Erasmus University Rotterdam
Erasmus School of Economics

General Formulation Of The Static Dynamic Stochastic Lot Sizing Problem Extended to Discrete Products With Arbitrary Demand Distributions

Philip Palim (526044)

Supervisor:	Professor Wilco van den Heuvel
Second assessor:	Professor Albert Wagelmans
Date final version:	2nd July 2023

Abstract

The stochastic lot sizing problem allows for the modeling and optimization of delivery schedules and quantities for goods which have stochastic demand. Given the quantity and diversity of actors who require such optimization, a generalized and time efficient solution is necessary. Thus, in this paper I replicate the methods of Tunc, Kilic, Tarim and Rossi (2018), who have created a generalized formulation for this problem, and solved it with an approximation and dynamic cut method. Their formulation treats continuous goods which have normal demand distributions. I extend their formulation and solution methods to the optimization of discrete items with arbitrary demand distributions.

1 Introduction

The stochastic lot sizing problem describes the problem which arises from a warehouse, store, or other entity deciding the quantity and timing of a resupply. We assume demand to be the realization of a known data generating process (DGP), positive holding costs for storing the good in question, a penalty for not being able to meet the demand of a customer immediately, and a positive fixed cost per resupply. Importantly, we assume that the DGP for demand in each time period is known before the first period. In this problem, a company incurs (and thus wishes to minimize) costs in three ways. The first comes from the amount of product in storage at any given time multiplied by the holding cost per unit. Accordingly, the amount of goods in storage should be kept to a minimum. Secondly, a cost is incurred when consumers wish to purchase a product and are either turned away or forced to wait for a restock, necessitating that stocks run out as infrequently as possible. Finally, because resupplying incurs a fixed cost, companies wish to minimize the number of times a reorder is called. Any two of these goals are trivially achievable, but when trying to achieve all three, tradeoffs must be made depending on firm specific factors such as the cost of missing or delaying a sale, the fixed costs of delivery, etc. The stochastic lot sizing problem is the mathematical formulation which models and solves this problem. The problem is distinct from the deterministic lot sizing problem because in a given time period demand is not known, and is instead the realization of a random variable.

The general formulation proposed by Tunc et al. (2018) assumes that the item being supplied and sold is continuous and infinitely divisible. This can be seen in the choice of continuous distributions for demand, as well as the lack of an integer restriction for resupply quantity ($q_{i,j}$). For many products (such as petroleum) this is true, but it is not possible to purchase half of a lego set, or three quarters of an SUV. Thus, in addition to replicating the results of Tunc et al. (2018), I adapt the formulation to work with discrete products, for which a user specifies probability mass functions (PMF's). This is helpful because it allows users to input empirical data (easily transformed into a PMF) instead of assuming a distribution. In addition, this method will work with any distribution, and is not limited to the normal distribution as the solution of Tunc et al. (2018) is. It is thus more realistic for many goods, as well as much more flexible in terms of which demand distributions can be incorporated.

It is immediately apparent that this problem is relevant to the efficient supply of stores, warehouses, gas stations, and more. Thus, the stochastic lot sizing problem is well studied, and solutions are widely applied. For this reason a general formulation such as that proposed

by Tunc et al. (2018) is necessary, and for this reason I ensure its accuracy and replicability, as well as that of their dynamic cut solution. The research question I investigate beyond the replication of their paper is whether this formulation can be modified and applied to the case in which products are discrete. This extension raises three subquestions, which are: How widely applicable and adaptable is this discrete formulation? Where and under what conditions can this new formulation can be solved in a feasible amount of time? Can the dynamic cut method be extended to this new formulation?

The paper is organized as follows: Section 2 describes other approaches to the stochastic lot sizing problem, the contribution of Tunc et al. (2018), the methods of dealing with this problem in a discrete context, and the contribution to the literature I make in this paper. Section 3 details the methods introduced by Tunc et al. (2018) and how this approach can be adapted to discrete products with arbitrary demand distributions. Finally, Section 4 presents the results from the replication and extension with a focus on the accuracy and computational feasibility of the methods discussed.

2 Literature Review

2.1 Background On Stochastic Lot Sizing Problem Solutions

To solve the static dynamic stochastic lot sizing problem one needs a reorder strategy which defines when to reorder and how much. There are many reorder policies and rules possible. For example, one can use an (s,q) reorder policy in which when supply falls to s one reorders q. an (r,s) policy is also possible, in which inventory is reviewed over a period r and reordered up to point s. Finally, one could follow an (s,S) policy, in which when supply falls below s one would reorder up to point S. The reorder policy of this paper (and that of Tunc et al. (2018)) works thusly; Before the start of the time periods in question, one determines when to order resupplies, and an order up to quantity. For example, if you have five widgets and an order up to quantity of ten, then you would order five additional widgets. This is the approach I use in this paper, and thus the problem addressed is how to create these resupply schedules and order up to quantities in an efficient and optimal manner.

As described by Tunc et al. (2018), previous solutions to the stochastic lot sizing problem have all required a custom optimization algorithm, or were limited in computation time, accuracy, or the types of problems they can solve. Many instances of the more specialized and complicated solutions are simply extensions on the heuristic provided by Silver and Meal (1973). Other solution methods include constraint programming, filtering techniques, and preprocessing methods (Tunc et al., 2018). These types of solutions are limited in how widely they can be applied because they require custom optimization algorithms which can be difficult to implement. Of the solutions which do not require specialized algorithms, some authors approximated the loss function with a single or even multiple linear functions. This is similar to the a-priori approximation methods discussed later on in this paper (Tunc et al., 2018). However, as shown by Tunc et al. (2018), and in 4, this method can be computationally inefficient/infeasible as the time horizon increases and as such is not always a valid method. Thus, the paper by Tunc et al. (2018) provides a much more efficient dynamic cut method (RM-cut) to dramatically increase

the size of problems which can be feasibly solved.

To do this, Tunc et al. (2018) first introduce a unified mixed integer programming formulation for the stochastic lot sizing problem. The authors take the approach of setting the dates and setting a base-stock level for each of these reorder dates. Their formulation can include many different variations of the problem, thus making it a flexible general formulation of the stochastic lot sizing problem (Tunc et al., 2018). Due to their formulation being a mixed integer linear programming formulation, it is solvable with off the shelf solvers such as Cplex and Gurobi in reasonable time when using the RM-cut method (Tunc et al., 2018), or the a-priori approximation method given that the problem is not too large. The ability to solve a stochastic lot sizing problem with off the shelf solvers is very helpful, as the lack of specialized algorithms make this solution method easy to apply for practitioners, as well as to researchers who may wish to solve the stochastic lot sizing problem as a subproblem.

2.2 Extensions From the Literature

The generalized formulation of the stochastic lot sizing problem has been extended in several ways. A few characterizing examples will be quickly addressed here. Most of these papers, as stated directly by Bindewald, Dunke and Nickel (2023), use the formulation from Tunc et al. (2018) as a "unified modeling framework" on which one can build new features or compare new methods. This is to be expected for a model which claims to provide a generalized framework, and the fact that others are using it as one shows its success and validity as such. Notably, this formulation has been extended for use with multilevel problems Gruson, Cordeau and Jans (2021) and for controllable processing times (Tunc, 2021). We thus see that the general formulation is being used as intended as a basis for further research, and therefore improving this general formulation will be more helpful to the field than designing a single model or algorithm.

2.3 Discrete Products

Finally, my extension regards modeling discrete products instead of continuous ones. By this I mean that the units being studied cannot be divided and sold/shipped in arbitrarily small portions. This is the difference between oil, which can be divided into quarters, eighths, and sixteenths of a gallon, and a car, which cannot be shipped to a dealership or sold to a consumer in parts. I now review some of the ways the stochastic lot sizing problem has been modeled with discrete products, and the extent to which researchers have used the formulation by Tunc et al. (2018) for this purpose.

Huang and Küçükyavuz (2008) include discrete random variables in a lot sizing problem, but do not use the formulation from Tunc et al. (2018), and instead use a scenario tree and a custom dynamic programming algorithm to find an optimal solution. Their solution method is likely more complex because it is intended to solve a problem in which "the stochastic process is very general, i.e., cost, demand and lead time distributions are non-stationary and are correlated" (Huang & Küçükyavuz, 2008). In addition, their method was unsuitable for larger instances, variations, and had to be solved with a specialized and complicated algorithm. This shows the academic interest in the problem, as well as the lack of a generalized MILP (Mixed Integer Linear Programming) formulation.

Ma, Rossi and Archibald (2022) quickly examine this problem, but solve it with an (s, Q) policy with poisson demands and make no claims about whether or not their results can be generalized, or whether or not they compare favorably to other methods. An (s, Q) policy sets a reorder point and a reorder quantity such that whenever supplies dip below s, one reorders quantity Q. While the problem is the same, this policy is different than the one used in the general formulation from Tunc et al. (2018), in which order dates are established ahead of time and a base-stock level is established for each of those dates.

Finally, Gutierrez-Alcoba, Rossi, Martin-Barragan and Embley (2023) deal with a problem which contains the stochastic lot sizing problem as a subpart of their problem. They discretize their variables for their numerical experiments, but not within their model itself. Their results show a gap of a few percentage points between their results and optimal, although it is very difficult to say exactly where this comes from (Gutierrez-Alcoba et al., 2023). Note how the authors did not apply the more correct (though very complicated) formulation from Huang and Küçükyavuz (2008), but instead chose to simply discretize continuous units at the testing phase. This paper in particular shows the need to incorporate discrete products in the generalized formulation to provide researchers a general formulation for the discrete instance.

3 Methodology

3.1 Stochastic Lot Sizing Problem General Formulation

The notation for the formulation introduced by Tunc et al. (2018) is maintained for ease of reading and such that this paper may be integrated into the general literature. All notation used in this paper (unless mentioned otherwise) was created by Tunc et al. (2018), and credit should be extended accordingly.

The parameters of the formulation include: K as the fixed cost of resupply, h as the holding cost per unit per time period, p as the shortfall penalty, $D_{i,j}$ as realized demand between times i and j, and M is an arbitrarily large real number. The decision variables for this problem are as follows: $x_{i,j}$ is a variable which takes value 1 if $[i, j)$ is a replenishment cycle (restocks made directly before time i and directly before time j) and 0 if not, $q_{i,j}$ is the cumulative quantity expected to be ordered up to and including i if $[i, j)$ is a replenishment cycle and 0 if not, and finally $H_{i,j,t}$ is the approximated loss value function at period t during replenishment cycle $[i, j)$.

The objective function is as follows,

$$\min \sum_{i=1}^{N} \{ \sum_{j=i+1}^{N+1} [K x_{i,j} + \sum_{t=i}^{j-i} \{ (h(q_{i,j} - \mathbb{E}[D_{1,t} x_{i,j}]) + (h+p) H_{i,j,t}) \}] \} \tag{1}$$

and the constraints for the penalty cost model are as such:

$$\sum_{i=1}^{t-1} x_{i,t} = \sum_{j=t+1}^{N+1} x_{t,j}, t \in [2, N], \tag{2}$$

$$\sum_{j=2}^{N+1} x_{1,j} = 1, \tag{3}$$

$$\sum_{i=1}^{N} x_{i,N+1} = 1, \tag{4}$$

$$q_{i,j} \leq M x_{i,j}, i \in [1, N], j \in [i+1, N+1], \tag{5}$$

$$\sum_{i=1}^{t-1} q_{i,t} \leq \sum_{j=t+1}^{N+1} q_{t,j}, t \in [2, N], \tag{6}$$

$$H_{i,j,t} \geq a x_{i,j} + b(q_{i,j} - \mathbb{E}[D_{1,i-1} x_{i,j}]), i \in [1, N], j \in [i+1, N+1], t \in [i, j-1], (a, b) \in W_{i,t} \tag{7}$$

$$H_{i,j,t} \geq 0, i \in [1, N], j \in [i+1, N+1], t \in [i, j-1], \tag{8}$$

$$q_{i,j} \geq 0, i \in [1, N], j \in [i+1, N+1], \tag{9}$$

$$x_{i,j} \in 0, 1, i \in [1, N], j \in [i+1, N+1]. \tag{10}$$

The first part of the objective function is the fixed cost per replenishment. The second part is the cost of having too much or too little product stocked at any given time period. Equation (2) is a flow constraint which ensures that when one replenishment period ends another begins, and that one cannot begin or end alone. This of course exempts the first and the last period, in which a period begins or ends without a partner. Equation (3) ensures that a replenishment period begins at time 1. Equation (4) forces the last replenishment period to end during the last time period. The combination of the above three equations ensures that all periods are covered by one and only one replenishment period. Equation (5) forces $q_{i,j}$ to be zero when $x_{i,j}$ is zero, but because M is sufficiently large there is no effective constraint when $x_{i,j}$ is one. Equation (6) forces q to be monotonically increasing over all time period $t's$. Equation (7) forces $H_{i,j,t}$ to be greater than our linear approximations for the loss function. By generating a piecewise linear approximation of the loss function and then forcing $H_{i,j,t}$ to be greater than or equal to all of these pieces we approximate the nonlinear part of our objective function. Equations (9) and (10) simply define the x and q variables on the relevant indices.

3.2 Solving the Generalized Formulation

3.2.1 A-priori Approximation

In Equation (1), the variable $H_{i,j,t}$ is an approximation of a nonlinear function. One of the approaches to do this (as shown in constraint (7)) is to model the function through a-priori piecewise approximation (also referred to as PM). The a-priori piecewise approximation of the loss functions is based on the previous work by Rossi, Tarim, Prestwich and Hnich (2014) which seeks to create accurate upper and lower bounds for normal loss functions. These techniques were used to create a piecewise lower bound approximation from 11 linear pieces. While Rossi et al. (2014) effectively approximate lower and upper bounds for the normal distribution, the fact that a whole paper is required for this task hints at a problem: Every time someone wants to include a new distribution in this generalized model, they will be required to do a large amount of analytical research to estimate loss functions. This is particularly troubling, because one of the key contributions of Tunc et al. (2018) is that their formulation is generalized and easy to adapt. A generalized formulation that is highly restricted in the distributions it can model is a

clear limitation, and thus there is need for a more general approximation method.

The mechanics of the approximation work thusly: New decision variables are created for each function one wants to replicate. Here, that would be the H variables. The objective function is directly and obviously minimized when these variables take on the smallest value possible and will take the value of whatever lower bounds one sets. Thus, we set a lower bound through piecewise linear functions which take roughly the shape of the function we want to approximate. Thus, in the context of the problem as a whole, these H variables approximate the function. The equations for a and b used in Constraint (7) are given in the paper by Rossi et al. (2014), and provide all we need to determine Equation(7). The addition of x to the simple linear function is to ensure that when $x_{i,j}$ is zero the equation is nonbinding.

3.2.2 Dynamic Cut Method

As an alternative to approximating so many nonlinear functions, Tunc et al. (2018) introduce a cut generation approach that dramatically increases the speed and accuracy with which a general solver can solve this problem. The first step in the cut method is to create a relaxation of the original problem, referred to as RM. This is done by substituting constraint (9) with the constraint

$$H_{i,j,t} \geq -(q_{i,j} - \mathbb{E}[D_{1,t}x_{i,j}]), i \in [1, N], j \in [i+1, N+1], t \in [i, j-1]. \tag{11}$$

It is important to notice that this is essentially relaxing the accuracy with which each $H_{i,j,t}$ is being approximated. The more relaxed this constraint, the less accurate (and lower) the total objective function. Thus, we can solve the RM to optimality, and achieve a solution which underestimates the true objective function. We then evaluate the difference between this underestimation and the real value which we determine with our x and q variables. If this difference exceeds some value ϵ, then we include a new set of cuts. A single constraint is added to each $H_{i,j,t}$ that is underestimated. The new cut is a tangent line to the real cost function, perfectly estimating the cost at our temporary optimum, and doing a relatively good job in its proximity. Equation (14) is the final cut added to RM for each inaccurate $H_{i,j,t}$

$$b = F_{i,t}(q_{i,j} - \mathbb{E}[D_{1,i-1}]) - 1 \tag{12}$$

$$a = L_{i,t}(q_{i,j} - \mathbb{E}[D_{1,i-1}]) - b(q_{i,j} - \mathbb{E}[D_{1,i-1}]) \tag{13}$$

$$H_{i,j,t} \geq ax_{i,j} + b(q_{i,j} - \mathbb{E}[D_{1,i-1}x_{i,j}]). \tag{14}$$

After these new constraints (cuts) are added, the RM can be run again, finding a new 'optimal' solution and a new set of cuts. This repeats until the real and expected values are acceptably close (within a prespecified ϵ). We know that no other points are significantly better than the solution we settle on because our solution has a true objective equal to the objective value of the relaxed problem (referred to as g) plus ϵ. Additionally, since our temporary optimal is chosen, we know that no other feasible point has a relaxed objective less than g. While it is possible for another point to have a true objective equal to $g + 0$ instead of $g + \epsilon$, this would mean that our chosen solution is within ϵ of the true optimal objective value. Thus, by setting

epsilon sufficiently small we can get arbitrarily close to an optimal solution. For reasons of page constraints, the proofs which undergird this method are omitted from this paper, but can be seen in Tunc et al. (2018).

While showing improvements in accuracy and computational requirements, dynamic cut generation reintroduces an old problem: One of the benefits of a general formulation is the elimination of complicated and specialized algorithms, which make the solving of these problems difficult in practice (Tunc et al., 2018). Thus, the inclusion of a complicated and specialized cut generation algorithm reduces some of the progress this new formulation made in simplifying the modeling and optimization of stochastic lot sizing problems in practice.

3.3 Formulation and Solution Methods For Discrete Items

3.3.1 Changes In The Formulation

To adapt the formulation proposed by Tunc et al. (2018) we must force all resupplies and demands to be integers. For demands we achieve this by using a discrete distribution instead of a continuous one. This changes the loss function and how we approximate it, as discussed below. Forcing the resupplies to be integers is more simple in that it can be done entirely by introducing new variables and adding constraints in the Mixed Integer Linear Programming Problem. As defined by Tunc et al. (2018) the expression $q_{i,j} - \mathbb{E}[D_{1,i-1}]$ represents the order up to level if one is restocking at time period i and time period j. We force this expression to be an integer so that at every order up to level is achievable with discrete products. Crucially, neither the expectation, nor $q_{i,j}$ must be an integer for this whole expression to be an integer. In fact, many very simple discrete distributions (like the roll of a die) have non-integer expectations, and thus it is vital that $q_{i,j}$ be continuous in order to compensate for this. We introduce a new variable

$$v_{i,j} = q_{i,j} - \mathbb{E}[D_{1,i-1}]x_{i,j}, \tag{15}$$

and constraint

$$v_{i,j} \in \mathbb{Z}, i \in [1, N], j \in [i+1, N+1]. \tag{16}$$

By declaring an integer variable and forcing this expression to be equal to it, we effectively force this expression to be an integer without placing any further restrictions on it. The drawback of this strategy is that we roughly double the number of integer variables, from just $x_{i,j}$ to $x_{i,j}$ and $v_{i,j}$.

3.3.2 Approximation Method

To approximate $H_{i,j,t}$ it is helpful to first observe that the definition of $H_{i,j,t}$ is

$$H_{i,j,t} = L_{i,t}(q_{i,j} - \mathbb{E}[D_{1,i-1}). \tag{17}$$

By Constraints (15) and (16), it is established that the argument of the loss function in Equation (18) is forced to be an integer. Thus, we only need an accurate approximation of the loss function for all feasible integers of the relevant distribution. The space in between the integers is infeasible, and thus does not need to be modeled accurately. While the argument of

the loss function is guaranteed to be an integer value, the value of the loss function ($H_{i,j,t}$) has no such restrictions and can take on any positive real number.

To approximate Equation (18) I take a similar approach to what is done with the a-priori approximation in Section 3.2.1. However, instead of trying to approximate a continuous function with piecewise linear functions, I am focused only on modeling the discrete points of the loss function exactly, with no error. We thus create a set of linear functions on whose border are the real points of the function being modeled. One way of defining a set of linear functions like this is to create linear functions which intersect subsequent integers. By defining each linear piece as the line which intersects two consecutive feasible points, we can accurately model every point with $n/2$ approximation lines as shown in Figure 1 in which six points are modelled exactly with three linear functions. Normally, optimal points will be found on vertices, but this does not apply here because the vertices fall on non-integer points, which are infeasible due to Constraint (15) and (16). One should note that while the domain in Figure 1 is restricted to integer numbers, the range is not, and is only done as such for the sake of readability.

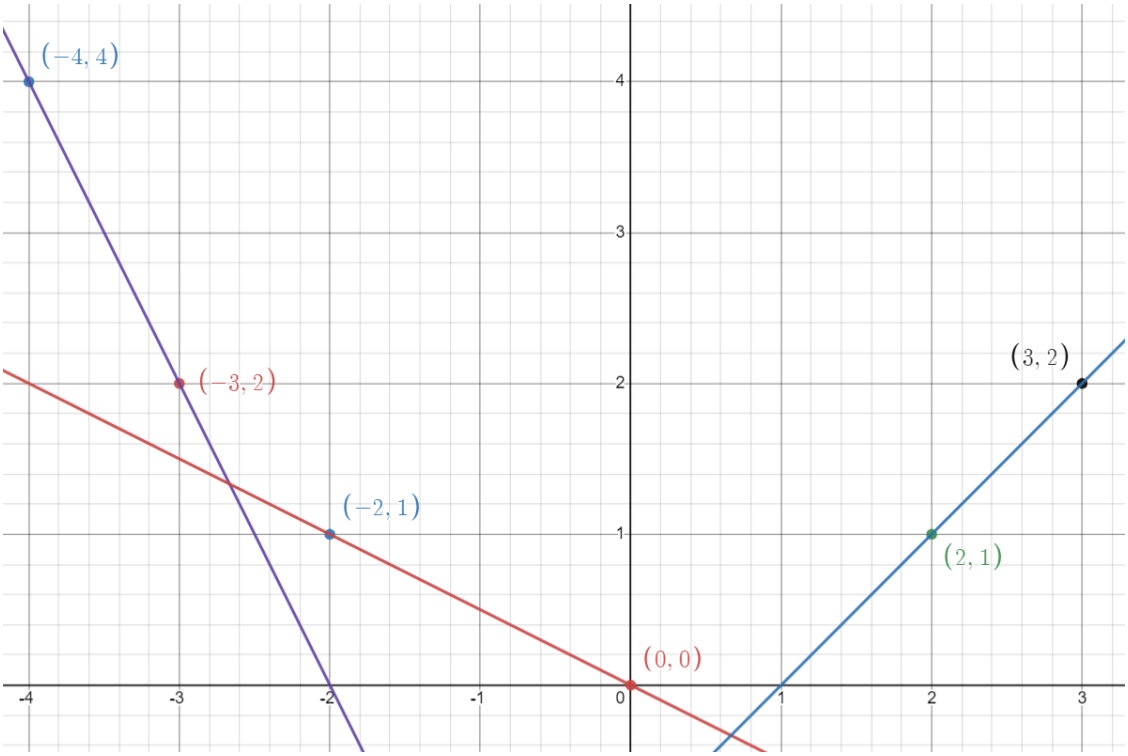

Figure 1: Example of a discrete function such as $L_{i,t}(v_{i,j})$ approximated as discussed in 3.3.2

One obvious result of making the approximation function like this is that we only need the values of (18) at each of its feasible points. We do not need to calculate derivatives, optimal cutoff points or anything else. All we need are the realized values for the loss function, whose equation is

$$L_{i,t}(x) = \sum_{z=x+1}^{\infty} pr(z)(z - x), \tag{18}$$

although with bounded PMF's this series stops when the probability of z is zero for all numbers

above z. Like this we can fully formulate a model with only a (possibly empirical) PMF. Once we have these approximations we can find the optimal solution through the default Cplex solver of Mixed Integer Linear Programming Problems.

3.3.3 Dynamic Cut In Discrete Formulation

Creating a method analogous to the RM-cut method for the discrete formulation (henceforth referred to as DRM-cut) requires a few changes from the RM-cut. Firstly, the initial, cheap, and inaccurate approximation of the loss function must be adapted to the discrete loss functions. Secondly, the constraint(s) which are added at each cut must be adapted for the discrete formulation. A minor constraint is added for ease of computation and much of the code and functions must be rewritten, although in a very intuitive manner which still follows the RM-cut method specified by Tunc et al. (2018).

Instead of Equation (11), I include four constraints on $H_{i,j,t}$ for the cheap and inaccurate initial approximation. These constraints are generated as described in Section 3.3.2, but instead of including every single constraint, I include only those constraints which fall on the 20th, 40th, 60th, and 80th percentile. Thus I include only four constraints instead of hundreds while maintaining reasonable accuracy throughout the range of the function. The other advantage of these constraints is that we know they are valid, and that they never overestimate $H_{i,j,t}$. Because they never overestimate H, we know that in the worst case scenario we can add one constraint for every point in the range of the loss function being estimated and have a valid formulation. It is easy to imagine constraints that whilst more accurate with fewer pieces, result in invalid approximations of H because of occasional overestimation.

Naturally, this leads to the topic of how to include new constraints. Similarly to RM-cut, a feasible point is investigated, and if the accuracy is insufficient, than new constraints are added. At every $H_{i,j,t}$ I add two constraints so that the optimum can shift slightly without needing to add more cuts and thus repeat the process. These constraints are similar to those used in previous parts, and are

$$H_{i,j,t} \geq ax_{i,j} + b(q_{i,j} - \mathbb{E}[D_{1,i-1}x_{i,j}]), i \in [1, N], j \in [i+1, N+1], t \in [i, j-1]. \tag{19}$$

It is important to remember that due to Equation (15), Constraint (19) is equivalent to

$$H_{i,j,t} \geq ax_{i,j} + bv_{i,j}, i \in [1, N], j \in [i+1, N+1], t \in [i, j-1]. \tag{20}$$

The big difference between the way constraints are added in the RM-cut and DRM-cut is the number of constraints being added, as well as the generation of a and b. In accordance with the RM-cut method, constraints are only added if both $x_{i,j}$ is greater than one half and $H_{i,j,t}$ is sufficiently inaccurate, however in the DRM-cut constraints are added two at a time with two a and b variables per $H_{i,j,t}$. As for the a and b variables, they are constructed to form the lines which intersect three selected points along $v_{i,j}$, as shown in Figure 2. These points are the temporary optimal value of $v_{i,j}$ as well as the two points immediately preceding and following.

As is clear in Figure 2, the point being investigated (here two), as well as the ones directly preceding and following are perfectly modeled, thus allowing the optimum to shift slightly

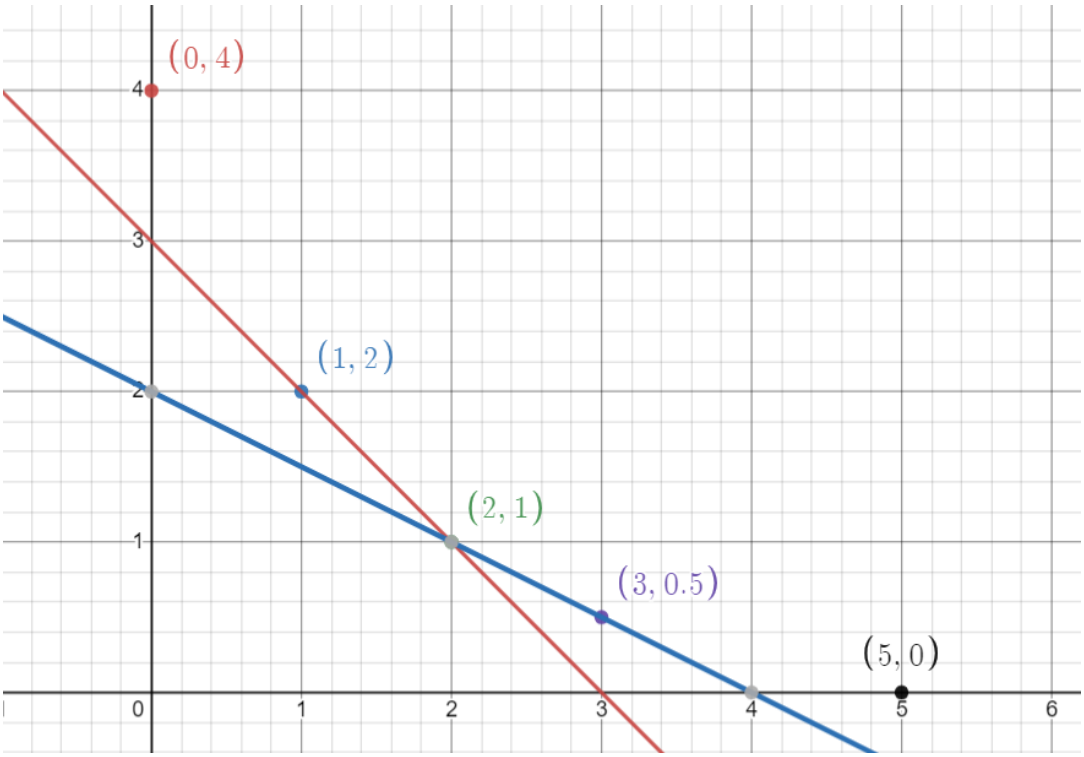

Figure 2: Example of two new constraints placed on $H_{i,j,t}$ through the DRM-cut method. Here the feasible solution investigated is $q_{i,j} - \mathbb{E}[D_{1,i-1}] = 2$

without necessitating more constraints.

3.3.4 Implications On Running Times In the Discrete Formulation

There are two sources for an increase in running time when compared to the continuous formulation. The first is the doubling in the number of discrete variables. This is compounded by the fact that we are adding integer variables, not just binary variables which are generally easier to branch on. Secondly, for every $H_{i,j,t}$ we include $L/2$ piecewise linear functions with L indicating the number of feasible points in the PMF for $D_{i,t}$. A short example follows to show why this is an issue.

For an $H_{i,j,t}$ in which $t - i = 10$ and the demand is distributed with equal probability over $\{0, 1, 2, 3, 4, 5\}$, I would need $5 * 10/2 = 25$ pieces in the piecewise linear function. This problem only gets worse as the distributions get wider, and as we increase the time horizon. The number of piecewise linear functions needed for each individual H grows linearly with the range of the demand distributions, and with the number of periods investigating (due to $t - i$ increasing). However, the number of complete approximations needed grows cubically - $O(n^3)$ - because each of the three indices (i, j, and t) are limited by n, and thus the total number of linear approximations grows by $O(n^4)$ and linearly in L. Given that there are already significant time constraints in the generation of the approximations for the continuous problem (Tunc et al., 2018), this could prove to be a limitation in practice. To our advantage, as the range of possible demands increases, the need to use a discrete formulation decreases because rounding to the nearest whole number represents less of a distortion with large numbers than it does for small

ones. A restock level of about five implies that rounding would be a meaningful distortion, while a restock level of about 1,000 means that rounding would be an insignificant error. Thus, we are interested in observing where exactly this formulation becomes infeasible with respect to the number of periods we are investigating and the size of the possible range of demands.

4 Results

4.1 Description of Code

The code which has been used to obtain these results can be found alongside the paper. The code consists of six main parts. Firstly is the main method where one can generate or input the demand distributions along with the settings and parameters of a problem instance. The other five parts consist of the four solution methods discussed here (a-priori approximation, RM-cut, discrete a-priori approximation, and DRM-cut) as well as one experimental method not discussed in this paper. Each method can be called from the main method with the appropriate parameters. The rest of the code consists of helper methods which calculate various points and functions, although none of these methods need to be accessed directly to reproduce results. If one wishes to replicate specific results, the parameters and distributions are described along with each result presented. The only exception are the running time averages, which have parameters of $p = 5$ and $K = 50$, $K = 500$, or $K = 5,000$ for low, medium, and high demand respectively. K is increased because if demand is increased while K is kept constant, the relative penalty for resupply decreases, and solutions will eventually just resupply at every time period.

4.2 Replication Results

In this subsection I review and compare what I replicate from the paper of Tunc et al. (2018) to ensure both that one can reproduce their paper, and so that we can proceed with the discrete formulation knowing that our foundation is accurate. I firstly compare the objective values achieved with both the a-priori approximation method and the RM-cut method to the results achieved by Tunc et al. (2018). Finally, I compare relative computation times at $n = 100$ to ensure that the cut method is working as intended. This is necessary because the main utility of the RM-Cut method is the reduction in running time, not the objective value achieved, so we must check whether or not my replication achieves this.

4.2.1 Objective Value Comparison

Line Identifier	Their PM	My PM	Their RM-cut	My RM-cut
Backlog_Lumpy_20_1_225.0_1.0_2.0_0.1	1643.17	1643.30	1645.20	1645.53
Backlog_Lumpy_20_1_225.0_1.0_2.0_0.2	1957.47	1957.63	1960.90	1961.07
Backlog_Lumpy_20_1_225.0_1.0_2.0_0.3	2181.44	2181.69	2185.07	2185.32

Table 1: Objective Value Comparison n=20

As one can see in Tables 1, 2, and 3, the objectives achieved by me and by Tunc et al. (2018) are exceedingly similar. Specifically, the differences between the optimal objective values

Line Identifier	Their PM	My PM	Their RM-cut	My RM-cut
Backlog_Lumpy_30_1_900.0_1.0_2.0_0.1	5219.36	5219.59	5224.70	5224.86
Backlog_Lumpy_30_1_900.0_1.0_5.0_0.1	6056.53	6056.70	6066.29	6066.48
Backlog_Lumpy_30_1_900.0_1.0_10.0_0.1	6691.01	6691.14	6707.56	6707.74

Table 2: Objective Value Comparison n=30

Line Identifier	Their PM	My PM	Their RM-cut	My RM-cut
Backlog_Lumpy_40_1_900.0_1.0_10.0_0.3	9182.03	9182.27	9227.25	9227.46
Backlog_Lumpy_40_1_2500.0_1.0_5.0_0.1	13229.44	13229.60	13236.47	13236.53
Backlog_Lumpy_40_1_2500.0_1.0_10.0_0.2	14867.75	14868.21	14894.67	14894.79

Table 3: Objective Value Comparison n=40

determined by the PM method never exceed 0.5, and generally represent a discrepancy of at most one in ten thousand. Moreover, there is little increase in this discrepancy when comparing Tables 1, 2, and 3. For the RM-cut method we see a similar level of accuracy, with differences not exceeding 0.25, and relative differences again around one in ten thousand or less. We can thus say that as far as the results, I faithfully replicate the methods discussed by Tunc et al. (2018). The reason for these small discrepancies is likely rounding, as well as solvers not solving to perfect optimality. Specifically, the PM and RM-cut method require some manual calculation and copying of figures from Rossi et al. (2014). These figures are rounded after the 5th decimal place (out of more than 20) and so this is likely the cause for differences on the scale of one in ten thousand.

4.2.2 Time Cost Comparison

	My PM	My RM-cut	Their PM	Their RM-cut
Setup Time	498.13	484.458	NA	NA
Solving Time	1268.172	129.679	401.98	29.22
Total Time	1766.302	614.137	NA	NA

Table 4: Average time cost for n=100

Table 4 shows the solving time for the PM and RM-cut when $n = 100$. I choose $n = 100$ because it effectively shows behavior as n increases, and because it can be compared to the same experiment done by Tunc et al. (2018) as shown in Table 4. I choose to report setup and solving times separately, because the RM-cut method affects them both differently, and it is unclear which one Tunc et al. (2018) provides, although I believe it to be the solving time. The setup time is very high because the number of H variables scales cubically with the time horizon, and can thus be extremely expensive when n becomes large.

What is important to observe from the table is not the absolute value, but the ratio between the two methods. The absolute value is determined by the speed of the computer, and the ratio is determined by the improvement in the method. What we hope to see is a similar increase in efficiency when comparing the methods as implemented by me and them. Table 4 shows exactly this, with the increase in efficiency achieved by Tunc et al. (2018) being about thirteen times, and the increase in efficiency which I achieve being about ten times. Although this may

not be exact, it is very close given the fact that both are averages of different problem sets run on different computers using different solvers. It is worth noting that the setup time does not change at all between the PM and RM-cut methods. This is due to the initialization of the variables. As n grows the number of $H_{i,j,t}$ variables grows by roughly $O(n^3)$, and thus initializing all of these variables dominates the other setup costs. The cost to calculate and input all of the constraints is negligible in comparison to the costs in initializing all of the H variables.

We can thus conclude that the RM-cut is accurately reproduced, because it achieves appropriate computational gains over the PM method, and reaches correct optimal points (as shown in Section 4.2.1).

4.3 Discrete Results

4.3.1 Correctness of the Approximation Method

To demonstrate the efficacy of this new formulation I present an example problem, the optimal solution I generate, and how this optimal solution changes as a result of different input parameters. The examples treated will have 20 to 40 periods with two types of demand distributions. In the low demand instance, demand is a uniform distribution between 0 and x, with x uniformly distributed between one and ten. The medium and high demand instances are similar, with x distributed between one and 100 and one and 1,000 respectively. We assume that the PMF for each time period is known prior to the first time period. The uniform distribution is chosen because the lack of any thinning towards the extreme values of the distribution means we can not make the problem computationally easier by only approximating the middle 99.9% of the distribution as we could for the sum of many binomial distributions. This makes the uniform a sort of worst case scenario, while the simple nature of these distributions makes it easy to understand and explain in this paper. Ultimately, the exact distribution you use does not matter beyond its range, and whether or not a PMF is available. I run this algorithm with setup costs (K) equal to 50 and 500, shortfall penalty (p) equal to five and 20, as well as with $n = 20$ and $n = 40$. The remainder of this subsection is an inspection of how each solution varies depending on the input, thus showing that each of these variables is correctly incorporated into the model.

Firstly, we inspect the solution for the problem in which we have low demand, $k = 50$, and $p = 5$. In Table 5 we see each resupply period, the order up to quantity enforced at the start of that period, the expected demand in each period, and the expected total cumulative quantity ordered after the resupply of that period. In Table 5 we verify that the order up to quantities are whole numbers and that they make sense with respect to expected demand.

	i	j	Order up to Quantity	Expected Demand Per Interval	$q_{i,j}$
Interval 1	0	6	18	18	18
Interval 2	6	10	31	29	49
Interval 3	10	15	22	25	69
Interval 4	15	20	26	23	98

Table 5: $K = 50$, $p = 5$, $n = 20$, and low demands.

In Table 6 I dramatically increase the fixed cost of a resupply and thus incentivize fewer resupply periods. Accordingly, we see that the optimal solution is one in which there is a single

resupply at time zero that supplies all 20 demands.

	i	j	Order up to Quantity	Expected Demand Per Interval	$q_{i,j}$
Interval 1	0	20	84	95	84

Table 6: $K = 500$, $p = 5$, $n = 20$, and low demands.

Next, we inspect the results of dramatically increasing the costs associated with failing to immediately meet demand. We thus dramatically increase p from 5 to 20 and observe how the optimal reorder schedule adapts. Prior to this transformation in Table 5 we see that the order up to quantity is about equal to the expected demand for that period. This is because in a given period some of the demand can be treated with the next resupply without incurring too much cost. By contrast, in Table 7, we see that the order up to quantity generally exceeds the expected demand by a litt under 50%. This reflects the reality that fulfilling demand late is now much more expensive, and is thus avoided by keeping more product on hand.

	i	j	Order up to Quantity	Expected Demand Per Interval	$q_{i,j}$
Interval 1	0	6	23	18	23
Interval 2	6	9	33	22.5	51
Interval 3	9	13	24	18	64.5
Interval 4	13	16	27	19.5	85.5
Interval 5	16	20	26	17	104

Table 7: $K = 50$, $p = 20$, $n = 20$, and low demands.

Table 8 is the optimal solution when demands are between one and 100 instead of one and ten. As we can see the solution is roughly similar to Table 7 because the effect of p does not change as demands increase, n is the same, while both demands and k are roughly ten times as large. The reduced effects of the integrality constraints account for the difference observed outside of resupplies and expected demand being ten times as large.

	i	j	Order up to Quantity	Expected Demand Per Interval	$q_{i,j}$
Interval 1	0	6	203	207	203
Interval 2	6	10	323	308	530
Interval 3	10	15	245	272.5	7060
Interval 4	15	20	283	252.5	1070.5

Table 8: $K = 500$, $p = 5$, $n = 20$, and medium demands.

Finally, I demonstrate that the model works for instances with different time horizons as shown in Table 9. Notably, even with low demands and $k = 500$, 40 periods is a long enough time horizon that there are multiple replenishment periods.

Thus, it is clear that the order up to quantities are integer values, and that the model can incorporate different PMF's, as well as different values for n, p, and k. Each time the solutions adjust as demanded by the different inputs and remain the same in the appropriate ways. There is no unpredicted or undesirable behavior, supporting the correctness of this formulation for modeling the discrete stochastic lot sizing problem.

	i	j	Order up to Quantity	Expected Demand Per Interval	$q_{i,j}$
Interval 1	0	13	65	58.5	65
Interval 2	13	27	67	63.5	125.5
Interval 3	27	40	78	68	200

Table 9: $K = 500$, $p = 20$, $n = 40$, and low demands.

4.4 Correctness Of The DRM-cut Method

The DRM-cut method must do two things to be considered successfully implemented. Firstly, it must achieve the same optimal solution as the approximation method above, and secondly, it must yield significant computational advantages. The latter is treated in Section 4.5 while the accuracy of optimal solutions is treated here. I compare optimal objective values across multiple time horizons and demand levels to determine whether or not the same optimal solution is reached, similar to what is done in Section 4.2.1. In Table 10 it is clear that in all instances investigated the DRM-cut solution method reaches the correct solution. Running times will be compared in Section 4.5.

Time Horizon	Demand Level	A-priori Approximation	DRM-cut
20	medium	4924.79	4925.12
30	medium	7253.91	7253.71
40	medium	9788.28	9788.11
50	medium	12117.39	12117.12
20	high	49384.15	49384.11
30	high	72714.49	72714.34

Table 10: Optimal Objective Comparison

4.5 Running Times

Time Horizon	Low Demand	Medium Demand	High Demand
20	1.183	11.927	89.884
40	16.009	202.068	NA
50	47.852	723.443	NA
60	214.441	5,268.270	NA
80	398,845	NA	NA
100	1655.437	NA	NA

Table 11: Changing Time Cost For Discrete PM Method

In Table 11 one can see that running time increases as both n and demand increases. As predicted, as demand increases by a factor of ten, the running time increases linearly by a factor of ten as well. This is consistent when demands are increased from single to double digits, and from double digits to triple digits. Additionally, increasing n results in increased time costs consistent with n^4. When n is lower other costs are important as well, and thus the cost from increasing n does not yet make up the vast majority of time costs. This is why we see an increase more in line with n^3 for smaller time horizons before increases on the order of n^4.

Next, we investigate the running times of the DRM-cut method. In Table 12 are the running times for different demand levels and time horizons, directly comparable to Table 11. When applying the DRM-cut method to the discrete formulation, we succeed in reducing computation times by between a factor of ten and fifty. Thus, we succeed in adapting the RM-cut method to the discrete formulation, and as a result can treat instances with longer time horizons and higher demands.

It must be noted that the high demand setting may not be used in practice due to the adequacy of a continuous model. Rounding three or four figure reorder quantities generated from the continuous formulation is a low cost approximation, with approximation error limited to less than one percent. Because the rounding errors become negligible long before the problem becomes computationally infeasible, there is always a viable solution (the discrete formulation with low demand products and rounding with higher demand ones). Thus, the problem of a general stochastic lot sizing formulation for discrete products is effectively treated in this paper.

Time Horizon	Medium Demand	Highest Demand
20	1.147	7.08
40	10.281	204.09
50	34.67	599.59
60	145.58	1,180.83

Table 12: Time Cost for DRM-cut

5 Conclusion and Future Research

In this paper I replicate and extend the solution methods to the general stochastic lot sizing problem introduced by Tunc et al. (2018). In particular, I reproduce the approximation and RM-cut methods successfully, showing that these solutions are accurately described, and easily reproducible. I extend the general formulation to include the case in which discrete products with arbitrary demand distributions are handled. This is solved using both an adaptation of the approximation method, and the dynamic cut method from the continuous formulation.

The benefits introduced by the discrete formulation and solution methods are three-fold. The first is the ability to create resupply and order up to quantities for discrete products. This is important, because there are many goods (a car model at a dealership for example) which are discrete, and demanded in small enough quantities that rounding is a significant distortion. With this formulation, the stochastic lot sizing problem can be solved with respect to these products.

The second benefit is more indirect. In the continuous formulation of the problem, demands are assumed to be normally distributed. If demands are distributed along a non-normal distribution, then it is impossible to model the demands accurately with the general formulation. By contrast, the approximation and DRM-cut method developed in this paper need nothing more than a PMF, and so can handle almost every possible discrete distribution. In particular, it can incorporate empirical distributions, which are often all that is available in real world applications. Combined with the existence of non-normal demand distributions, it is clear that this is a more generalized and practically useful way to model demands.

Thirdly, my a-priori approximation of the loss function for the discrete problem results in perfect accuracy, without any approximation error from linearizing the loss functions. The DRM-cut method is much more efficient, accurate to within an arbitrary epsilon, and can solve problems with high demands and long time horizons. While the DRM-cut method is better in many respects, the a-priori approximation method is easier to implement, making it practically useful for computationally easier instances.

In summary, the static dynamic discrete stochastic lot sizing problem can be solved for almost all demand profiles for which rounding makes a significant difference. Whilst not computationally feasible for all instances, instances with up to 60 time horizons and triple digit demands can be solved easily with DRM-cut. While this is quite expansive, it is possible that a discrete problem with an exceptionally long time horizon cannot be adequately treated. Thus, it would be interesting for future researchers to improve the efficiency of the DRM-cut method so that these instances can be modeled accurately.

One way to do this would be to improve the quality of the constraints added to the model at each iteration of DRM-cut. At each iteration I include two new constraints which perfectly model three points: the point investigated, as well as those preceding and following. With careful implementation one could add a single constraint which perfectly models the investigated point along with either the preceding or following one. Additionally, one could add two constraints which perfectly model the point investigated, two before and one after - or two after and one before. Experiments could be conducted to ascertain which of these methods is most computationally efficient under which circumstances.

Finally, it would be useful to compare the discrete and rounded continuous versions to evaluate at which point the difference between the two becomes insignificant. This is not currently possible, because the continuous version of the stochastic lot sizing problem only supports demands which are distributed normally. Thus, to do a proper comparison, one must first build a formulation for the continuous stochastic lot sizing problem which supports arbitrary demand distributions. The best way to do this would be by modifying the RM-Cut, since it requires only a function and the first derivative. Although the first derivative is difficult to get for an arbitrary function, the empirical derivative will suffice, and the self testing nature of the RM-Cut method ensures an arbitrary level of accuracy. Thus, a future researcher could easily adapt the RM-Cut method so that continuous goods with non-normal demand distributions can be modeled. Besides its utility in testing my own extension, this generalization of demand distributions is a worthy goal in and of itself.

References

Bindewald, V., Dunke, F. & Nickel, S. (2023). Comparison of different approaches to multistage lot sizing with uncertain demand. *International Transactions in Operational Research*.

Gruson, M., Cordeau, J.-F. & Jans, R. (2021). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. *European Journal of Operational Research*, *291*(1), 206-217.

Gutierrez-Alcoba, A., Rossi, R., Martin-Barragan, B. & Embley, T. (2023). The stochastic inventory routing problem on electric roads. *European Journal of Operational Research*. doi: https://doi.org/10.1016/j.ejor.2023.02.024

Huang, K. & Küçükyavuz, S. (2008). On stochastic lot-sizing problems with random lead times. *Operations Research Letters*, *36*(3), 303-308. doi: https://doi.org/10.1016/j.orl.2007.10.009

Ma, X., Rossi, R. & Archibald, T. W. (2022). Approximations for non-stationary stochastic lot-sizing under (s,q)-type policy. *European Journal of Operational Research*, *298*(2), 573-584. doi: https://doi.org/10.1016/j.ejor.2021.06.013

Rossi, R., Tarim, S. A., Prestwich, S. & Hnich, B. (2014). Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, *231*, 489-502. doi: https://doi.org/10.1016/j.amc.2014.01.019

Silver, E. & Meal, H. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Prod. Inventory Manage.*, *2*, 64-74.

Tunc, H. (2021). A mixed integer programming formulation for the stochastic lot sizing problem with controllable processing times. *Computers Operations Research*, *132*, 105302. doi: https://doi.org/10.1016/j.cor.2021.105302

Tunc, H., Kilic, O. A., Tarim, S. A. & Rossi, R. (2018). An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem. *INFORMS Journal on Computing*, *30*(3), 492–506.