y) \in S'$.

Now, we will show that $S' \subseteq conv(S)$. Consider $(x,k,r,y) \in S'$. Note that, if $y \in \{0,1\}$, then $(x,k,r,y) \in S \subseteq conv(S)$. Thus, assume that $0 < y < 1$. Then, (x,k,r,y) can be expressed as a convex combination of $(0,0,0,0) \in S$ and $(\frac{x}{y}, \frac{k}{y}, \frac{r}{y}, 1)$ with coefficients $1-\lambda$ and $\lambda = y \in (0,1)$, respectively. As $(x,k,r,y) \in S'$, $v\frac{x}{y} - \frac{k}{y} \le C$, $\frac{k}{y} \le u\frac{x}{y}$, and

$$k^a \le r^b y^{a-b} \Rightarrow k^{a/b} \le r y^{a/b-1} \Rightarrow \left(\frac{k}{y}\right)^{a/b} \le \frac{r}{y}.$$

Consequently, $(\frac{x}{y}, \frac{k}{y}, \frac{r}{y}, 1) \in S$ and $(x,k,r,y) \in conv(S)$. □

Now, we will reformulate constraint (4.15) using conic quadratic inequalities. As given in Ben-Tal and Nemirovski [101], for a positive integer l, and $\varepsilon, \pi_1, \ldots, \pi_{2^l} \geq 0$,

$$\varepsilon^{2^l} \leq \pi_1, \ldots, \pi_{2^l}, \tag{4.16}$$

can be represented by using $O(2^l)$ variables and $O(2^l)$ hyperbolic inequalities of the form

$$w^2 \leq w_1 w_2 \tag{4.17}$$

where $w, w_1, w_2 \geq 0$. Moreover, inequality (4.17) is conic quadratic representable:

$$\left\| \begin{pmatrix} 2w \\ w_1 - w_2 \end{pmatrix} \right\| \leq w_1 + w_2. \tag{4.18}$$

Using these results, one can show that for given t, $a \geq b > 0$ and $a, b \in \mathbb{Z}_+$, inequality (4.15) can be represented by $O(log_2(a))$ variables and conic quadratic constraints of the form (4.18) [63]. Note that if we fix $y_t = 1$, then we obtain (4.14), thus these constraints are also conic quadratic representable. We will refer to the conic quadratic formulations of SLS-II and SLS-III as CSLS-II and CSLS-III, respectively.

In CSLS-II and CSLS-III, for each period t, inequalities (4.14) and (4.15) are replaced with their conic quadratic representations. Therefore, these formulations are quadratically constrained MIP's (MIQCP) with linear objective functions that can be solved by fast algorithms of commercial MIQCP solvers like IBM ILOG CPLEX. In the next example, we illustrate the generation of conic quadratic constraints.

Example 4.2.1. *Our compression cost for period t is given by $\gamma_t(k_t) = \kappa_t k_t^{a/b}$. We first introduce auxiliary variable r_t, add inequality $k_t^{a/b} \leq r_t$ to the formulation and replace $\gamma_t(k_t)$ by $\kappa_t r_t$ in the objective function. Suppose that $a = 5$ and $b = 2$. Then, for period t, we have inequality $k_t^{5/2} \leq r_t$, which can be rewritten as $k_t^5 \leq r_t^2$. By strengthening the latter inequality, we obtain $k_t^5 \leq r_t^2 y_t^3$ and it is equivalent to*

$$k_t^8 \leq r_t^2 y_t^3 k_t^3. \tag{4.19}$$

This inequality can be expressed with the following four inequalities where three new nonnegative auxiliary variables $w_{1t}, w_{2t}, w_{3t} \geq 0$ *are introduced:*

$$\begin{aligned} w_{1t}^2 &\leq r_t y_t, \\ w_{2t}^2 &\leq y_t k_t, \\ w_{3t}^2 &\leq w_{2t} k_t, \\ k_t^2 &\leq w_{1t} w_{3t}. \end{aligned}$$

Figure 4.2 illustrates the generation of these inequalities.

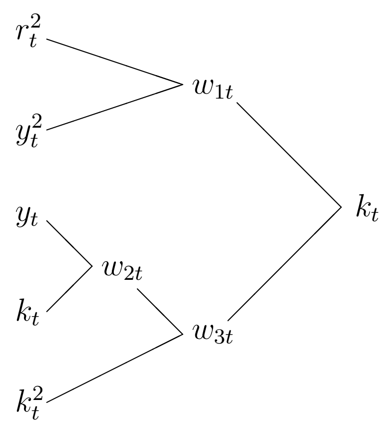

Figure 4.2: Illustration of generation of conic quadratic inequalities

These constraints can be represented by the following conic quadratic inequalities:

$$\begin{aligned} 4w_{1t}^2 + (r_t - y_t)^2 &\leq (r_t + y_t)^2, \\ 4w_{2t}^2 + (y_t - k_t)^2 &\leq (y_t + k_t)^2, \\ 4w_{3t}^2 + (w_{2t} - k_t)^2 &\leq (w_{2t} + k_t)^2, \\ 4k_t^2 + (w_{1t} - w_{3t})^2 &\leq (w_{1t} + w_{3t})^2. \end{aligned}$$

Consequently, for a given period t*, each inequality (4.19) is represented by four conic quadratic inequalities and additional three nonnegative variables* $w_{1t}, w_{2t}, w_{3t} \geq 0$*. These inequalities can be easily input to a MIQCP solver.*

Table 4.1: Parameter settings for the first part

Parameter	Explanation	# of levels	Parameter settings		
			1	2	3
α	Service level	2	0.95	0.98	–
f	Setup cost	3	1750	3500	7000
κ	Compression cost coefficient	2	0.1	0.3	–
$C/(\theta v)$	Avg. capacity tightness	2	3	5	–
β	$\theta_t \sim U\left[(1-\beta)\theta, (1+\beta)\theta\right]$	1	0.1	–	–

4.3 Computational Experiments

In this section, first we will test the effect of strengthening (4.14) by performing a computational experiment for comparing formulations CSLS-II and CSLS-III. Then, we will investigate the effect of controllable processing times in terms of cost reduction by comparing optimal costs of the system with and without controllable processing times. In our computational experiments, we consider quadratic and cubic compression cost functions $\gamma_t(k_t) = \kappa_t k_t^2$ and $\gamma_t(k_t) = \kappa_t k_t^3$. We implement all the formulations in IBM ILOG CPLEX 12.5 and perform the experiments on a 2.4 GHz Intel Core i7 Machine with 16GB memory running Windows 8.

4.3.1 Comparison of formulations

In the first part of our study, we consider the data sets for $T = 50$ periods and with time-invariant parameters. Therefore, we delete the subscript t from the parameters. We assume that unit inventory holding cost (h) is 1, unit production cost (c) is 0, capacity of a period in terms of time units (C) is 300, production time without any compression (v) is 1, maximum possible compression amount (u) for a unit is 30% of the processing time and coefficient of variation (hereafter CV) is 10%. We determine the rest of the parameters according to the values given in Table 4.1. We set time limit as 2000 seconds.

Most of the commercial solvers, such as IBM ILOG CPLEX, can solve MIP formulations with a quadratic objective function. Therefore, we also use formulation

SLS-Q where we keep the quadratic compression cost function in the objective. We note that SLS-Q is the same as SLS-II except that $\kappa_t r_t$ is replaced by $\kappa_t k_t^2$ in the objective function, constraints (4.14) and variables r_t, for $t = 1, \ldots, T$, are removed. We solve SLS-Q by CPLEX MIQP. Note that for the quadratic compression cost function, conic reformulations CSLS-II and CSLS-III are equivalent to SLS-II and SLS-III, respectively. Thus, performance differences of SLS-Q and CSLS-II will show the effect of having quadratic terms in the objective function and in the constraints. The effect of proposed conic strengthening can be observed by comparing CSLS-II and CSLS-III.

SLS-Q

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} \left(f_t y_t + \bar{c}_t x_t + \kappa_t k_t^2 \right) \\
\text{s.t.} \quad & \hat{s}_{t-1} + x_t = \hat{d}_t + \hat{s}_t && t = 1, \ldots, T, \\
& v_t x_t - k_t \leq C_t y_t && t = 1, \ldots, T, \\
& k_t \leq u_t x_t && t = 1, \ldots, T, \\
& \hat{s}_0 = 0, \\
& x_t, k_t, \hat{s}_t \geq 0 && t = 1, \ldots, T, \\
& y_t \in \{0, 1\} && t = 1, \ldots, T.
\end{aligned}
$$

Results of this experiment are given in Tables 4.2 and 4.5. In these tables, the percentage gap between the continuous relaxation at the root node and the optimal solution (rgap) (root gap, hereafter) and the number of branch-and-bound nodes explored are reported. If the solver is terminated due to the time limit, final gap is given under the column (gap), otherwise solution time is reported (cpu).

Results of this experiment for quadratic compression cost function are given in Table 4.2. This table clearly indicates that CSLS-III outperforms CSLS-II both in terms of root gap and solution time. Note that the root gap of CSLS-II is twice as large as of the one of CSLS-III for some instances. Moreover, all the instances are solved to optimality in less than 800 seconds by CSLS-III (average solution

Table 4.2: Effect of strengthening - quadratic compression cost

Parameters				SLS-Q			CSLS-II			CSLS-III		
α	f	κ	$\frac{C}{\theta v}$	rgap	cpu (gap)	node #	rgap	cpu (gap)	node #	rgap	cpu (gap)	node #
2	1	1	1	5.43	96	2,359,855	5.43	(0.11)	23,100,921	3.7	473	5,787,570
			2	9.35	36	847,454	9.35	331	5,375,748	6.55	71	978,035
		2	1	3.93	4	102,059	3.93	473	6,561,712	3.37	289	2,974,113
			2	7.43	11	274,120	7.43	166	1,827,603	6.3	95	1,438,351
	2	1	1	8.27	(0.1)	30,941,006	8.27	(1.57)	13,608,874	3.97	514	4,071,992
			2	11.54	46	1,036,294	11.54	194	2,570,270	6.42	10	127,100
		2	1	5.49	170	4,091,687	5.49	(0.46)	17,685,610	3.52	434	3,874,277
			2	9.7	37	870,979	9.7	216	2,405,941	5.9	30	355,178
	3	1	1	9.04	1885	26,624,840	9.04	(1.61)	16,855,483	3.27	21	246,465
			2	12.23	29	663,867	12.23	109	1,146,528	5.82	5	55,868
		2	1	8.87	(0.96)	36,503,808	8.87	(2.38)	10,696,593	3.04	635	4,206,533
			2	12.82	87	1,742,690	12.82	217	2,522,865	4.86	8	111,879
1	1	1	1	5.59	76	1,914,746	5.59	(0.11)	17,113,480	3.8	573	6,126,573
			2	9.34	28	643,479	9.34	291	3,729,740	6.52	67	867,485
		2	1	3.91	3	82,141	3.91	700	9,037,474	3.34	187	2,046,082
			2	7.45	9	240,922	7.45	130	1,467,783	6.32	81	1,016,680
	2	1	1	8.17	1954	31,230,960	8.17	(0.92)	15,831,461	3.87	287	2,426,309
			2	11.82	54	1,175,641	11.82	191	2,418,919	6.66	14	163,615
		2	1	5.32	140	3,276,211	5.32	1965	13,261,271	3.36	233	2,195,079
			2	9.54	27	630,580	9.54	105	1,584,731	5.71	25	301,468
	3	1	1	9.12	1951	28,910,438	9.12	(1.39)	18,760,285	3.34	31	369,543
			2	12.25	34	736,537	12.25	99	1,178,800	5.84	2	27,418
		2	1	8.96	(1.02)	39,450,904	8.96	(2.25)	10,807,958	3.04	663	5,630,375
			2	12.92	137	2,942,472	12.92	467	4,241,231	4.99	19	231,237

Table 4.3: Number of variables and constraint of the formulations

a/b	# of	SLS-Q	CSLS-II	CSLS-III	CSLS-IV
2	variables	$4T$	$5T$	$5T$	-
	linear constraints	$3T$	$3T$	$3T$	-
	quadratic constraints	-	T	T	-
3	variables	-	$6T$	$6T$	$7T$
	linear constraints	-	$3T$	$3T$	$3T$
	quadratic constraints	-	$2T$	$2T$	$3T$

- : not applicable

Table 4.4: Hyperbolic inequalities for cubic compression cost function

CSLS-II	CSLS-III	CSLS-IV
$w_t^2 \leq r_t k_t$	$w_t^2 \leq r_t k_t$	$w_t^2 \leq r_t y_t$
$k_t^2 \leq w_t$	$k_t^2 \leq w_t y_t$	$o_t^2 \leq k_t$
$w_t \geq 0$	$w_t \geq 0$	$k_t^2 \leq w_t o_t$
		$w_t, o_t \geq 0$

time is about 200 seconds) whereas CSLS-II stops with positive gap due to time limit for 10 out of 24 instances. When we examine the results of SLS-Q, an interesting result is obtained: it can solve an instance within 2 seconds, whereas for another one it stops with 1% optimality gap due to time limit. Moreover, SLS-Q solves 10 instances in less time than CSLS-III, but its solution time seems not so stable. It solves an instance which is solved by CSLS-III in about 300 seconds in only 4 seconds. On the other hand, another instance that is solved by CSLS-III in less than 40 seconds is solved by SLS-Q in about 2000 seconds. When we investigate the instances in detail, we observe that when setup cost increases and capacities become tighter, solution time of SLS-Q increases. These results may be related to root gaps and sizes of the formulations. Note that root gap of CSLS-II and SLS-Q are the same and root gap of CSLS-III is better for all of the instances. In Table 4.3, we report the number of variables and constraints of the formulations for quadratic and cubic compression functions. Note that, for the quadratic case, SLS-Q has the smallest number of constraints and variables and CSLS-II and CSLS-III have the same number of variables and constraints. What can be observed form these results is the following. Although number of variables and constraints are increased for conic quadratic reformulation in CSLS-II compared to SLS-Q, as gaps on the root nodes are the same for both of the formulations, SLS-Q performs better than CSLS-II. On the other hand, root gap of CSLS-III is improved at the expense of increasing model size. Therefore, for relatively easier instances, smaller formulation, as in SLS-Q, may perform better whereas for the harder ones the formulation with smaller root gaps, as in CSLS-III, may be better.

For the cubic compression cost function, we need to add all the conic inequalities. Hyperbolic inequalities, used in the conic reformulations, can be seen in

Table 4.4. Note that, the first inequalities used are the same for both of the formulations and the second inequality used in CSLS-III implies the one used in CSLS-II. For the cubic compression cost function, we also consider another strengthened formulation, in which rather than using inequalities $k_t^3 \le r_t y_t^2$ (given by (4.15) for $a = 3$, $b = 1$), we use inequalities $k_t^3 \le r_t y_t$, for $t = 1, \ldots, T$. This formulation and its conic reformulation will be referred as SLS-IV and CSLS-IV, respectively. Inequalities used for CSLS-IV are also given in Table 4.4. Note that more variables and hyperbolic inequalities are used for CSLS-IV, and the inequalities are different from the inequalities used in CSLS-II and CSLS-III.

Table 4.5: Effect of strengthening - cubic compression cost

Parameters				**CSLS-II**			**CSLS-III**			**CSLS-IV**		
α	f	κ	$\frac{C}{\theta v}$	rgap	cpu (gap)	node #	rgap	cpu (gap)	node #	rgap	cpu (gap)	node #
2	1	1	1	3.91	838	5,078,538	3.56	1341	6,749,843	3.69	33	121,618
			2	7.09	225	2,250,718	6.42	329	2,445,704	6.62	111	681,670
		2	1	3.86	765	4,589,460	3.66	(0.01)	9,577,594	3.73	31	125,007
			2	7.17	550	4,163,002	6.78	575	4,485,975	6.9	195	876,346
	2	1	1	4.13	1305	7,609,046	3.3	815	3,392,463	3.6	159	396,016
			2	7.6	285	2,424,222	6.01	157	1,400,172	6.49	50	222,179
		2	1	3.65	785	4,396,585	3.17	416	3,148,587	3.34	18	56,223
			2	6.67	84	463,807	5.72	78	486,854	6.01	28	137,715
	3	1	1	4.43	(0.73)	6,272,523	2.67	525	2,301,886	3.3	137	378,028
			2	8.15	126	968,464	4.61	52	354,709	5.66	29	93,946
		2	1	3.62	(0.19)	7,861,026	2.6	770	4,104,475	2.97	23	69,675
			2	6.43	67	571,358	4.37	26	178,464	4.99	28	138,135
1	1	1	1	4.01	1105	6,042,570	3.66	884	5,961,144	3.79	46	140,596
			2	7.49	650	4,722,629	6.83	484	4,625,674	7.03	192	897,207
		2	1	3.96	(0.26)	7,772,856	3.76	(0.55)	16,234,025	3.83	43	191,853
			2	7.21	573	4,411,070	6.82	523	3,540,305	6.94	173	878,045
	2	1	1	4.03	1425	6,803,350	3.21	331	1,813,792	3.5	43	109,090
			2	7.66	181	1,403,050	6.06	253	1,429,436	6.54	59	281,301
		2	1	3.52	476	2,942,293	3.04	282	1,702,281	3.21	36	102,870
			2	6.83	70	546,947	5.9	121	774,107	6.18	44	179,112
	3	1	1	4.92	2000	7,669,462	3.11	(0.14)	8,461,099	3.75	1231	2,744,415
			2	8.37	254	1,837,744	4.82	110	873,418	5.87	96	373,115
		2	1	3.86	(0.84)	6,647,400	2.82	918	4,570,382	3.19	183	569,582
			2	6.42	54	382,068	4.38	31	160,493	5	20	87,244

According to the results for cubic compression cost function, given in Table 4.5, conic strengthening again improves the root gap of CSLS-II. However, for this case improvement is not as good as for the quadratic case: for the quadratic compression cost function average root gap reduction is about 4% (40%, relatively), but for the cubic compression cost function it is about 1% (20%, relatively). Although root gap for CSLS-III is the best, the performance of CSLS-IV could be viewed as better since it solves all the instances within the time limit and its average solution time is about 120 seconds. The difference between CSLS-II and

CSLS-III is not clear for this case: 18 out of 24 instances are solved by both of the formulations, and 13 of them are solved in less time by CSLS-III. There is one instance that is solved by CSLS-II but not by CSLS-III, but three of the instances that cannot be solved by CSLS-II are solved by CSLS-III. Moreover, if we investigate the results in more detail, we can observe that CSLS-III mostly performs better than CSLS-II in harder instances (with large setup costs and tighter capacities). The number of variables and constraints for these formulations are also given in Table 4.3. Note that, the number of variables and constraints of CSLS-IV are larger than the ones for CSLS-II and CSLS-III, and the latter two formulations have equal number of variables and constraints. Although the size of CSLS-IV is larger, the root gap of this formulation is not the best. On the other hand, this formulation performs better in terms of solution times. This situation may be caused by the different types of conic inequalities added to this formulation (Table 4.3).

Overall, we observed that conic strengthening improves root gaps. This improvement is more definite for the quadratic compression cost function, since CSLS-III outperforms CSLS-II for this case. But for the cubic compression cost function, CSLS-IV, in which more conic inequalities are used, outperforms CSLS-III, for our instances. In summary, by utilizing second order cone programming, we could solve the relatively practical sizes of stochastic capacitated lot sizing problem with a nonlinear compression cost function in a reasonable computation time instead of relying on a heuristic approach.

4.3.2 Effect of controllable processing times

Controlling the capacity of the system can be a beneficial tool to hedge against demand uncertainty. For this purpose, in this section, we report the results of several experiments to show the benefits of controlling processing times under different uncertainty/cost/capacity settings. In order to achieve this, we will compare the optimal costs for the problem with and without controllable processing times, which will be called as SLS-C and SLS, respectively, and report

Table 4.6: Experimental design factors and their settings

Factor	**Explanation**	**# of levels**	**Factor settings**		
			1	2	3
α	Service level	2	0.95	0.99	
f	Setup cost	3	1750	3500	7000
κ	Compression cost coefficient	3	0.01	0.5	1
$C/(\theta v)$	Avg. capacity tightness	3	5	10	20
u	Max. possible compression (%)	3	10	30	50
β	$\theta_t \sim U\left[(1-\beta)\theta, (1+\beta)\theta\right]$	2	0.1	0.5	–
CV	Coeff. of variation (%)	3	10	30	50

the cost reduction. In this part, we again assume that all the parameters are time-invariant, and the compression cost function is quadratic or cubic. We consider instances with $T = 20$, $h = 1$, $c = 0$, $C = 300$ and $v = 1$. The rest of the parameters are generated according to the ratios given in Table 4.6. We consider different capacity and demand scenarios by considering different $C/\theta v$ and β values. For example, for $\beta = 0.5$, and $C/\theta p = 5$, mean demand of period t is generated as $\theta_t \sim U\left[30, 90\right]$ since $\theta = 60$ for this setting. Thus, when β is smaller, mean demand of each period becomes close to each other and when it increases it is possible to have fluctuating mean demand. We also consider different demand variability levels by considering different coefficient of variation settings. Note that according to Table 4.6, there are 972 different parameter settings for both of quadratic and cubic compression cost functions. Moreover, we generated 5 replications for each setting, thus we generated 4860 randomly generated problem instances for both of the functions. We summarize the results of this experiment in Tables 4.7 - 4.10. As all of the instances are solved to optimality in less than one second, we do not report solution times in this section. In order to see the effect of controllable processing times under different scenarios, we report the improvements for different combinations of parameters. In these tables, the value on the first row of each cell represents the average percentage cost reduction (Δ) for given parameter settings, and the maximum percentage cost reduction (Δ_{max}) obtained over all instances with this setting is given in the second row.

4.3.2.1 Effect of Setup Costs

We obtain an overall 6.54% average cost improvement for the quadratic compression cost function. Table 4.7 gives the percentage improvements for different service level α, setup cost f, coefficient of variation CV and capacity values. We first observe that Δ increases as set up cost increases. While the set up cost increases, compressing the processing times and reducing the number of production periods becomes more valuable. When we examine the difference between the number of production periods for SLS and SLS-C, we see that the average reduction in the number of production periods is about 0.45, 0.63 and 0.73 for $f = 1750, 3500, 7000$, respectively. For these setup cost values, average percentage cost reduction is 1.48 , 6.15 and 11.99, respectively, and Δ_{max} may be as high as 30% when setup cost is high.

Table 4.7: Service level vs. Setup Cost vs. Capacity vs. CV (Quadratic)

		Setup Cost											
		1750				3500				7000			
		CV				CV				CV			
$C/\theta v$	α	10	30	50	Avg.	10	30	50	Avg.	10	30	50	Avg.
5	0.99	2.41	1.6	2.15	2.05	6.44	4.07	4.86	5.12	10.89	7.34	8.18	8.8
		6.63	7.18	7.41		15.64	17.43	17.38		24.18	28.75	28.62	
	0.95	2.96	1.52	1.5	1.99	6.99	4.02	3.74	4.92	11.39	8.09	6.6	8.69
		6.95	6.61	6.99		15.95	15.62	16.91		24.43	24.16	28.29	
	Avg.	2.69	1.56	1.83	2.03	6.72	4.05	4.3	5.02	11.14	7.72	7.39	8.75
10	0.99	2.62	1.08	0.62	1.44	8.78	3.77	2.34	4.96	15.62	8.64	4.54	9.6
		5.69	5.87	4.79		13.47	13.43	12.29		20.62	20.5	19.53	
	0.95	3.22	1.73	0.89	1.95	9.83	6.69	3.23	6.58	17.2	14.08	7.26	12.85
		5.83	6.3	5.59		13.62	13.9	13.13		20.75	20.9	20.25	
	Avg.	2.92	1.41	0.76	1.7	9.31	5.23	2.79	5.78	16.41	11.36	5.9	11.22
20	0.99	0.42	1.02	0.26	0.57	4.5	11.36	3.95	6.6	8.41	23.47	12	14.63
		3.94	4.96	3.65		17.53	18.18	16.7		29.58	29.95	28.7	
	0.95	0.59	1.47	0.64	0.9	4.76	12.74	8.68	8.73	8.59	24.06	19.45	17.37
		4.14	3.84	4.63		17.77	17.65	17.79		29.78	29.77	29.62	
	Avg.	0.51	1.25	0.45	0.74	4.63	12.05	6.32	7.67	8.5	23.77	15.73	16
Avg.		2.04	1.4	1.01	1.48	6.88	7.11	4.47	6.15	12.02	14.28	9.67	11.99

When we investigate the results in detail, we observe that all of the improvements are not due to reduction in the number of production periods. In about 688 (out of 4860) instances, though the number of production periods are the same for SLS and SLS-C, cost reduction is obtained by compressing the processing times and reducing the total inventory holding cost. However, since setup cost is the dominant one among the other cost terms, the average improvement for these instances is about 0.16% (maximum is 1.77%).

Table 4.8: Setup Cost vs. κ (Quadratic)

f	κ			Avg.
	0.01	0.5	1	
1750	2.92	0.9	0.62	1.48
	7.41	5.81	5.78	
3500	9.73	4.96	3.77	6.15
	18.18	17.52	17.3	
7000	16.27	10.8	8.89	11.99
	29.95	29.65	29.54	
Avg.	9.64	5.55	4.43	6.54

4.3.2.2 Effect of Capacity and Demand Parameters

Parameters α, CV and $C/\theta v$ affect the difference between capacity and modified demand $\hat{d}$. Since we assume that the capacity C and the unit processing time p are constant, $C/\theta v$ increases only when the mean demand θ decreases. Thus, capacity compared to the modified demand increases with $C/\theta v$. When the service level α increases, Z_α and consequently, the modified demand parameter $\hat{d}$ increases. Similarly, when CV increases σ increases and again, $\hat{d}$ increases. Thus, for larger α or CV values it is possible to have tighter capacities relative to the demand. Note that, when capacities are large enough to satisfy the demand, which is possible when α and CV are smaller and $C/\theta v$ is larger, compressing the processing times may not be a preferred option. For example, controllable processing times have no advantage if the system is uncapacitated. On the other hand, when capacities are tight which is possible for larger α and CV values and smaller $C/\theta v$, even though the processing times are compressed, it may not be possible to obtain a better solution or the improvement may be small relative to the total cost. Note that, in this case, more compression should be done in order to reduce the number of production periods and as compression cost is convex, compression may not be beneficial anymore. Therefore, controllable processing times are more beneficial when capacities are medium sized relative to the modified demand.

Results given in Table 4.7 confirm the observations explained above. For example, for $C/\theta v = 5$ or 10, Δ is maximum when $CV = 10$ and if $C/\theta v$ is increased to 20, Δ is larger for $CV = 30$.

α and the coefficient of variation have the same effect on the modified demand, but according to Table 4.7, Δ is more affected by the changes in the coefficient of variation. Note that, the changes in CV affect the modified demand in larger amounts and this is the reason of larger changes of Δ with respect to CV.

Table 4.9: Capacity vs. mean demand variability vs. max. possible compression (Quadratic)

$\frac{C}{\theta v}$	β							
	10				50			
	u				u			
	10	30	50	Avg.	10	30	50	Avg.
5	3.66 12.98	6.08 24.43	7.18 28.62	5.64	3.08 13.1	5.15 23.75	6.43 28.75	4.89
10	5.47 20.75	8.07 20.75	8.07 20.75	7.2	3.35 20.9	6.16 20.9	6.26 20.9	5.26
20	6.39 29.77	8.7 29.77	8.7 29.77	7.93	5.91 29.95	9.54 29.95	9.54 29.95	8.33
Avg.	5.17	7.62	7.98	6.92	4.11	6.95	7.41	6.16

When we investigate the results in more detail, we observe that as the capacity increases, the total cost of SLS decreases, in general. Therefore, even though the cost reduction due to controllable processing times is the same for different capacity settings, as Δ indicates the percentage cost improvement, Δ may be higher for larger capacity settings. An example of this situation is observed for $CV = 10$ and $C/\theta v = 5$ or 10.

To sum up, according to Table 4.7, we can conclude that controllable processing times are more beneficial when setup costs are high and the difference between the capacities and the modified demand is medium sized.

4.3.2.3 Other Parameters

Table 4.8 shows the percentage improvements for setup cost and compression cost coefficient κ. When κ increases, as it is expected, the compression cost increases and consequently, the cost reduction that can be obtained by compressing the processing times decreases. But note that when setup cost increases, it is more beneficial to compress the processing times, even with larger κ, and it is possible to have a reduction of 29%.

In Table 4.9, the percentage improvements for different capacity levels, maximum possible compression amounts (u) and mean demand scenarios are given. As expected, if u increases, the cost difference between SLS and SLS-C, and consequently, Δ may increase. If capacities are large enough, larger u values may not change the optimal production plan. Note that, Δ and Δ_{max} are the same for different u values when capacities are large enough. Moreover, average cost and number of production periods differences are the same for these settings. On the other hand, when capacity is tighter, increasing the maximum possible compression amount increases the improvement.

According to Table 4.9, effect of β on Δ is relatively small. More improvement is observed when capacities are looser and mean demand fluctuates more or capacities are tighter and mean demand is stable. When we examine the results in detail, we obtain that when capacity is large enough and mean demand is stable, without reducing processing times an optimal production plan can be observed. Moreover, we note that in none of the test instances with $C/\theta v = 20$, $CV = 10$ and $\beta = 0.1$, a cost reduction is observed. But this setting is an extreme case as capacity is large enough and with only one production period all the necessary demand is produced.

4.3.2.4 Cubic Compression Cost Function

In Table 4.10, results for the cubic compression cost function are given. In this table, we report the percentage improvements for different service level α, setup cost f, coefficient of variation CV and capacity values. The average improvement is 3.03%. When the compression cost function is changed from quadratic to cubic, Δ decreases since compression becomes more expensive (when $k_t \geq 1$). Although the average improvement decreases, behavior of Δ is very similar to one of the quadratic case. Note that, for some settings Δ_{max} may be as high as the one for quadratic compression cost function, but also for some parameter combinations it may be zero. As similar results are obtained for this case, we do not report the impact of other problem parameters in detail.

Table 4.10: Service level vs. Setup Cost vs. Capacity vs. CV (Cubic)

		Setup Cost											
		1750				3500				7000			
		CV				CV				CV			
$C/\theta v$	α	10	30	50	Avg.	10	30	50	Avg.	10	30	50	Avg.
5	0.99	0.85	0.56	1.02	0.81	2.53	1.45	2.33	2.1	4.9	3.16	3.79	3.95
		4.85	3.92	3.98		8.94	8.12	7.32		12.82	12.35	10.55	
	0.95	1.59	0.31	0.37	0.76	3.66	1.27	0.77	1.9	6.46	2.76	1.7	3.64
		5.08	4.7	3.65		9.14	8.82	7.06		12.97	12.74	11.62	
	Avg.	1.22	0.44	0.7	0.79	3.1	1.36	1.55	2	5.68	2.96	2.75	3.8
10	0.99	1.26	0.16	0.01	0.48	4.31	0.88	0.03	1.74	9.5	2.63	0.28	4.14
		5.68	4.77	0.1		13.46	12.63	0.99		20.61	19.98	12.13	
	0.95	1.64	0.58	0.1	0.77	6.72	3.16	0.44	3.44	12.69	6.13	1.98	6.93
		5.83	6.07	2.93		13.62	13.74	11.19		20.75	20.79	18.99	
	Avg.	1.45	0.37	0.06	0.63	5.52	2.02	0.24	2.59	11.1	4.38	1.13	5.54
20	0.99	0.17	0.34	0	0.17	2.27	4.33	0.62	2.41	5.67	10.24	3.32	6.41
		3.64	4.73	0		17.32	18.02	10.95		29.45	29.85	25.02	
	0.95	0.22	0.78	0.16	0.39	3.47	7.28	2.92	4.56	6.54	16.37	6.77	9.89
		4.06	3.81	3.67		17.71	17.63	17.11		29.75	29.75	29.19	
	Avg.	0.2	0.56	0.08	0.28	2.87	5.81	1.77	3.48	6.11	13.31	5.05	8.16
	Avg.	0.96	0.46	0.28	0.56	3.83	3.06	1.19	2.69	7.63	6.88	2.97	5.83

As a result of our computational experiments, we can say that taking processing times as constant increases system costs and even with small compressions it is possible to have less costly production plans. Although the controllable processing times provide cost improvements in most of the randomly generated problem instances, we observe largest improvements when setup costs are high, compression costs are low and capacities are medium sized compared to the modified demands.

4.4 Conclusions

In this chapter, we consider the stochastic capacitated lot sizing problem with controllable processing times where compression cost function is a convex function of the compression amount. To the best of our knowledge, this problem is not studied before. We formulate this problem using SOCP, strengthen the formulation and solve the proposed formulations using a commercial solver. In our computational experiments, we observe that conic strengthening reduces the root gaps, and consequently, the computation times.

Although controllable processing times may be applicable to many real life production and inventory systems, processing times are assumed as constant in the classical lot sizing problems. In the second part of our computational experiments, we show that this assumption causes higher system costs. We observe that controllable processing times are more valuable when the system has medium sized capacities and larger setup costs.

In the next chapter, we consider the stochastic lot sizing problem with controllable processing times under a different setting where demand follows a discrete distribution and the problem can be formulated as a multistage stochastic program.

Chapter 5

Multistage Stochastic Lot Sizing Problem with Nervousness Considerations

In this chapter, we again consider the stochastic lot sizing problem with controllable processing times. In the previous chapter, we studied this problem under static uncertainty strategy. In the static uncertainty strategy all the decisions are taken at the beginning of the planning horizon. Since the production plan is known with certainty, it is easier to make the necessary arrangements (material planning, supplier contracts, etc.) under this strategy. On the other hand, it is possible to reduce the total cost by dynamically updating the production plan according to demand realizations. However, dynamic strategies have a negative impact on the system which is called "nervousness". System nervousness is caused by uncertainty of the production plans. Since the production plan may change over time, it becomes harder to talk about the future production decisions, and this situation may affect all the supply chain.

In this chapter, we study the stochastic lot sizing problem with controllable processing times and nervousness considerations under a dynamic strategy. We formulate the problem such that nervousness of the system is kept under control by some additional parameters and restrictions. In Section 5.1 we formally define the problem and formulate it. As we did in our previous chapter, we will reformulate the problem as SOCP in Section 4.2. In Section 4.3 we develop several classes of valid inequalities for our formulation and we test these inequalities in Section 4.4. Finally in Section 4.5 we present some concluding remarks.

5.1 Problem Definition and Formulation

We assume that the demand follows a discrete distribution and it can be represented by a scenario tree. Suppose that possible demand realizations are represented by a scenario tree $\mathcal{T} = (V, E)$ with T (number of periods) stages and n nodes. Let $t(i)$ be the time period of node $i \in V$ and π_i be the probability associated with the state represented by node i. Set of nodes defined for period t is denoted by V_t. Unique predecessor of node i is given by i^- and demand of node i is denoted by d_i. Assume that 0 is the root node. Let $P(i, j)$ be the path from node i to node j and $V(u)$ be the set of descendants of node u, including u. These definitions are illustrated in Figure 5.1.

Let f_t be the setup cost for period t, and c_i and h_i be the unit production and inventory holding costs for node i, respectively. We assume that all the cost parameters include the probabilities.

We assume that capacity of period t is C_t time units. Processing time without any compression is p time units, and processing time of any item can be reduced by at most u $(< p)$ time units. Without loss of generality, we assume that $p = 1$. If processing times in node i is decreased by r_i time units in total, then compression cost is given by $\gamma_i(r_i) = \kappa_{3i} r_i^{a_3/b_3}$ where $\kappa_{3i} \geq 0$ and $a_3 \geq b_3 > 0$. Note that γ is a convex function of the compression amount r and this function can represent increasing marginal cost of decreasing the processing times in larger amounts.

In the classical scenario tree formulation of the lot sizing problem, even though optimal production decisions for each node are known, we may not know the exact

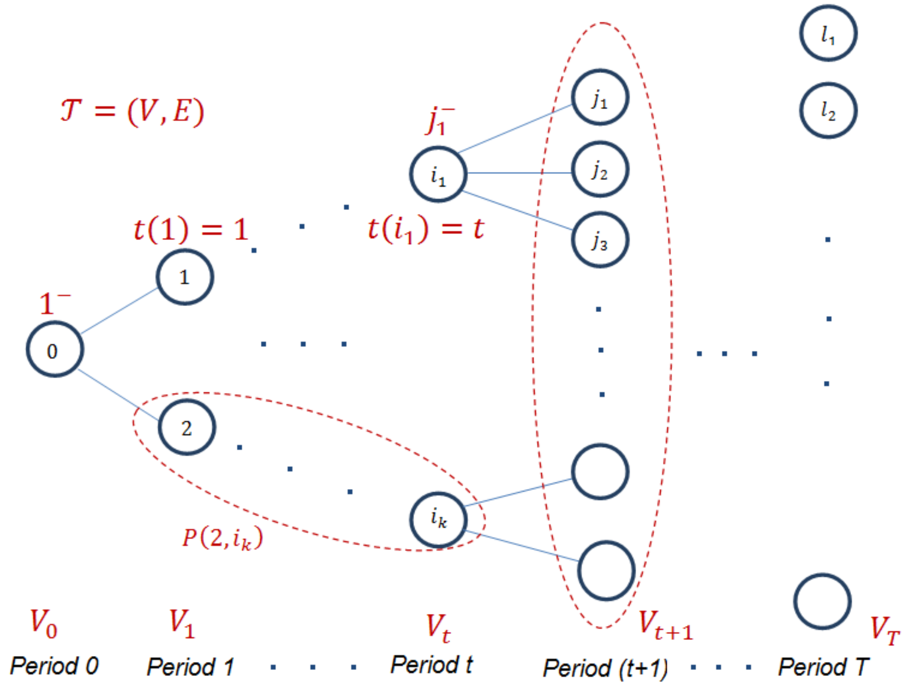

Figure 5.1: Scenario tree representation

production amount in a period until the demand of the period is realized. This situation causes both setup and quantity oriented nervousness in the system. In this study, we want to find a minimum cost solution to the problem which results less nervousness in the system. In other words, we have two different objectives: we still want to find a minimum cost production plan for the system and we want to decrease the system nervousness by considering additional constraints on the production decisions.

In order to reduce the setup oriented nervousness, we consider setup decisions for periods rather than deciding for each node separately. Thus, we determine the production periods at the beginning of the planning horizon, and in this way we exclude the setup oriented nervousness from our formulation. Moreover, we control quantity oriented nervousness by restricting production amounts under different scenarios and penalizing different production decisions for the same time period. Specifically, we assume that there is a monetary expression for quantity oriented nervousness which may be determined according to flexibility of the system. We define two types of decisions related to production amounts (*exact*

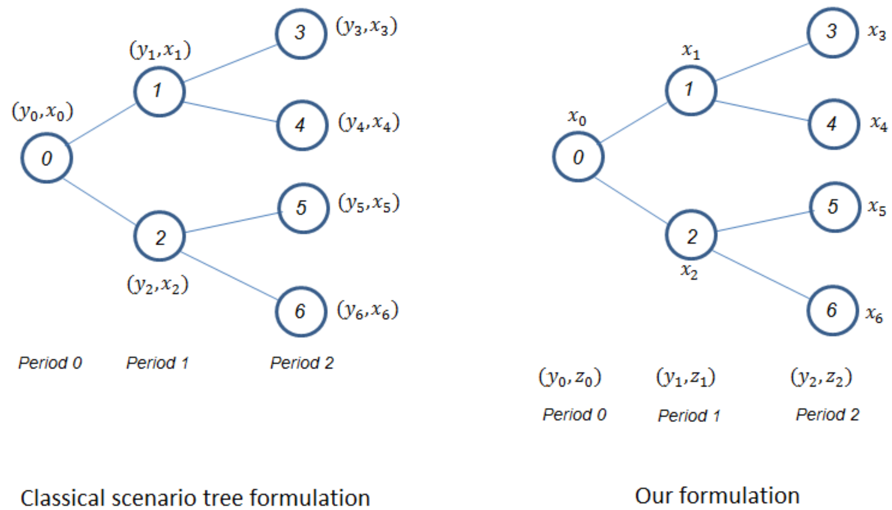

Figure 5.2: Scenario tree decisions

production amounts for each node and *promised* production amounts for each period) and assume that there is a nervousness cost depending on the relation of these two decisions (see Figure 5.2).

Let y_t be the setup variable which is equal to 1 if there exists a production in period t and 0 otherwise. Let x_i be the production amount in node i and $z_{t(i)}$ be the promised production amount in period $t(i)$. We relate the production decisions for each node and the promised production amount for the period of that node by the range parameters $\lambda_{t(i)} \leq 1$ and $\delta_{t(i)} \geq 1$: $\lambda_{t(i)} z_{t(i)} \leq x_i \leq \delta_{t(i)} z_{t(i)}$ for $i \in V$. In other words, production amount of any node defined for period t should be in the interval $[\lambda_t z_t, \delta_t z_t]$.

Note that, promised production amounts are decided for each period. Therefore, when the problem is solved and optimal promised production amounts are obtained, these values can be used in arrangements before the production, purchasing or transportation. For example, a producer can make arrangements for the production like material planning by using these values. Moreover, promised production amounts can be seen as reserved capacities in a buying or transportation system. A buyer can inform its supplier or transporter and reserve capacities for its future orders according to these values (see e.g. [102, 103]).

Accuracy of promised production amounts (z) is controlled by the parameters λ_t and δ_t. When these parameters get close to 1, the interval $[\lambda_t z_t, \delta_t z_t]$ shrinks and $z_{t(i)}$ becomes closer to the exact production/order amount x_i. Moreover, in this case, production amounts for different nodes also get closer to each other and quantity oriented nervousness decreases since $\lambda_t z_t \leq x_i \leq \delta_t z_t$ should hold for all $i \in V_t$.

Suppose that $g_i\left(x_i, z_{t(i)}\right)$ denotes the nervousness (or penalty) cost for node i given that x_i units are produced in node i and $z_{t(i)}$ units are promised to be produced in period $t(i)$. We assume the system is flexible enough for letting the production amount in the interval $[\lambda'_t z_t, \delta'_t z_t]$ where $\lambda_t \leq \lambda'_t \leq 1$ and $1 \leq \delta'_t \leq \delta_t$. Thus, if $x_i \in [\lambda'_t z_t, \delta'_t z_t]$ then no nervousness cost is incurred. But if the production amount is not in this range, then a nervousness cost which is a convex function of the minimum distance between the production amount and this interval is incurred. Note that, we can keep the production amounts closer to this interval as much as possible by a convex nervousness cost function. To this end, we define x_{1i} and x_{2i} as $x_{1i} = \left[\lambda'_{t(i)} z_{t(i)} - x_i\right]^+$ and $x_{2i} = \left[x_i - \delta'_{t(i)} z_{t(i)}\right]^+$ where $[a]^+ = \max\{a, 0\}$ and assume that the nervousness cost is given by the function

$$g_i\left(x_i, z_{t(i)}\right) = \kappa_{1i} x_{1i}^{a_2/b_2} + \kappa_{2i} x_{2i}^{a_3/b_3}$$

where $\kappa_{2i}, \kappa_{3i} \geq 0$ are nervousness cost coefficients and $a_2 \geq b_2 > 0$, $a_3 \geq b_3 > 0$. The cost function g_i is illustrated in Figure 5.3.

We assume that λ', δ', and the cost parameters are determined according to flexibility of the system. For example, if the system is very sensitive to changes in the production amounts, then it is appropriate to set $\lambda' = \delta' = 1$ and $\kappa_{1i}, \kappa_{2i}, a_1/b_1, a_2/b_2$ to larger values. On the other hand, if it is easy to adapt the system for changes, then one can set $\lambda' = \lambda$, $\delta' = \delta$ and get rid of the nervousness cost. We will explore special cases of this cost function in Section 4.5.

To summarize, we control nervousness in three different ways. First, as set up decisions are taken for each period, we reduce setup oriented nervousness (note that, if $\lambda_{t(i)} = 0$, it is still possible to have $y_{t(i)} = 1$ but $x_i = 0$). Secondly, by constraints (5.5) and parameters λ and δ, quantity oriented nervousness can be

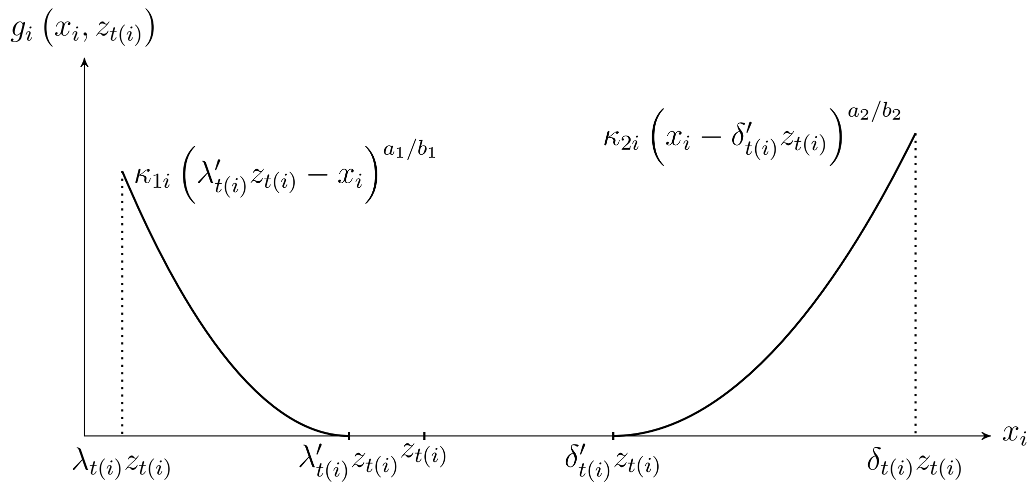

Figure 5.3: Cost of nervousness

reduced. Finally, quantity oriented nervousness is penalized by the cost function $g_i\left(x_i, z_{t(i)}\right)$.

In addition to the variables defined above, let s_i be the inventory on hand at the end of period $t(i)$ for node i. Note that, we have three types of recourse actions to hedge against demand uncertainty: production, inventory, and compression amounts are defined for each node separately.

The problem can be formulated as the following:

MLSI

$$\min \sum_{t=1}^{T} f_t y_t + \sum_{i\in V}(c_i x_i + h_i s_i) + \sum_{i\in V}\left(\kappa_{1i} x_{1i}^{a_1/b_1} + \kappa_{2i} x_{2i}^{a_2/b_2}\right) + \sum_{i\in V}\kappa_{3i} r_i^{a_3/b_3} \tag{5.1}$$

$$s.t.\ s_{i^-} + x_i = d_i + s_i \qquad i \in V, \tag{5.2}$$

$$x_i \le C_{t(i)} y_{t(i)} + r_i \qquad i \in V, \tag{5.3}$$

$$r_i \le u x_i \qquad i \in V, \tag{5.4}$$

$$\lambda_{t(i)} z_{t(i)} \le x_i \le \delta_{t(i)} z_{t(i)} \qquad i \in V \tag{5.5}$$

$$x_{1i} \ge \lambda'_{t(i)} z_{t(i)} - x_i \qquad i \in V, \tag{5.6}$$

$$x_{2i} \ge x_i - \delta'_{t(i)} z_{t(i)} \qquad i \in V, \tag{5.7}$$

$$s_{0^-} = 0, \tag{5.8}$$

$$x_i, x_{1i}, x_{2i}, s_i, r_i \ge 0 \qquad i \in V, \tag{5.9}$$

$$y_t \in \{0,1\},\ z_t \ge 0 \qquad t = 1, \ldots, T. \tag{5.10}$$

In the objective function (5.1) we minimize the total (expected) setup, production, inventory holding, nervousness and compression costs. Constraints (5.2) are the classical demand satisfaction (inventory balance) constraints for each node i: total amount due to production in node i, and inventory left from the unique parent of node i should be equal to the total demand in node i, and inventory at the end of node i. We assume that the initial inventory s_{0^-} is zero by constraints (5.8). Due to constraints (5.3), total time necessary for producing x_i units (x_i time units since we assume $p = 1$) minus the total compression amount in node i (r_i time units) should be less than or equal to the capacity of node i. Constraints (5.4) ensure that processing time of each job is reduced by at most u time units. Constraints (5.5) relate the variables x_i and $z_{t(i)}$ and also the production amounts of different nodes defined for the same time period $t(i)$. Note that, according to these constraints, if nodes i and i' are both defined for period t, then production amounts of these nodes should be in the same range: $x_i, x_{i'} \in [\lambda_t z_t, \delta_t z_t]$. Definitions of the variables x_{1i} and x_{2i} are expressed in the constraints (5.6) and (5.7) (and nonnegativity constraints), respectively. Constraints (5.9) and (5.10) define the ranges and types of the variables.

Note that due to constraints (5.5), for any period t, $\lambda_t z_t \leq \min_{i\in V_t}\{x_i\}$, and $\delta_t z_t \geq \max_{i\in V_t}\{x_i\}$, and consequently,

$$\frac{1}{\delta_t}\max_{i\in V_t}\{x_i\} \quad \leq z_t \quad \leq \frac{1}{\lambda_t}\min_{i\in V_t}\{x_i\}. \tag{5.11}$$

Thus, feasible z and x should satisfy inequality (5.11).

In the next section, we will reformulate the problem as SOCMIP and strengthen the formulation.

5.2 Reformulation of the Problem

As we did in the previous chapter, we can reformulate $MLSI$ as a SOCMIP. To do this, we first introduce nonnegative auxiliary variables w_{1i}, w_{2i} and v_i such that

$$\begin{array}{rcll} x_{1i}^{a_1/b_1} & \leq & w_{1i} & i \in V \\ x_{2i}^{a_2/b_2} & \leq & w_{2i} & i \in V \\ r_i^{a_3/b_3} & \leq & v_i & i \in V. \end{array}$$

We replace nonlinear terms in the objective function (5.1) by these auxiliary variables. Moreover, as $b_1, b_2, b_3 > 0$ and $y_{t(i)} = 1$ implies $x_i = x_{1i} = x_{2i} = r_i = 0$, we can strengthen these inequalities as

$$\begin{array}{rcll} x_{1i}^{a_1} & \leq & w_{1i}^{b_1}\, y_{t(i)}^{a_1-b_1} & i \in V \\ x_{2i}^{a_2} & \leq & w_{2i}^{b_2}\, y_{t(i)}^{a_2-b_2} & i \in V \\ r_i^{a_3} & \leq & v_i^{b_3}\, y_{t(i)}^{a_3-b_3} & i \in V. \end{array}$$

Thus, $MLSI$ can be reformulated as

MLSII

$$\min \sum_{t=1}^{T} f_t y_t + \sum_{i\in V}(c_i x_i + h_i s_i) + \sum_{i\in V}(\kappa_{1i} w_{1i} + \kappa_{2i} w_{2i}) + \sum_{i\in V} \kappa_{3i} v_i \tag{5.12}$$

$$\begin{aligned}
s.t.\ & s_{i^-} + x_i = d_i + s_i \qquad i \in V,\\
& x_i \le C_{t(i)} y_{t(i)} + r_i \qquad i \in V,\\
& r_i \le u x_i \qquad i \in V,\\
& \lambda_{t(i)} z_{t(i)} \le x_i \le \delta_{t(i)} z_{t(i)} \qquad i \in V\\
& x_{1i} \ge \lambda'_{t(i)} z_{t(i)} - x_i \qquad i \in V,\\
& x_{2i} \ge x_i - \delta'_{t(i)} z_{t(i)} \qquad i \in V,\\
& x_{1i}^{a_1} \le w_{1i}^{b_1}\, y_{t(i)}^{a_1-b_1} \qquad i \in V, && (5.13)\\
& x_{2i}^{a_2} \le w_{2i}^{b_2}\, y_{t(i)}^{a_2-b_2} \qquad i \in V, && (5.14)\\
& r_{i}^{a_3} \le v_{i}^{b_3}\, y_{t(i)}^{a_3-b_3} \qquad i \in V, && (5.15)\\
& x_i, x_{1i}, x_{2i}, s_i, r_i, w_{1i}, w_{2i}, v_i \ge 0 \qquad i \in V,\\
& y_t \in \{0,1\},\ z_t \ge 0 \qquad t = 1, \ldots, T.
\end{aligned}$$

Note that, inequalities (5.13), (5.14) and (5.15) are in the same form with inequalities (4.15). Thus, these inequalities are also conic quadratic representable. We will refer to the conic quadratic representation of *MLSII* as *CMLSII*.

5.3 Valid Inequalities

As it is stated in Section 2.2, there exist dynamic programming algorithms for special cases of the stochastic lot sizing problem (e.g. [59, 104]). But note that our problem is different from the problems considered in these studies since

- we consider setup decisions for periods; not for each node separately,
- production amounts of nodes defined for the same time period should be in a range,

- our cost function includes convex terms.

Most of the dynamic programming algorithms developed for different variations of the lot sizing problem use modifications of the "regeneration interval" property. In previous chapter, we show that even for a two period problem instance, this property does not hold for the problem we studied, due to the convex terms in the total cost function. Moreover, in the existing dynamic programming algorithms in the literature, value functions are defined for each node separately. Thus they cannot be used for solving our problem since decisions for different scenarios are related to each other in our problem (see the first two items above).

Valid inequalities developed for the uncapacitated stochastic lot sizing problem are also valid for our problem (e.g. [53]). But, it is possible to derive new valid inequalities for our problem due to the new variables and constraints introduced. In this section we will derive some new valid inequalities for the formulation $MLSII$.

For $u \in V$ and $k \in V(u)$, if balance constraints (5.2) are summed for $i \in P(u,k)$, we obtain the following inequalities:

$$s_{u^-} + \sum_{i \in P(u,k)} x_i \;=\; d_{uk} + s_k \qquad k \in V(u) \tag{5.16}$$

where $d_{uk} = \sum_{i \in P(u,k)} d_i$. Since $s \geq 0$,

$$s_{u^-} + \sum_{i \in P(u,k)} x_i \;\geq\; d_{uk} \qquad k \in V(u). \tag{5.17}$$

For the classical capacitated stochastic lot sizing problem, capacity is an upper bound for the production amount: $x_i \leq C_{t(i)}$. But when the processing times are controllable, we can produce more than the capacity by reducing the processing times. So for our problem, upper bound for the production amount is $C_{t(i)}y_{t(i)}+r_i$. Due to these upper bounds, inequalities (5.17) imply

$$s_{u^-} + \sum_{i \in P(u,k)} r_i + \sum_{i \in P(u,k)} C_{t(i)}y_{t(i)} \;\geq\; d_{uk} \qquad k \in V(u). \tag{5.18}$$

Note that, binary variables y relate inequalities (5.18) for different nodes: for $k_1, k_2 \in V(u)$, $C_t y_t$ appears in both of the inequalities for $t \leq \min\{t(k_1), t(k_2)\}$.

In this section, we assume that capacities are the same for all periods, $C_t = \bar{C}$ for $t = 1, \ldots, T$ and demand is expressed in terms of capacities, i.e. $\bar{d}_i = \frac{d_i}{\bar{C}}$. So, inequalities (5.18) are rewritten as

$$s_{u^-} + \sum_{i \in P(u,k)} r_i + \sum_{i \in P(u,k)} y_{t(i)} \geq \bar{d}_{uk} \qquad k \in V(u). \tag{5.19}$$

Note that, in these inequalities the variables are also expressed in terms of capacity (for example $\bar{r}_i = \frac{r_i}{\bar{C}}$), but for ease of notation we do not rename them.

We will drive two different sets of valid inequalities for the problem. First, we will show that for given u, inequalities (5.19) define a continuous mixing set structure. Van Vyve [90] introduced valid inequalities and an extended formulation for the continuous mixing set and showed that these inequalities are sufficient for describing the convex hull of the set. We will make use of his study to derive valid inequalities for our problem. Next, we will apply the mixing scheme of Günlük and Pochet [91] to inequalities (5.19) and obtain valid inequalities.

5.3.1 Continuous Mixing Set Structure

For given u, inequalities (5.19) define a continuous mixing set. To see this, let $\alpha_k = \sum_{i \in P(u,k)} y_{t(i)} - \lfloor \bar{d}_{uk} \rfloor$, $\sigma_k = \sum_{i \in P(u,k)} r_i$, $\hat{d}_k = \bar{d}_{uk} - \lfloor \bar{d}_{uk} \rfloor$ for $k \in V(u)$, and $s = s_{u^-}$. Then, inequalities (5.19) are equivalent to

$$\begin{aligned} s + \sigma_k + \alpha_k &\geq \hat{d}_k \quad k \in V(u) \\ \sigma_k \in \mathbb{R}_+,\ \alpha_k &\in \mathbb{Z} \quad k \in V(u) \\ s &\in \mathbb{R}_+ \end{aligned}$$

Valid inequalities of Van Vyve [90] for the continuous mixing set are based on costs of cycles in a graph constructed as the following:

Let $V' = \{j_0, j_1, \ldots, j_{|V(u)|}\}$ be an ordered set including all the descendants of node u such that $\hat{d}_{j_i} \leq \hat{d}_{j_{i+1}}$ for $i = 1, \ldots, |V(u)| - 1$. In this set, j_0 is a dummy node with $\hat{d}_{j_0} = 0$. Let $G'_u = (V', A')$ be a graph defined on the node set V' with the arc set $A' = \{(j_0, j_i) : 1 \leq i \leq |V(u)|\} \cup \{(j_i, j_0) : 1 \leq i \leq |V(u)|\} \cup \{(j_i, j_l) :$

$\hat{d}_{j_i} \neq \hat{d}_{j_l}$, $1 \leq i, l \leq |V(u)|\}$. Suppose that cost of an arc $(j_i, j_l) \in A'$ is given by:

$$\Phi_{j_i,j_l}(s,\sigma,\alpha) = \begin{cases} s + \sigma_{j_i} + \left(\hat{d}_{j_i} - \hat{d}_{j_l} + 1\right)\alpha_{j_i} - \hat{d}_{j_l} & \text{if } 1 \leq i < l \leq |V(u)|, \\ \sigma_{j_i} + \left(\hat{d}_{j_i} - \hat{d}_{j_l}\right)\alpha_{j_i} & \text{if } 1 \leq l < i \leq |V(u)|, \\ s + \sigma_{j_i} + \alpha_{j_i} - \hat{d}_{j_i} & \text{if } i = l \text{ and } 1 \leq i \leq |V(u)|, \\ s - \hat{d}_{j_l} & \text{if } i = 0 \text{ and } 1 \leq l \leq |V(u)|, \\ \sigma_{j_i} + \hat{d}_{j_i}\alpha_{j_i} & \text{if } l = 0 \text{ and } 1 \leq i \leq |V(u)|. \end{cases}$$

Graph G'_u should not include a negative cycle. Thus, each cycle in this graph leads to a valid inequality for the set defined by inequalities (5.19). For a given elementary cycle $\mathcal{C} \subset A'$, the associated cycle inequality is

$$\sum_{(j_i,j_l)\in\mathcal{C}} \Phi_{j_i,j_l}(s,\sigma,\alpha) \;\; \geq \;\; 0. \tag{5.20}$$

Separation of these valid inequalities is equivalent to finding a negative cost cycle in the graph G'_u and this can be done in $O(|V(u)|^3)$ time by the shortest path algorithm of Floyd-Warshall [23]. Moreover, Van Vyve [90] developed an extended formulation for the continuous mixing set based on the dual of this separation problem. Note that the graph G'_u does not contain any negative cycle if there exists $\theta \in \mathbb{R}^{|V(u)|+1}$ such that

$$\Phi_{j_i,j_l}(s,\sigma,\alpha) \;\; \geq \; \theta_{j_i} - \theta_{j_l} \qquad \text{for all } \; (j_i, j_l) \in A' \tag{5.21}$$

Inequalities (5.21) are equivalent to:

$$s + \sigma_{j_i} + \left(\hat{d}_{j_i} - \hat{d}_{j_l} + 1\right)\alpha_{j_i} \geq \; \hat{d}_{j_l} + \theta_{j_i} - \theta_{j_l} \quad (j_i, j_l) \in A' : \; 1 \leq i < l \leq |V(u)| \tag{5.22}$$

$$\sigma_{j_i} + \left(\hat{d}_{j_i} - \hat{d}_{j_l}\right)\alpha_{j_i} \geq \; \theta_{j_i} - \theta_{j_l} \qquad (j_i, j_l) \in A' : 1 \leq l < i \leq |V(u)| \tag{5.23}$$

$$s + \sigma_{j_i} + \alpha_{j_i} \geq \; \hat{d}_{j_i} \qquad 1 \leq i \leq |V(u)| \tag{5.24}$$

$$s \geq \; \hat{d}_{j_l} + \theta_{j_0} - \theta_{j_l} \qquad 1 \leq l \leq |V(u)| \tag{5.25}$$

$$\sigma_{j_i} + \hat{d}_{j_i}\alpha_{j_i} \geq \; \theta_{j_i} - \theta_{j_0} \qquad 1 \leq i \leq |V(u)| \tag{5.26}$$

Note that the extended formulation given by inequalities (5.22) - (5.26) and the valid inequalities (5.20) are derived for a fixed node u and for all the descendants of node u.

In our computational experiments, we consider the extended formulation for each $u \in V$ and add the extended formulation to our formulation. Results for this experiment are given in the computational experiments section.

5.3.2 Mixing Set Structure

In this subsection, we will apply the mixing procedure of Günlük and Pochet [91] to inequalities (5.19):

$$s_{u^-} + \sum_{i \in P(u,k)} r_i + \sum_{i \in P(u,k)} y_{t(i)} \;\; \geq \;\; \bar{d}_{uk} \qquad\qquad k \in V(u).$$

Suppose that u is fixed and let $\hat{d}_k = \bar{d}_{uk} - \lfloor \bar{d}_{uk} \rfloor$ be the fractional part of the total demand from node u to node k. Let $R = \{i_1, \ldots, i_K\} \subseteq V(u)$ be an ordered set such that $0 = \hat{d}_{i_0} \leq \hat{d}_{i_1} \leq \hat{d}_{i_2} \leq \ldots \leq \hat{d}_{i_K}$. Set of nodes in the paths from node u to the nodes in R are given by $V_R = \cup_{k \in R} P(u,k)$. Note that

$$s = s_{u^-} + \sum_{i \in V_R} r_i \;\; \geq \;\; s_{u^-} + \sum_{i \in P(u,k)} r_i \qquad \forall k \in R.$$

Thus, s is an upper bound for the continuous part of inequalities (5.19) for the nodes in R and inequalities (5.19) imply

$$s + \sum_{i \in P(u,k)} y_{t(i)} \;\; \geq \;\; \bar{d}_{uk} \qquad\qquad k \in R. \tag{5.27}$$

If we define $y_{uk} = \sum_{i \in P(u,k)} y_{t(i)}$ for $k \in R$, then inequalities (5.27) can be rewritten as

$$s + y_{uk} \;\; \geq \;\; \bar{d}_{uk} \qquad\qquad k \in R. \tag{5.28}$$

Inequalities (5.28) define a mixing set. We will apply the mixing procedure to these inequalities to obtain valid inequalities for our problem. Mixing of inequalities (5.28) leads to the following inequalities

$$s \geq \sum_{j=1}^{K} \left(\hat{d}_{i_j} - \hat{d}_{i_{j-1}}\right) \left(\lfloor \bar{d}_{u,i_j} \rfloor + 1 - y_{u,i_j}\right) \tag{5.29}$$

$$s \geq \sum_{j=1}^{K} \left(\hat{d}_{i_j} - \hat{d}_{i_{j-1}}\right) \left(\lfloor \bar{d}_{u,i_j} \rfloor + 1 - y_{u,i_j}\right) + \left(1 - \hat{d}_{i_K}\right) \left(\lfloor \bar{d}_{u,i_1} \rfloor - \bar{y}_{u,i_1}\right) \tag{5.30}$$

Let $\bar{t}_R$ be the maximum time period for the nodes in the set R, i.e., $\bar{t}_R = \max\{t(i) : i \in R\}$. For $t = t(u), \ldots, \bar{t}_R$, define

$$\Delta_t(R) = \sum_{i_j \in R : t(i_j) \geq t} \left(\hat{d}_{i_j} - \hat{d}_{i_{j-1}}\right).$$

Then inequalities (5.29) and (5.30) are equivalent to

$$s_{u^-} + \sum_{i \in V_R} r_i + \sum_{t=t(u)}^{\bar{t}_R} \Delta_t(R)\, y_t \geq \sum_{j=1}^{K} \left(\hat{d}_{i_j} - \hat{d}_{i_{j-1}}\right) \left(\lfloor \bar{d}_{u,i_j} \rfloor + 1\right) \tag{5.31}$$

$$\begin{aligned} s_{u^-} + \sum_{i \in V_R} r_i + \sum_{t=t(u)}^{t(i_1)} \left(\Delta_t(R) + 1 - \hat{d}_{i_K}\right) y_t + \sum_{t=t(i_1)+1}^{\bar{t}_R} \Delta_t(R)\, y_t \\ \geq \sum_{j=1}^{K} \left(\hat{d}_{i_j} - \hat{d}_{i_{j-1}}\right) \left(\lfloor \bar{d}_{u,i_j} \rfloor + 1\right) + \left(1 - \hat{d}_{i_K}\right) \lfloor \bar{d}_{u,i_1} \rfloor \end{aligned} \tag{5.32}$$

Inequalities (5.31) and (5.32) are valid for our problem and can be separated in $O(|V| log |V|)$ time as it is described in Section 2.5. Note that these inequalities are derived for fixed u, and in our computational experiments we consider these inequalities for all $u \in V$.

5.4 Computational Experiments

In this section, we test the valid inequalities developed in the previous section.

Note that, there are several parameters for controlling the nervousness caused by the production plan obtained by our formulation. Different values of λ', λ, δ, δ' and different forms of g may lead to different problem settings:

1. If $\lambda' = \lambda$ and $\delta' = \delta$, then there is no nervousness cost. But note that, with λ and δ parameters, it is still possible to decrease quantity oriented nervousness by controlling range of the production amounts by constraints (5.5).

2. If $\lambda_t = \lambda'_t = 0$, $\delta_t = \delta'_t$ and δ_t is sufficiently large, i.e. $\delta_t = \frac{\max_{i \in V_t} C_t}{p-u}$, then the formulation is equivalent to the static - dynamic uncertainty strategy as production periods are determined beforehand and production amount of each period is determined as an answer to the demand realization. In this case, variables $z_{t(i)}, x_{1i}, x_{2i}$ and constraints $(5.5)-(5.7)$ can be ignored.

3. If $\lambda_t = \lambda'_t = \delta_t = \delta'_t = 1$ for all t, then the formulation is equivalent to static uncertainty strategy since production decision for a period is the same regardless of the demand realization.

4. If $a_1 = b_1$ and $a_2 = b_2$, then the nervousness cost is a piecewise linear function (with at most three pieces): $g_i\left(x_i, z_{t(i)}\right) = \kappa_{1i}\left[\lambda'_{t(i)} z_{t(i)} - x_i\right]^+ + \kappa_{2i}\left[x_i - \delta'_{t(i)} z_{t(i)}\right]^+$ for $x_i \in \left[\lambda_{t(i)} z_{t(i)}, \delta_{t(i)} z_{t(i)}\right]$ (see Figure 5.4(i)).

5. If $a_1 = b_1$, $a_2 = b_2$ and $\lambda'_{t(i)} = \delta'_{t(i)} = 1$, then the nervousness cost is a piecewise linear function (with at most two pieces): $g_i\left(x_i, z_{t(i)}\right) = \kappa_{1i}\left[z_{t(i)} - x_i\right]^+ + \kappa_{2i}\left[x_i - z_{t(i)}\right]^+$ for $x_i \in \left[\lambda_{t(i)} z_{t(i)}, \delta_{t(i)} z_{t(i)}\right]$ (see Figure 5.4(ii)).

6. If $\lambda'_{t(i)} = \delta'_{t(i)} = 1$, then the nervousness cost function is given by $g_i\left(x_i, z_{t(i)}\right) = \kappa_{1i}\left(\left[z_{t(i)} - x_i\right]^+\right)^{a_1/b_1} + \kappa_{2i}\left(\left[x_i - z_{t(i)}\right]^+\right)^{a_2/b_2}$ for $x_i \in \left[\lambda_{t(i)} z_{t(i)}, \delta_{t(i)} z_{t(i)}\right]$ (see Figure 5.4(iii)).

In our computational experiments we consider problem under the first setting in which there is no nervousness cost. Moreover, we assume that the compression

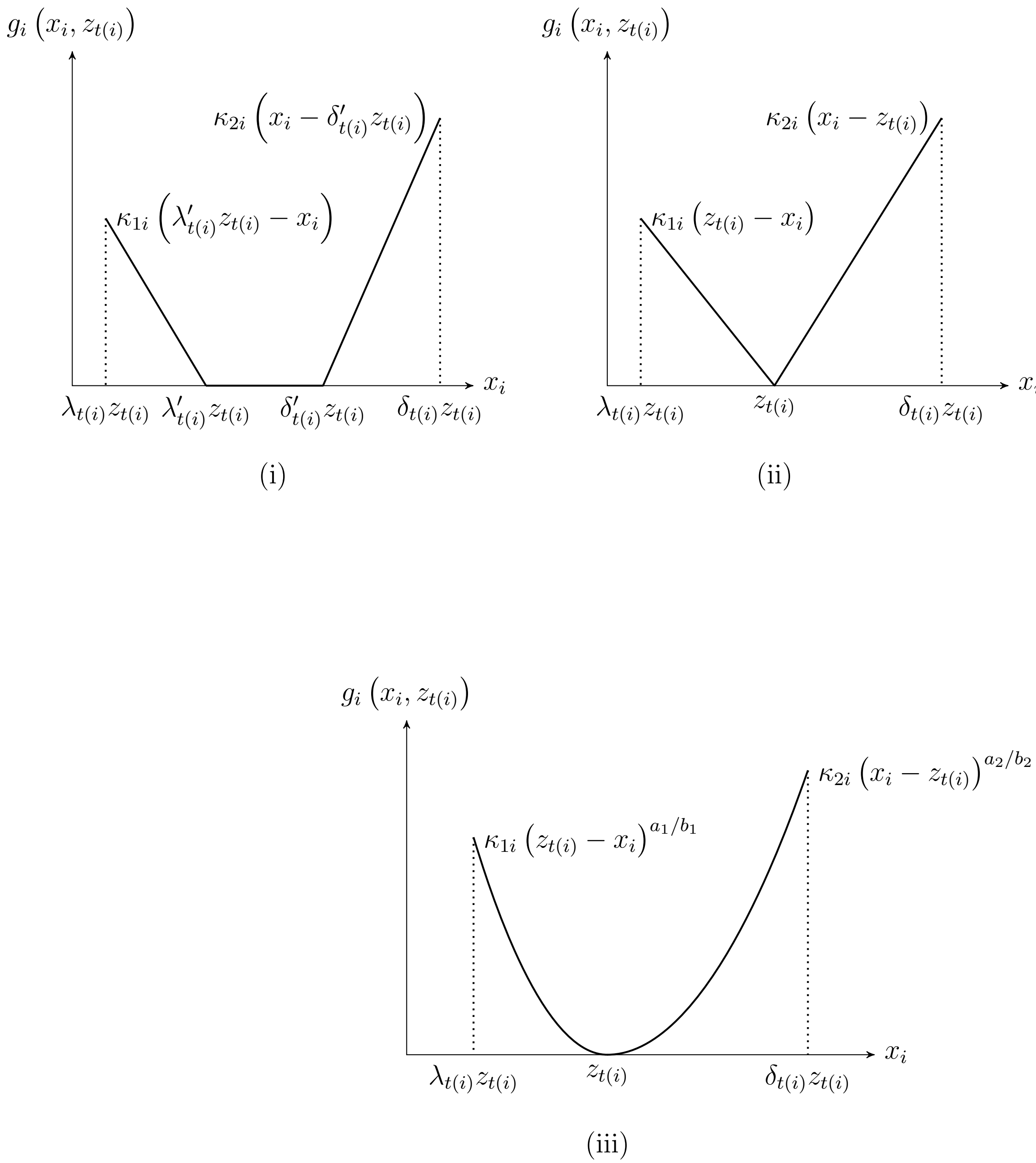

Figure 5.4: Special cases of the nervousness cost function

cost function is quadratic. We first tested the following formulation along with the valid inequalities developed in the previous section.

$LS - Q$

$$\begin{aligned}
\min \; & \sum_{t=1}^{T} f_t y_t + \sum_{i \in V} (c_i x_i + h_i s_i) + \sum_{i \in V} \kappa_i r_i^2 \\
s.t. \; & s_{i^-} + x_i = d_i + s_i \qquad i \in V, \\
& x_i \leq C_{t(i)} y_{t(i)} + r_i \qquad i \in V, \\
& r_i \leq u x_i \qquad i \in V, \\
& \lambda_{t(i)} z_{t(i)} \leq x_i \leq \delta_{t(i)} z_{t(i)} \qquad i \in V, \\
& x_i, s_i, r_i \geq 0 \qquad i \in V, \\
& y_t \in \{0,1\}, \; z_t \geq 0 \qquad t = 1, \ldots, T.
\end{aligned}$$

We consider the following parameter settings: $m = 2$ (number of children of each node), $p = 1$, $u = 0.6$, $d_i \sim U[1, 1000]$, $c_i \sim \pi_i U[20, 30]$, $h_i \sim \pi_i U[1, 10]$, $\kappa_i \sim \pi_i U[0.04, 0.06]$, $\lambda_{t(i)} = 0.5$, , $\delta_{t(i)} = 1.3$, for $i \in V$. We consider different values of setup costs and capacities: $f_t \sim U[4\bar{f}, 6\bar{f}]$, and $C = \bar{c}T$ where $\bar{f} \in \{1000, 2000, 3000\}$ and $\bar{c} \in \{150, 200, 250, 300\}$. We assume that all of the nodes defined for the same time period have equal probabilities (π_i). These parameter settings are very similar to the ones considered in [57]. In order to see the effect of inequalities we developed, we disable presolve and the automatic cuts of Cplex.

We first test the cycle inequalities developed in Section 4.1. We add the extended formulation (5.22) - (5.26) for each node $u \in V$ to the formulation LS-Q and solve the resulting formulation by Cplex. Remember that for given u, inequalities (5.22) - (5.26) are developed for all of the descendant nodes of u ($V(u)$). We also consider inequalities (5.19) for all paths starting at node u and add the extended formulation (5.22) - (5.26) for each $P(u, v)$ for all $v \in V(u)$. Note that in the development of the extended formulation it is assumed that there is an arc between each node in $V(u)$, and we exclude the arcs that should be defined between nodes of different paths by considering the extended formulation for each path separately. We test both of these approaches for small problem instances.

Table 5.1: Test of different extended formulations

$\bar{f}$	C	opt	LS-Q			LS-Q with reformulation					
						for $V(u)$			for paths		
			rgap	cpu	node	rgap	cpu	node	rgap	cpu	node
1000	900	128,487.2	3.36	0	3	1.07	59	1	1.07	7	0
	1200	133,403.6	9.05	0	12	2.43	81	5	2.43	10	6
	1500	143,665.6	9.76	0	10	2.06	73	9	2.06	10	9
	1800	134,126.8	12.59	0	38	2.95	99	11	2.95	15	13
2000	900	163,505.9	5.82	0	4	2.33	56	4	2.33	7	4
	1200	162,495.3	11.73	0	15	3.82	69	10	3.82	13	10
	1500	119,990.6	17.86	0	23	4.48	81	6	4.49	12	5
	1800	141,390.1	19.02	0	45	5.84	76	15	5.84	16	9
3000	900	188,323.0	9.47	0	12	4.11	65	9	4.12	11	9
	1200	189,706.0	9.89	0	17	3.43	68	7	3.43	9	4
	1500	174,669.1	14.06	0	21	4.5	79	9	4.5	14	9
	1800	134,159.3	16.57	0	26	3.14	74	4	3.14	15	4

Results for 6 period instances can be seen in Table 5.1. In this table, the columns opt, rgap, cpu and node represent the optimal value of the problem instance, root gap (the percentage gap between the best lower bound obtained in the root node and the optimal value), solution time (in seconds) and the number of branch-and-bound nodes explored, respectively. Note that all the problem instances are solved in less than one second by the formulation LS-Q, but the main aim of this experiment is to see the improvement of root gap due to the reformulation. We observe that including the extended formulation reduces the root gap in all of the problem instances. The root gaps are reduced by more than 50% in all the problem instances. In one of the instances the root gap is decreased from 16.57 % to 3.14%. Moreover root gaps of the reformulations are the same for both of the approaches. But adding the extended formulation for each path separately causes less computation time.

We consider the same experiment for larger instances. But even for 7 period instances, the first approach (adding the extended formulation for $V(u)$) fails to find a feasible solution in 1000 seconds. Thus, we conduct our experiments for the second approach.

Results for 7 and 8 period instances are given in Table 5.2. In this table, if

Table 5.2: Test of the extended formulation

				LS-Q			LS-Q with reformulation		
T	$\bar{f}$	C	opt	rgap	cpu	node	rgap	cpu (fgap)	node
7	1000	1050	128,308.6	8.26	0	15	2.16	99	10
		1400	129,909.0	12.98	0	33	3.79	150	13
		1750	123,033.5	12.18	0	28	3.54	133	6
		2100	151,148.1	15.91	0	90	3.19	126	17
	2000	1050	162,314.4	11.2	0	45	5.33	143	16
		1400	155,902.9	12.49	0	48	5.33	153	18
		1750	185,823.6	15.3	0	63	3.68	166	15
		2100	181,497.6	17.24	0	63	3.83	168	15
	3000	1050	194,478.1	13.15	0	39	7.06	145	21
		1400	201,006.4	13.42	0	27	3.24	145	12
		1750	171,014.1	20.05	0	54	7.05	264	13
		2100	184,043.4	19.48	0	56	9.48	175	19
8	1000	1200	136,369.1	9.7	1	70	3.96	(3.96)	0
		1600	166,840.0	10.75	1	41	2.73	849	14
		2000	139,693.2	17.36	2	98	!		
		2400	153,089.3	14.71	2	96	!		
	2000	1200	220,515.3	10.62	1	52	3.89	956	24
		1600	194,126.3	11.65	1	51	4.32	(1.65)	15
		2000	181,875.2	17.56	2	88	!		
		2400	176,705.6	21.09	2	71	!		
	3000	1200	201,392.1	15.69	1	51	12.13	(12.13)	0
		1600	222,255.5	15.24	1	45	!		
		2000	202,682.8	18.71	2	64	!		
		2400	187,344.2	26.12	2	78	!		

!: no solution is obtained in 1000 seconds

the problem instance cannot be solved to optimality within the time limit (1000 seconds), the final percentage gap is reported under the column (fgap). For 7 period instances, the root gap is again reduced by more than 50% due to the reformulation. Besides, the number of branch and bound nodes explored decreases in all of the problem instances. However, solution times increase significantly due to increasing formulation sizes by the addition of inequalities. For most of the 8 period instances, even a feasible solution cannot be found in 1000 seconds. Therefore, adding all the inequalities is not a good option for solving larger instances. On the other hand, since these inequalities reduce the root gaps significantly, they may help solving larger instances in less computation time.

Table 5.3: Test of the partial extended formulation

			$n' = 2$			$n' = 3$			$n' = 4$		
T	$\bar{f}$	C	rgap	cpu	node	rgap	cpu	node	rgap	cpu	node
7	1000	1050	2.17	1	6	2.16	2	10	2.16	7	10
		1400	3.89	1	17	3.8	3	15	3.79	9	16
		1750	3.54	1	9	3.54	2	8	3.54	6	6
		2100	3.19	1	18	3.19	3	17	3.19	9	18
	2000	1050	5.33	1	18	5.33	3	20	5.33	9	12
		1400	6.24	1	24	5.33	4	18	5.33	10	18
		1750	3.73	2	29	3.7	4	15	3.68	11	15
		2100	3.84	2	16	3.84	4	16	3.83	11	15
	3000	1050	7.18	2	26	7.16	4	23	7.06	10	22
		1400	3.32	1	8	3.24	3	9	3.24	12	9
		1750	7.27	2	17	7.25	4	16	7.18	10	15
		2100	9.77	2	26	9.53	4	26	9.48	14	28
8	1000	1200	4.61	5	49	4.6	12	37	3.96	41	23
		1600	2.83	3	13	2.74	12	16	2.74	39	15
		2000	5.29	5	31	5.29	12	33	5.29	38	38
		2400	2.89	3	19	2.89	11	18	2.89	57	21
	2000	1200	4	4	16	3.97	13	20	3.89	43	19
		1600	4.5	4	33	4.33	13	23	4.33	37	24
		2000	8.16	4	26	8.15	14	24	8.15	42	26
		2400	4.45	4	15	4.45	12	15	4.45	33	14
	3000	1200	7.68	6	62	7.66	19	55	7.45	53	48
		1600	4.69	4	14	4.65	13	11	4.65	35	14
		2000	6.27	5	18	6.27	16	34	6.21	46	27
		2400	5.2	4	8	5.2	12	15	5.2	39	15

In order to make use of the extended formulation for solving larger instances, we add the extended formulation for paths with maximum n' nodes. In other words, we add the extended formulation for inequality (5.19) for the path $P(u, v)$ if $t(v) - t(u) \leq n' - 1$. We first consider different n' values in order to see the effect of the reformulation for different n' values. In Table 5.3, results of the partial extended formulation for the problem instances considered in Table 5.2 are given. Note that for different n' values the root gaps are every close to each other and to the root gaps of the formulation with the full extended formulation (Table 5.2). For example, for one of the instances, adding the extended formulation for all paths including at most two nodes decreases the root gap from 26.12% to 5.2%. Moreover, smaller n' values cause less computation times. All the instances are solved to optimality in less than 6 seconds by the reformulation with $n' = 2$.

Table 5.4: Results for larger instances

T	$\bar{f}$	C	opt	LS-Q rgap	LS-Q cpu (fgap)	LS-Q node	LS-Q with reform. & mixing rgap1	rgap2	cpu (fgap)	node
12	1000	1800	232,999.4	13.35	744	461	7.61	7.49	558	282
		2400	240,818.3	16.71	544	442	9.44	8.22	387	149
		3000	253,305.6	16.36	622	670	9.07	6.38	372	65
		3600	237,514.3	19.4	850	671	10.2	5.79	429	141
	2000	1800	263,318.3	16.07	528	263	8.61	8.52	329	108
		2400	265,892.2	17.01	668	321	10.57	8.98	449	148
		3000	270,252.4	22.76	780	457	11.1	7.91	378	89
		3600	275,400.7	24.93	899	682	13.51	9.56	582	224
	3000	1800	282,766.5	15.96	416	154	8.29	7.86	339	51
		2400	279,501	20.37	524	221	12.89	11.09	511	192
		3000	291,605.8	22.54	456	272	13.78	10.89	492	163
		3600	289,074.9	29.34	954	538	16.98	13.19	573	162
13	1000	1950	270,926.8	12.6	3438	506	7.6	7.51	1957	195
		2600	247,753.1	16.83	(5.85)	500	10.41	8.99	(2.86)	188
		3250	238,439.7	19.39	(5.99)	540	11.71	10.35	3550	251
		3900	246,249.6	18.58	(5.04)	618	11.42	6.92	2474	216
	2000	1950	295,713.8	16.44	3196	396	10.49	10.2	2269	248
		2600	300,889.3	18.57	3486	590	11.27	10.65	2087	241
		3250	286,874.5	20.96	(5.38)	619	13.37	12.37	2547	290
		3900	270,537.6	23.53	3317	616	13.34	9.14	2750	114
	3000	1950	306,009	16.87	1793	210	10.44	10.15	2237	197
		2600	292,159.6	20.47	2870	307	13.09	12.7	3574	234
		3250	298,256.1	24.27	3120	421	16.13	15.28	2792	279
		3900	300,704.4	29.5	(8.31)	540	17.29	13.67	2073	183

For solving larger problem instances, we make use of the partial extended formulation and the mixing inequalities (5.31) and (5.32). We first add the partial extended formulation for $n' = 2$ for the nodes of the scenario tree defined for the first 5 periods. Then, we solve the relaxation of the reformulation and call the separation routine of mixing inequalities in the root node. The separation routine is called maximum 15 times and at each round at most 100 mixing inequalities are added. We consider 12 and 13 period instances and set the time limit to 3600 seconds.

Results of this experiment can be seen in Table 5.4. In this table, value of the optimal solution is given under the column opt, the percentage gap between the best lower bound obtained in the root node before calling the separation routine and the optimal value is given under the column rgap1, and the root gap after the addition of valid (mixing) inequalities is reported under rgap2. If the problem is solved to optimality within the time limit the solution time is reported under the column cpu. But if the solver terminates with positive optimality gap, the final percentage gap is given under the same column in parenthesis. Note that one of the problem instances cannot be solved by both of the formulations, but the optimal value for this problem instance is obtained in our next experiment.

As it can be observed from Table 5.4, the root gaps are decreased by the addition of the partial extended formulation and the mixing inequalities in all of the instances. For 12 period instances, the reformulation reduces the root gap by 40% in the average (51% maximum), and the mixing inequalities reduce this gap about 20% (47% maximum). Average solution time of LS-Q is about 650 seconds and LS-Q with the reformulation and mixing inequalities solves these instances in 450 seconds in the average. Similar results are observed for 13 period instances. For these instances the root gap is reduced about 35% (43% maximum) and 15%(62% maximum) by the addition of the extended formulation and the valid inequalities, respectively. Note that, for five of the 13 period instances the formulation LS-Q terminates with positive optimality gap at the end of 3600 seconds. On the other hand, only one problem instance cannot be solved within the time limit by LS-Q with the reformulation and the mixing inequalities. But the final gap is reduced from 5.8% to 2.8% for this instance. Moreover, LS-Q with the reformulation and the mixing inequalities solves the problem instances by exploring less branch and bound nodes.

All the experiments explained above are conducted for the formulation LS-Q. Now we will consider the reformulation CMLSII given in Section 5.2 and conduct the same experiment for the 12 and 13 period instances considered in Table 5.4. Results of this experiment are given in Table 5.5. Note that only one problem instance cannot be solved within the time limit by CMLSII, and all the problem instances are solved to optimality by CMLSII with reformulation and mixing inequalities. The root gaps of LS-Q and CMLSII are very close to each other, but CMLSII solves the instances in less computation time in the average: 12

Table 5.5: Results for CMLSII

			CMLSII			**CMLSII with reform. & mixing**			
T	$\bar{f}$	C	rgap	cpu (fgap)	node	rgap1	rgap2	cpu	node
12	1000	1800	13.27	550	642	7.48	7.38	306	179
		2400	16.65	602	690	9.35	8.96	334	185
		3000	16.33	623	1418	8.99	8.49	224	136
		3600	19.38	503	1133	10.11	9.36	367	142
	2000	1800	15.75	407	419	8.15	8.06	349	126
		2400	16.81	391	396	10.06	9.67	278	92
		3000	22.63	571	875	10.7	10.15	358	134
		3600	24.82	775	962	12.94	12.28	411	165
	3000	1800	15.44	361	178	7.25	7	329	81
		2400	20	577	286	11.94	11.48	430	133
		3000	22.17	386	468	12.92	12.21	333	139
		3600	29.12	535	678	14.88	14.1	416	152
13	1000	1950	12.52	1908	705	7.5	7.42	1273	229
		2600	16.77	2668	1323	10.21	9.98	1262	168
		3250	19.35	2651	1543	11.59	11.28	1148	224
		3900	18.55	1887	1434	11.37	10.99	1220	285
	2000	1950	16.16	1927	332	9.5	9.34	1622	138
		2600	18.36	2561	970	10.81	10.59	1433	177
		3250	20.83	(4.66)	703	12.94	12.61	1626	290
		3900	23.43	2806	1635	12.88	12.42	1564	170
	3000	1950	16.33	2813	321	9.43	9.35	1412	133
		2600	20.1	1594	251	12.14	11.88	1508	136
		3250	23.98	2235	519	15.18	14.85	1714	247
		3900	29.3	2168	845	16.22	15.73	1494	183

period instances are solved in 500 seconds in the average, and only one 13 period instance cannot be solved within the time limit and the average solution for the other 13 period instances is about 2300 seconds. We obtain the best solution times by reformulating and adding valid inequalities to CMLSII for all of the problem instances: solution time is about 300 and 1400 seconds for 12 period and 13 period instances, respectively. Note that the root gaps after the reformulation (rgap1) are better for the formulation CMLSII. But mixing inequalities does not perform so good for this reformulation: the root gaps (rgap2) given in Table 5.4 are better for 15 of the instances.

Overall, we observed that conic strengthening, reformulating and adding valid inequalities reduces the root gap and the solution time in our problem instances. Significant improvement is obtained due to the reformulation, but conic strengthening and adding valid inequalities also reduce the solution times.

5.5 Conclusions

In this chapter, we study the multistage stochastic lot sizing problem with controllable processing times. In order to reduce the system nervousness caused by multistage stochastic programming approach, we restrict the production decisions under different scenarios. To the best of our knowledge, this problem is not studied before. We develop valid inequalities for the problem based on continuous mixing and mixing set relaxations of the problem. We computationally test these inequalities and observe that these inequalities reduce the root gaps and solution times. Although these inequalities strengthen the formulation, root gaps and solution times may be further reduced by new inequalities. In a future research, this formulation may be further improved by the development of stronger valid inequalities. Moreover, in our computational experiments we assume that there is no nervousness cost and the compression cost function is quadratic. Similar experiments can be considered for different compression cost functions and nervousness costs. Note that determination of the nervousness cost parameters is also a problem and this problem can be considered in a future research. Furthermore, new solution techniques for solving larger instances should be searched for since even a feasible solution cannot be obtained for larger instances (for example, 14 period instances) in 3600 seconds.

Chapter 6

Conclusion

In this dissertation, we focused on different variations of the lot sizing problem. First, we considered the deterministic lot sizing problem with piecewise concave production cost functions. Piecewise concave functions can represent decreasing marginal costs, quantity discounts, subcontracting, minimum production quantities, and capacities. Thus, this problem is a generalization of many problems studied in the literature. Computational complexity of this problem was an open question in the literature. We developed a dynamic programming algorithm to solve this problem in polynomial time when the breakpoints of the production cost function are time invariant and the number of breakpoints is fixed. Thus, we showed that the problem is polynomially solvable under our assumptions. We computationally tested the DP algorithm and observed that it performs quite well for small and medium sized instances. Solution time of the algorithm increases with the number of periods and breakpoints. We designed a DP based heuristic to obtain good quality solutions for larger instances in reasonable computation time.

Next, we considered the stochastic lot sizing problem where demand follows a stochastic process. Stochastic lot sizing problems are more appropriate for modeling real life problems since in most of the production, procurement and transportation systems demand may not be known with certainty. We studied this problem with controllable processing times where processing times can be reduced in return for extra compression cost. We assumed that the compression cost is a convex function of the compression amount. This problem was not studied in the literature before. We formulated the problem as a SOCMIP, strengthened this formulation and solved it by a commercial solver. As a result of our computational experiments, we observe that conic strengthening performs quite well, and it improves gaps in the root node. Consequently, we can solve realistic size problems in a reasonable computation time using an exact approach. Moreover, these formulations maybe further improved by adding new valid inequalities in a future research.

As a result of our computational experiments we concluded that controllable processing times are more valuable when the system has medium sized capacities, and larger setup costs. By relaxing the fixed processing time assumption and by utilizing the recent advances in second order cone programming, we could decrease the overall production costs significantly. Although we explored these results for the static uncertainty strategy under α service level constraints, these results are also applicable to the deterministic lot sizing problem. As a future research direction, it is possible to consider the same problem under different settings like considering different uncertainty strategies, service level constraints or joint chance constraints.

As a final problem, we studied the multistage stochastic lot sizing problem with nervousness considerations. System nervousness is one of the main problems of dynamic solution strategies developed for stochastic lot sizing problems. When the demand follows a discrete distribution, the problem can be formulated as a multistage stochastic program. In the studies on the the multistage stochastic lot sizing problem, production decisions are considered for each node separately. Thus existing formulations contain both setup and quantity oriented nervousness. We formulated the problem so that nervousness of the system is reduced by restricting the recourse actions. We considered this problem with controllable processing times and formulated the problem as SOCMIP. We observed that due to the compression decisions, this formulation contains the continuous mixing set as a relaxation. We developed valid inequalities for the problem based on this relaxation and tested these inequalities computationally.

Several special cases of the third problem can be studied as future research problems. For example, the multistage stochastic lot sizing problem with restricted production quantities is a special case of the third problem in which controllable processing times and convex nervousness cost functions are excluded from the formulation. Moreover, the third problem can be considered with controllable processing times but linear cost functions. For these special cases, some optimal solution properties may be searched for and dynamic programming algorithms based on these properties may be developed.

The interest in environmental issues has grown in the last decade. Lot sizing with carbon emission constraints is recently studied by Absi et al. [105] and Helmrich et al. [106]. In these studies carbon emission constraints are assumed either linear or concave. Note that, convex functions may represent increasing marginal costs and may limit higher resource usage due to environmental issues. Thus, lot sizing with convex carbon emission constraints may be an interesting future research problem.

In our second and third problems, we assumed that the processing times are controllable and studied these problems with setup costs. These problems can be considered with setup times. Note that setup times can be used in place of (or together with) setup costs. Lot sizing problem with controllable setup times may be an interesting extension of the problems considered in this thesis.

Bibliography

[1] S. Ghosh Dastidar and R. Nagi, "Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs," *Computers & Operations Research*, vol. 32, no. 11, pp. 2987–3005, 2005.

[2] D. Gupta and T. Magnusson, "The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times," *Computers & Operations Research*, vol. 32, no. 4, pp. 727–747, 2005.

[3] A. Dumoulin and C. Vercellis, "Tactical models for hierarchical capacitated lot-sizing problems with setups and changeovers," *International Journal of Production Research*, vol. 38, no. 1, pp. 51–67, 2000.

[4] R. Jans and Z. Degraeve, "An industrial extension of the discrete lot-sizing and scheduling problem," *IIE Transactions*, vol. 36, no. 1, pp. 47–58, 2004.

[5] L. A. Wolsey, "Solving multi-item lot-sizing problems with an mip solver using classification and reformulation," *Management Science*, vol. 48, no. 12, pp. 1587–1602, 2002.

[6] B. Karimi, S. M. T. Fatemi Ghomi, and J. M. Wilson, "The capacitated lot sizing problem: a review of models and algorithms," *Omega*, vol. 31, no. 5, pp. 365–378, 2003.

[7] N. Brahimi, S. Dauzere-Peres, N. M. Najid, and A. Nordli, "Single item lot sizing problems," *European Journal of Operational Research*, vol. 168, no. 1, pp. 1–16, 2006.

[8] H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Management Science*, vol. 5, no. 1, pp. 89 – 96, 1958.

[9] M. Florian and M. Klein, "Deterministic production planning with concave costs and capacity constraints," *Management Science*, vol. 18, pp. 12 – 20, 1971.

[10] A. Atamtürk and D. S. Hochbaum, "Capacity acquisition, subcontracting, and lot sizing," *Management Science*, vol. 47, no. 8, pp. 1081–1100, 2001.

[11] B. Hellion, F. Mangione, and B. Penz, "A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs," *European Journal of Operational Research.*, vol. 222, no. 1, pp. 10–16, 2012.

[12] M. Florian, J. K. Lenstra, and A. H. G. R. Kan, "Deterministic production planning: Algorithms and complexity," *Management Science*, vol. 26, no. 7, pp. 669–679, 1980.

[13] T. De Kok and K. Inderfurth, "Nervousness in inventory management: comparison of basic control rules," *European Journal of Operational Research*, vol. 103, no. 1, pp. 55–82, 1997.

[14] A. F. Veinott, "Production planning with convex costs: A parametric study," *Management Science*, vol. 10, no. 3, pp. 441–460, 1964.

[15] W. I. Zangwill, "A deterministic multi-period production scheduling model with backlogging," *Management Science*, vol. 13, pp. 105–119, 1966.

[16] G. R. Bitran and H. H. Yanasse, "Computational complexity of the capacitated lot size problem," *Management Science*, vol. 28, pp. 1174–1186, 1982.

[17] S. M. Johnson, "Sequential production planning over time at minimum cost," *Management Science*, vol. 3, pp. 435–437, 1957.

[18] E. H. Bowman, "Production scheduling by the transportation method of linear programming," *Operations Research*, vol. 4, pp. 100–103, 1965.

[19] G. D. Eppen and F. J. Gould, "A lagrangian application to production models," *Operations Research*, vol. 16, pp. 819–826, 1968.

[20] V. Lotfi and W.-H. Chen, "An optimal algorithm for the multi-item capacitated production planning problem," *The Journal of the Operational Research*, vol. 52, pp. 179–193, 1991.

[21] K. S. Hindi, "Algorithms for capacitated, multi-item lot-sizing without set-ups," *the Journal of the Operational Research Society*, vol. 46, no. 4, pp. 465–472, 1995.

[22] R. K. Ahuja and D. S. Hochbaum, "Solving linear cost dynamic lot-sizing problems in O(nlogn) time," *Operations Research*, vol. 56, no. 1, pp. 255–261, 2008.

[23] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[24] S. S. Erenguc and Y. Aksoy, "A branch and bound algorithm for single item nonconvex dynamic lot sizing problem with capacity constraints," *Computers & Operations Research*, vol. 17, no. 2, pp. 199–210, 1990.

[25] D. X. Shaw and A. P. M. Wagelmans, "An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs," *Management Science*, vol. 44, no. 6, pp. 831–838, 1998.

[26] C. P. M. van Hoesel and A. P. M. Wagelmans, "An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities," *Management Science*, vol. 42, pp. 142 – 150, 1996.

[27] W. I. Zangwill, "The piecewise concave function," *Management Science*, vol. 13, no. 11, pp. 900–912, 1967.

[28] C. Swoveland, "A deterministic multi-period production planning model with piecewise concave production and holding-backorder costs," *Management Science*, vol. 21, pp. 1007–1013, 1975.

[29] A. Federgruen and C.-Y. Lee, "The dynamic lot size model with quantity discount," *Naval Research Logistics*, vol. 37, pp. 707–713, 1990.

[30] C.-L. Li, J. Ou, and V. N. Hsu, "Dynamic lot sizing with all-units discount and resales," *Naval Research Logistics*, vol. 59, no. 3-4, pp. 230 – 243, 2012.

[31] L. Chan, A. Muriel, Z.-J. Shen, and D. Simchi-Levi, "On the effectiveness of zero-inventory-ordering policies for economic lot sizing model with a class of piecewise linear cost structures," *Operations Research.*, vol. 50, no. 6, pp. 1058–1067, 2002.

[32] C. Archetti, L. Bertazzi, and M. G. Speranza, "Polynomial cases of the economic lot sizing problem with quantity discounts," tech. rep., Department of Quantitative Methods, University of Brescia, 2011.

[33] B. Hellion, F. Mangione, and B. Penz, "Corrigendum to "A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs" [Eur. J. Oper. Res. 222 (2012) 10–16]," *European Journal of Operational Research.*, vol. 229, no. 1, p. 279, 2013.

[34] I. Okhrin and K. Richter, "An $O(T^3)$ algorithm for the capacitated lot sizing problem with minimum order quantities," *European Journal of Operational Research.*, vol. 211, no. 3, pp. 507–514, 2011.

[35] E. Silver, "Inventory control under a probabilistic time-varying, demand pattern," *AIIE Transactions*, vol. 10, no. 4, pp. 371–379, 1978.

[36] J. B. Laserre, C. Bes, and F. Roubellat, "The stochastic discrete dynamic lot size problem: an open-loop solution," *Operations Research*, vol. 33, no. 3, pp. 684 – 689, 1985.

[37] J. H. Bookbinder and J.-Y. Tan, "Strategies for probabilistic lot-sizing problem with service-level constraints," *Management Science*, vol. 34, no. 9, pp. 1096 – 1108, 1988.

[38] S. Nahmias, *Production and Operations Analysis.* McGraw-Hill/Irwin, 1993.

[39] F. Y. Chen and D. Krass, "Inventory models with minimal service level constraints," *European Journal of Operational Research*, vol. 134, no. 1, pp. 120–140, 2001.

[40] V. Vargas, "An optimal solution for the stochastic version of the Wagner-Whitin dynamic lot-size model," *European Journal of Operational Research*, vol. 198, pp. 447 – 451, 2009.

[41] C. Sox, "Dynamic lot sizing problem with demand and non-stationary demand," *Operations Research Letters*, vol. 20, pp. 155 – 164, 1997.

[42] S. A. Tarım and B. G. Kingsman, "The stochastic dynamic production/inventory lot-sizing problem with service-level constraints," *International Journal of Production Economics*, vol. 88, no. 1, pp. 105–119, 2004.

[43] U. Özen, M. K. Doğru, and A. S. Tarim, "Static-dynamic uncertainty strategy for a single-item stochastic inventory control problem," *Omega*, vol. 40, no. 3, pp. 348–357, 2012.

[44] H. Tunç, O. A. Kılıç, S. A. Tarım, and B. Ekşioğlu, "A reformulation for the stochastic lot sizing problem with service-level constraints," *Operations Research Letters*, vol. 42, no. 2, pp. 161–165, 2014.

[45] R. Rossi, O. A. Kılıç, and S. A. Tarım, "Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing," *to appear in Omega*, 2014.

[46] H. Tunç, O. Kılıç, S. A. Tarım, and B. Ekşioğlu, "The cost of using stationary inventory policies when demand is non-stationary," *Omega*, vol. 39, no. 4, pp. 410–415, 2011.

[47] H. Tunç, O. A. Kılıç, S. A. Tarım, and B. Ekşioğlu, "A simple approach for assessing the cost of system nervousness," *International Journal of Production Economics*, vol. 141, no. 2, pp. 619–625, 2013.

[48] H. Tempelmeier, "A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint," *Omega*, vol. 39, pp. 627 – 633, 2011.

[49] S. Helber, F. Sahling, and K. Schimmelpfeng, "Dynamic capacitated lot sizing with random demand and dynamic safety stocks," *OR Spectrum*, vol. 35, pp. 75 – 105, 2013.

[50] L. F. Escudero and P. Kamesam, "On solving stochastic production planning problems via scenario modelling," *Top*, vol. 3, no. 1, pp. 69–95, 1995.

[51] S. Ahmed, A. J. King, and G. Parija, "A multi-stage stochastic integer programming approach for capacity expansion under uncertainty," *Journal of Global Optimization*, vol. 26, no. 1, pp. 3–24, 2003.

[52] P. Brandimarte, "Multi-item capacitated lot-sizing with demand uncertainty," *International Journal of Production Research*, vol. 44, no. 15, pp. 2997–3022, 2006.

[53] Y. Guan, S. Ahmed, G. L. Nemhauser, and A. J. Miller, "A branch-and-cut algorithm for the stochastic uncapacitated lot-sizing problem," *Mathematical Programming*, vol. 105, no. 1, pp. 55–84, 2006.

[54] Y. Guan, S. Ahmed, A. J. Miller, and G. L. Nemhauser, "On formulations of the stochastic uncapacitated lot-sizing problem," *Operations Research Letters*, vol. 34, no. 3, pp. 241–250, 2006.

[55] M. Di Summa and L. A. Wolsey, "Lot-sizing on a tree," *Operations Research Letters*, vol. 36, no. 1, pp. 7–13, 2008.

[56] Y. Guan, S. Ahmed, and G. L. Nemhauser, "Sequential pairing of mixed integer inequalities," *Discrete Optimization*, vol. 4, no. 1, pp. 21–39, 2007.

[57] Y. Guan, S. Ahmed, and G. L. Nemhauser, "Cutting planes for multistage stochastic integer programs," *Operations Research*, vol. 57, no. 2, pp. 287–298, 2009.

[58] N. Halman, D. Klabjan, M. Mostagir, J. Orlin, and D. Simchi-Levi, "A fully polynomial-time approximation scheme for single-item stochastic inventory control with discrete demand," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 674–685, 2009.

[59] Y. Guan, "Stochastic lot-sizing with backlogging: computational complexity analysis," *Journal of Global Optimization*, vol. 49, no. 4, pp. 651–678, 2011.

[60] J. Luedtke, "A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support," *Mathematical Programming*, vol. 146, pp. 219–244, 2014.

[61] R. G. Vickson, "Two single machine sequencing problems involving controllable job processing times," *AIIE transactions*, vol. 12, no. 3, pp. 258–262, 1980.

[62] R. K. Kayan and M. S. Aktürk, "A new bounding mechanism for the CNC machine scheduling problems with controllable processing times," *European Journal of Operational Research*, vol. 167, no. 3, pp. 624–643, 2005.

[63] M. S. Aktürk, A. Atamtürk, and S. Gürel, "A strong conic quadratic reformulation for machine-job assignment with controllable processing times," *Operations Research Letters*, vol. 37, no. 3, pp. 187–191, 2009.

[64] K. Jansen and M. Mastrolilli, "Approximation schemes for parallel machine scheduling problems with controllable processing times," *Computers & Operations Research*, vol. 31, no. 10, pp. 1565–1581, 2004.

[65] S. Gürel, E. Körpeoğlu, and M. S. Aktürk, "An anticipative scheduling approach with controllable processing times," *Computers & Operations Research*, vol. 37, no. 6, pp. 1002–1013, 2010.

[66] A. Türkcan, M. S. Aktürk, and R. H. Storer, "Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties," *IIE Transactions*, vol. 41, no. 12, pp. 1080–1095, 2009.

[67] D. Shabtay and M. Kaspi, "Minimizing the total weighted flow time in a single machine with controllable processing times," *Computers & Operations Research*, vol. 31, no. 13, pp. 2279–2289, 2004.

[68] D. Shabtay and M. Kaspi, "Parallel machine scheduling with a convex resource consumption function," *European Journal of Operational Research*, vol. 173, no. 1, pp. 92–107, 2006.

[69] D. Shabtay and G. Steiner, "Single machine batch scheduling to minimize total completion time and resource consumption costs," *Journal of Scheduling*, vol. 10, no. 4-5, pp. 255–261, 2007.

[70] D. Shabtay and G. Steiner, "A survey of scheduling with controllable processing times," *Discrete Applied Mathematics*, vol. 155, no. 13, pp. 1643–1666, 2007.

[71] A. Atamtürk and D. S. Hochbaum, "Capacity acquisition, subcontracting, and lot sizing," *Management Science*, vol. 47, no. 8, pp. 1081–1100, 2001.

[72] Y. Merzifonluoğlu, J. Geunes, and H. Romeijn, "Integrated capacity, demand, and production planning with subcontracting and overtime options," *Naval Research Logistics*, vol. 54, no. 4, pp. 433–447, 2007.

[73] H. Luss, "Operations research and capacity expansion problems: A survey," *Operations Research*, vol. 30, no. 5, pp. 907–947, 1982.

[74] S. Rajagopalan and J. M. Swaminathan, "A coordinated production planning model with capacity expansion and inventory management," *Management Science*, vol. 47, no. 11, pp. 1562–1580, 2001.

[75] G. Heisig, "Comparison of (s, S) and (s, nQ) inventory control rules with respect to planning stability," *International Journal of Production Economics*, vol. 73, no. 1, pp. 59–82, 2001.

[76] J. D. Blackburn, D. H. Kropp, and R. A. Millen, "A comparison of strategies to dampen nervousness in MRP systems," *Management Science*, vol. 32, no. 4, pp. 413–429, 1986.

[77] C. A. Yano and R. C. Carlson, "Interaction between frequency of rescheduling and the role of safety stock in material requirements planning systems," *International Journal of Production Research*, vol. 25, no. 2, pp. 221–232, 1987.

[78] J. Roberta Minifie and R. A. Davis, "Interaction effects on MRP nervousness," *The International Journal of Production Research*, vol. 28, no. 1, pp. 173–183, 1990.

[79] V. Sridharan and W. L. Berry, "Master production scheduling make-to-stock products: a framework for analysis," *The International Journal of Production Research*, vol. 28, no. 3, pp. 541–558, 1990.

[80] S. Kadipasaoglu and S. Sridharan, "Measurement of instability in multi-level MRP systems," *International Journal of Production Research*, vol. 35, no. 3, pp. 713–737, 1997.

[81] D. H. Kropp, R. C. Carlson, and J. V. Jucker, "Heuristic lot-sizing approaches for dealing with MRP system nervousness," *Decision Sciences*, vol. 14, no. 2, pp. 156–169, 1983.

[82] D. H. Kropp and R. C. Carlson, "A lot-sizing algorithm for reducing nervousness in MRP systems," *Management Science*, vol. 30, no. 2, pp. 240–244, 1984.

[83] K. Inderfurth, "Nervousness in inventory control: analytical results," *Operations-Research-Spektrum*, vol. 16, no. 2, pp. 113–123, 1994.

[84] G. Heisig, "Planning stability under (s, S) inventory control rules," *Operations-Research-Spektrum*, vol. 20, no. 4, pp. 215–228, 1998.

[85] H. Scarf, "The optimality of (s, S) policies in the dynamic inventory problem. mathematical methods in the social science, kj arrow, s. karlin, p. suppes, eds," 1960.

[86] O. A. Kilic and S. A. Tarim, "An investigation of setup instability in non-stationary stochastic inventory systems," *International Journal of Production Economics*, vol. 133, no. 1, pp. 286–292, 2011.

[87] K. R. Baker, "An experimental study of the effectiveness of rolling schedules in production planning," *Decision Sciences*, vol. 8, no. 1, pp. 19–27, 1977.

[88] O. Kazan, R. Nagi, and C. M. Rump, "New lot-sizing formulations for less nervous production schedules," *Computers & Operations Research*, vol. 27, no. 13, pp. 1325–1345, 2000.

[89] H. Vladimirou and S. A. Zenios, "Stochastic linear programs with restricted recourse," *European Journal of Operational Research*, vol. 101, no. 1, pp. 177–192, 1997.

[90] M. Van Vyve, "The continuous mixing polyhedron," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 441–452, 2005.

[91] O. Günlük and Y. Pochet, "Mixing mixed-integer inequalities," *Mathematical Programming*, vol. 90, no. 3, pp. 429–457, 2001.

[92] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*. Springer, 2006.

[93] A. J. Miller and L. A. Wolsey, "Tight formulations for some simple mixed integer programs and convex objective integer programs," *Mathematical Programming*, vol. 98, no. 1-3, pp. 73–88, 2003.

[94] E. Koca, H. Yaman, and M. S. Aktürk, "Lot sizing with piecewise concave production costs," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 767–779, 2014.

[95] M. Florian and M. Klein, "Deterministic production planning with concave costs and capacity constraints," *Management Science*, vol. 18, pp. 12 – 20, 1971.

[96] K. L. Croxton, B. Gendron, and T. L. Magnanti, "A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems," *Management Science*, vol. 49, no. 9, pp. 1268–1273, 2003.

[97] S. Sanjeevi and K. Kianfar, "Mixed n-step MIR inequalities: Facets for the n-mixing set," *Discrete Optim.*, vol. 9, no. 4, pp. 216 –235, 2012.

[98] E. Koca, H. Yaman, and M. S. Aktürk, "Stochastic lot sizing problem with controllable processing times," *Omega*, vol. 53, pp. 1–10, 2015.

[99] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*. Springer, 2006.

[100] K. L. Croxton, B. Gendron, and T. L. Magnanti, "A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems," *Management Science*, vol. 49, no. 9, pp. 1268–1273, 2003.

[101] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Society for Industrial and Applied Mathematics, 2001.

[102] L. van Norden and S. van de Velde, "Multi-product lot-sizing with a transportation capacity reservation contract," *European Journal of Operational Research*, vol. 165, no. 1, pp. 127–138, 2005.

[103] D. A. Serel, "Capacity reservation under supply uncertainty," *Computers & Operations Research*, vol. 34, no. 4, pp. 1192–1220, 2007.

[104] K. Huang and S. Küçükyavuz, "On stochastic lot-sizing problems with random lead times," *Operations Research Letters*, vol. 36, no. 3, pp. 303–308, 2008.

[105] N. Absi, S. Dauzère-Pérès, S. Kedad-Sidhoum, B. Penz, and C. Rapine, "Lot sizing with carbon emission constraints," *European Journal of Operational Research*, vol. 227, no. 1, pp. 55–61, 2013.

[106] M. J. R. Helmrich, R. Jans, W. van den Heuvel, and A. P. M. Wagelmans, "The economic lot-sizing problem with an emission capacity constraint," *European Journal of Operational Research*, vol. 241, no. 1, pp. 50–62, 2015.

VITA

Esra Koca was born on March 30, 1985 in Amasya, Turkey. She attended Kırıkkale Fen Lisesi in 2000. She graduated from Bilkent University Department of Industrial Engineering with a degree of honor in 2007. She got her M.S. degree from the same department in 2010. In her M.S. study, she worked as a research assistant in a project sponsored by Tofaş - FIAT under the guidance of Assoc. Prof. Emre Alper Yıldırım. In her Ph.D., she has studied with Prof. Hande Yaman and Prof. M. Selim Aktürk, and she has worked in two Tubitak projects (109M149 and 112M220) during this time.