Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Technische Bedrijfsvoering

Aggregate Constrained Inventory Systems with Independent Multi-Product Demand

Voorraadbeheersystemen met geaggregeerde beperkingen en een onafhankelijke multi-itemvraag

---

Steven De Schrijver

Proefschrift tot het bekomen van de graad van
Doctor in de Ingenieurswetenschappen:
Bedrijfskundige Systeemtechnieken en Operationeel Onderzoek
Academiejaar 2011-2012

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Technische Bedrijfsvoering

Promotoren: Prof. Dr. Ir. Hendrik Vanmaele
Prof. Dr. El-Houssaine Aghezzaf

Universiteit Gent Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Technische Bedrijfsvoering
Technologiepark 903, B-9050 Zwijnaarde, België

Tel.: +32-9-264.55.02
Fax.: +32-9-264.58.47

Proefschrift tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen:
Bedrijfskundige Systeemtechnieken en Operationeel Onderzoek
Academiejaar 2011-2012

Dankwoord

Circumstance does not make a man, it reveals him to himself.

James Allen 1902

Hierbij wens ik mijn dank te betuigen aan iedereen die bijgedragen heeft aan het verwezenlijken van mijn doctoraat. In eerste instantie denk ik hierbij aan mijn promotoren Prof. dr. ir. Hendrik Vanmaele en Prof. dr. El-Houssaine Aghezzaf, die mij de kans gaven aan dit doctoraat te werken. Graag dank ik ook de leden van de lees- en examencommissie: Prof. dr. ir. Hendrik Van Landeghem, Prof. dr. ir. Pieter Vansteenwegen, Prof. dr. ir. Joris Walraevens, Prof. dr. Trijntje Cornelissens en Prof. dr. ir. Robert Boute.

Daarnaast heb ik ook veel te danken aan MÖBIUS, een veeleisende en kansrijke omgeving. De rijkdom schuilt in de mensen, mijn collega's, en deze is groots en divers. Ik mag er ook geregeld een vleugje genialiteit ervaren. Maar dit alles staat enkel ten dienste van onze klanten en ik ben een blij mens hen te mogen dienen. Het is me vooral een eer mee te mogen bouwen aan succesvolle oplossingen die over de landsgrenzen heen erkend worden. Bovenal koester ik de diepe relaties die hieruit gegroeid zijn, over de projecten en bedrijfsnamen heen, van mens tot mens.

Naast dit boek, is dit doctoraat vooral een fase in mijn leven waarin ik veel gaf, maar hierdoor ook bewust een aantal paden niet heb bewandeld. Het combineert een aantal waarden waarvoor ik wil staan. Met een lach in mijn hart stel ik jullie dan ook graag dit creatieve resultaat voor waar ik als dienstverlener bewust mocht aan werken in een warme omgeving. Ik had geluk toen de kaarten geschud werden, hiervan ben ik me bewust nu ik zie met welke mensen ik mijn leven mag delen en welke kansen mij geboden worden.

Uiteraard wens ik vooral mijn dank te betuigen aan mijn vrouw Lien, mijn ouders en mijn familie voor de onvoorwaardelijke steun. Ook onze kinderen Stan en Noor wil ik bedanken voor hun geduld om het spelen nog even uit te stellen, maar hier ben ik.

Liedekerke, mei 2012
Steven De Schrijver

Table of Contents

List of Figures

List of Tables

List of Contributions

List of Examples

List of Used Symbols

The subscript $_j$ used in this document refers to inventory item j of the multi-product portfolio.

a	Shortage cost per time unit ($moneys/time\text{-}unit$)
$A(t)$	Stockout indicator, $A(t) = 1$ if $IP(t) \leq 0$ and $A(t) = 0$ if $IP(t) > 0$
$\overline{A}$	Stockout frequency, $\overline{A} = 1 - S_3$
$\mathbf{A}$	Stockout indicator stochastic process
A	Stockout indicator equilibrium random variable
$\widetilde{A}_{cP}$	Approximation for compound Poisson stockout frequency
b	Shortage cost per unit and time unit ($moneys/[quantity\text{-}unit * time\text{-}unit]$)
$B(t)$	Backorders at time t
$\overline{B}$	Average backorders
$\mathbf{B}$	Backorders stochastic process
B	Backorders equilibrium random variable
$\widetilde{B}_{cP}$	Approximation for compound Poisson average backorders
c	Purchase cost, variable cost to place an order ($moneys/quantity\text{-}unit$)
c_o	Overage cost for remaining inventory at period end in newsvendor problem ($moneys/quantity\text{-}unit$)
c_u	Underage cost for unsatisfied period demand in newsvendor problem ($moneys/quantity\text{-}unit$)
C	Total average cost ($moneys/time\text{-}unit$)
C_v	Total average variable cost $C_v = C - c\lambda$, ($moneys/time\text{-}unit$)

C_v^a	Marginal cost per item, used in MISSC heuristic
C_v^b	Reset marginal cost per item, used in MISSC heuristic
C_v^c	Normalized reset marginal cost per item, used in MISSC heuristic
C_v^d	Normalized reset marginal cost per item approximation, used in MISSC heuristic
$D(t)$	Cumulative demand through time t, $D(t) = D(0,t)$
$D(t,t+L)$	Demand in the interval (t, t+L)
$\mathbf{D}$	Demand process
D	Demand equilibrium random variable
$\underline{D}$	Uniform mixture of demand over the periodic review interval $[0,T)$
$\mathbf{D}$	Demand process
D_L	Leadtime demand equilibrium random variable
e_{an}	Upper limit for aggregate constraint n
e_{io}	Upper limit for individual item constraint o
$E[X]$	Mean or expectation of X
f	Goal function
f_d	Probability of demand size = d in a compound Poisson process
f_d^k	Probability of total demand size = d by k customers in a compound Poisson process
g	Poisson probability mass function
g_Y	Compound Poisson probability mass function
$g(d\|t)$	Conditional Poisson probability mass function
g_{an}	Aggregate constraint function of aggregate constraint n
g_{so}	Individual item constraint function of individual item constraint o
G	Discrete cumulative Poisson distribution
G^{inv}	Inverse of discrete cumulative Poisson distribution

G^0	Discrete complementary cumulative Poisson distribution
G^{0inv}	Inverse of discrete complementary cumulative Poisson distribution
G^1	Discrete first order loss function of Poisson demand distribution
G^{1inv}	Inverse of discrete first order loss function of Poisson distribution
G^2	Discrete second order loss function of Poisson demand distribution
G^{2inv}	Inverse of discrete second order loss function of Poisson distribution
G_Y	Discrete cumulative compound Poisson distribution
G_Y^{inv}	Inverse of discrete cumulative compound Poisson distribution
G_Y^0	Discrete complementary cumulative compound Poisson distribution
G_Y^{0inv}	Inverse of discrete complementary cumulative compound Poisson distribution
G_Y^1	Discrete first order loss function of compound Poisson demand distribution
G_Y^{1inv}	Inverse of discrete first order loss function of compound Poisson distribution
G_Y^2	Discrete second order loss function of compound Poisson demand distribution
G_Y^{2inv}	Inverse of discrete second order loss function of compound Poisson distribution
h_j	Holding cost, cost to hold one unit in inventory for one unit of time ($moneys/[quantity\text{-}unit * time\text{-}unit]$)
$I(t)$	Inventory at time t
$\overline{I}$	Average inventory
$\mathbf{I}$	Inventory stochastic process
I	Inventory equilibrium random variable
$\widetilde{I}_{cP}$	Approximation for compound Poisson average inventory

$IN(t)$	Net inventory at time t, $IN(t) = I(t) - B(t)$
IN	Net inventory stochastic process
IN	Net inventory equilibrium random variable
$\underline{IN}$	Long run frequency distribution of **IN** in a periodic review policy
$IO(t)$	Inventory on order, the total stock ordered before t but not yet received by t
$IP(t)$	Inventory position at time t, $IP(t) = IN(t) + IO(t)$
IP	Inventory position stochastic process
IP	Inventory position equilibrium random variable
j	Item index, $j = 1, .., J$
J	Number of items
J_{DS}	Number of instances in dataset
k	Ordering cost, fixed cost to place an order (*moneys*)
l	m lower bound value(s) for x_j
L	Demand leadtime *(time-units)*
m	Number of variables per item
m_k	Probability to reach $IP = k$ during an order cycle in an (s, S) policy, this is the discrete renewal function
M	Renewal function of Φ_j
n	Aggregate constraint index
N	Number of aggregate constraints
o	Individual item constraint index
O	Number of individual item constraints
$\overline{OF}$	Average order frequency
$\widetilde{OF}_{cP}$	Compound Poisson average order frequency approximation
p	Shortage cost per unit (*moneys/quantity-unit*)
p_a-p_f	Parameters for error reduction function δ
$\overline{P}$	Average amount new backlogs incurred, $\overline{P} = \lambda(1 - S_2)$

$\widetilde{P}_{cP}$	Approximation for compound Poisson average new backorders
$Pr()$	Probability
Q	Order quantity
r	Reorder point level for (r,Q) policy
s	Reorder point level for (s,S) policy or target stock level -1 in base-stock policy
S	Order up to level of (s,S) policy or target stock level in base-stock policy
SS	Safety stock *(quantity-unit)*
S_1	Replenishment rate service level: probability of no stockout per replenishment cycle
S_2	Fill rate service level: fraction of demand that can be satisfied immediately from stock on hand
$\widetilde{S}_{2cP}$	Compound Poisson fill rate service level approximation
S_3	Ready rate service level: fraction of time with positive net inventory $IN(t)$
$\widetilde{S}_{3cP}$	Compound Poisson ready rate service level approximation
S_{cP}	Service level computed with compound Poisson demand
S_{OL}	Percentage of orderlines delivered out of stock
$\widetilde{S}_{OLcP}$	Approximation for compound Poisson orderline service level
S_{nl}	Service level computed with normal demand
T	Review period: time between two reviews in a periodic order review policy *(time-unit)*
u	Cycle time: time between orders *(time-unit)*
$V[X]$	Variance of X
w_{nj}	Requirement of constrained resource n by item j
x_j	m decision variables for item j
$z_{(r)}$	Standardized value of r, so $z_{(r)} = (r - \nu)/\sigma$ and $z_{(r+Q)} = (r + Q - \nu)/\sigma$

α	Parameter that indicates which KPI's or decision variables are included in the goal function
β	Parameter that indicates which KPI's or decision variables are included in the aggregate constraint function(s)
γ	Parameter that indicates which KPI's or decision variables are included in the individual constraint function(s)
χ	Average quantity ordered by a customer in a compound Poisson distribution $= \sum_{d=0}^{\eta} df_d$
δ	An error reduction function for an inventory KPI
δ_{max}	Maximum error of an error reduction function
ι_k	Stationary IP value in an (s, S) policy, $\iota_k = Pr(IP = k)$
η	Maximum quantity ordered by a customer in a compound Poisson distribution
λ	Demand rate (*quantity-units/time-units*), $\lambda = E[D]$
ν	Mean of the demand during leadtime L, $\nu = E[D_L]$, *(quantity-unit)*
ξ_{oj}	Lagrange variable for individual constraint o on item j
σ^2	Variance of the demand during leadtime L, $\sigma^2 = V[D_L]$
θ_{Xj}	Parameter combining α, β and γ
φ	Standard normal probability density function
Φ	Standard normal distribution function
Φ^0	Standard normal complementary distribution function
Φ^1	Standard normal first order loss function
Φ^{1inv}	Inverse standard normal first order loss function
Φ^2	Standard normal second order loss function
Φ^{2inv}	Inverse standard normal second order loss function
Φ^*	(L+1)-fold convolution of Φ with itself, L is leadtime
ψ^2	Variance of demand process, $\psi^2 = V[D]$
τ	The customer arrival intensity in a compound Poisson process
ω	Cost ratio, $\omega = \frac{p}{p+h}$

ζ_n	Lagrange variable for aggregate constraint n
$Z(x, \zeta, \xi)$	The Lagrangian function

List of Acronyms

B

bS	base-stock

C

CLT	Central limit theorem
cP	compound Poisson demand
CVaR	Conditional value at risk

E

ELSP	Economic lotsize scheduling problem
EOQ	Economic order quantity
erf	Error function

F

FIFO	First in first out

I

IPOPT	Interior-Point Optimizer

K

KPI	Key performance indicator

L

LP	Linear programming

M

MIP	Mixed integer programming

N

nl	normal demand
NP-hard	non-deterministic polynomial-time hard

O

OS	Optimization services

P

PASTA	Poisson Arrivals See Time Average
Pn	Poisson demand

S

SKU	Stock keeping unit

V

VaR	Value at risk

Nederlandse samenvatting –Summary in Dutch–

Voorraadbeheer is alom tegenwoordig, men wordt er mee geconfronteerd in een industriële omgeving, op het werk en ook thuis. Overal vindt men voorraden: de opgeslagen goederen in een magazijn, het papier voor de printer op het dienstenbedrijf of gewoon de bewaarde voeding thuis. Om het doel van deze thesis aan te tonen zullen we een herkenbaar voorbeeld hanteren. Denk aan de opbergkast thuis en alle producten die men er kan terugvinden: koffie, koekjes, flessen, producten voor het onderhoud, tandpasta, aardappelen, ... We kunnen snel 50 producten bedenken die men thuis opslaat. De bevoorrading hiervan vindt plaats door een bezoekje aan de locale supermarkt, maar dit vergt een inspanning en heeft dus ook een kost. Om deze kosten te minimaliseren zouden we kunnen beslissen om in heel grote hoeveelheden aankopen te doen, dit zal zeker het aantal bezoeken aan de winkel reduceren. Maar er zijn ook een aantal nadelen aan verbonden: we moeten de ruimte hebben om deze grote hoeveelheden op te slaan, sommige van de aangekochte goederen hebben een beperkte houdbaarheid, bij grote aankopen moeten we ook onmiddellijk het geld hebben om deze aankopen te betalen, ... Een ander element dat een belangrijke rol speelt bij voorraadbeheer is de geboden dienstverlening: hoe vaak zal het voorkomen dat we een product nodig hebben, maar dat het niet beschikbaar is? We willen een hoge dienstverlening of service level behalen. Indien we bij 100 verzoeken zeker 98 keer correct willen bediend worden, dan spreekt men van een service level van 98%. Deze thesis beschrijft de modellen en algoritmes die nodig zijn om dit soort vragen op te lossen. Zo zullen we voor elk product dat we op stock willen houden aangeven wat de optimale bestelhoeveelheid is en wanneer een nieuw order dient geplaatst te worden.

Klassieke voorraadbeheerformules leggen de focus op problemen met één product. Wij zullen hier de nadruk leggen op problemen met meerdere producten en ook minimaal één geaggregeerde beperking. Er zijn ook twee andere types van modellen met meerdere producten welke binnen dit werk niet behandeld zullen worden: een netwerk van producten (multi-echelon modellen) en gedeelde bevoorradingsketens. Bij dit laatste probleem moet één middel gedeeld worden over verschillende producten terwijl de maximaal beschikbare capaciteit niet mag overschreden worden. De geaggregeerde beperkingen die wij zullen behandelen hebben betrekking op de voorraad

KPI's (key performance indicators), zoals gemiddelde voorraad, gemiddeld aantal achterstallige orders, gemiddelde herbestelfrequentie, fill rate service level, ready rate service level, ...

Voorraadmanagers hebben in praktijk doelstellingen en beperkingen die betrekking hebben op het ganse systeem en deze zijn in ieder geval van toepassing op meerdere producten. Zo kan een bedrijf een strategie hebben om een algemeen fill rate service level te behalen van 97% dit jaar. Dit service level zou onderdeel kunnen zijn van een klantencontract en er kunnen boetes voorzien zijn indien dit niet gehaald wordt. In de praktijk moeten de managers ook oplossingen vinden als ze geconfronteerd worden met beperkte middelen zoals: ruimte, werkmensen en kapitaal. Een magazijn is niet gemakkelijk uitbreidbaar zonder een significante investering. Er is ook een beperking op het werkkapitaal dat in goederen kan geïnvesteerd worden. In de praktijk zien we dat het orderlijn service level (het aantal orderlijnen dat onmiddellijk uit stock kan geleverd worden) heel vaak gebruikt wordt. Dit wordt dan vaak benaderd door het fill rate service level (percentage van het gevraagde volume dat rechtstreeks uit voorraad kan geleverd worden). Het orderlijn service level is heel beperkt behandeld tot heden in wetenschappelijke artikels. Onze analyse toont aan dat een fill rate service level vaak significant hoger is dan het orderlijn service level. Een orderlijn service level kan enkel berekend worden wanneer we gebruik maken van een exacte discrete distributie zoals de Poisson of compound Poisson, maar deze distributies zijn minder makkelijk in het gebruik dan de normale distributie. Daarom hebben we een benadering gecreëerd die gebaseerd is op de normale distributie die de fout op de KPI's reduceert, en voornamelijk ook de fout op het orderlijn service level. Deze fout-reducerende functies zijn van toepassing voor een (r, Q) politiek, hierbij wordt er een order van grootte Q besteld zodra het herbestelpunt r bereikt is. Het is ook van toepassing voor een (s, S) politiek, waar een order geplaatst wordt zodra de voorraad het niveau s bereikt, de ordergrootte is variabel maar leidt er toe dat het niveau zal stijgen tot S. De berekening van de optimale waarden r en Q bij een vraag met een normaal distributie maakt veelvuldig gebruik van de standaard normale eerste en tweede orde verlies-functies. Deze statistische functies hebben geen gesloten-vorm formulering. We creëren een rationele benadering met dubbele precisie voor elk van deze functies alsook voor hun inverse functies. Hiervoor baseren we ons op het Remez algoritme. Deze rationele benaderingsfuncties staan toe om bij optimalisaties veel sneller berekeningen te kunnen uitvoeren.

Een aanpak die tegelijkertijd meerdere producten in rekening brengt heeft twee voordelen: het laat toe om te differentiëren over de verschillende producten binnen bepaalde grenzen, wat kan leiden tot een lagere kost. Anderzijds laat het ook toe geaggregeerde systeembeperkingen in rekening te brengen, zoals beperkte magazijnruimte of kapitaalsinvestering. Aan de hand van enkele echte cases zullen we zien dat de kostendaling ten gevolge van een systeemaanpak in vergelijking met een product per product benadering

kan oplopen van 5% tot 34% en soms zelfs nog hoger.

Het doel van deze thesis is om dit soort van multi-product voorraad problemen op te lossen. We maken hiervoor gebruik van de normale distributie en de ontwikkelde fout-reducerende functies. Op deze manier benaderen we heel goed de exacte discrete vraagmodellen.

Deze thesis is als volgt gestructureerd: in hoofdstuk 1 geven we een inleiding van het probleem, de assumpties en we geven tevens ook al een overzicht van de contributies. Hoofdstuk 2 biedt een literatuur overzicht van de relevante statistische distributies, de beste benaderingsmethodes, voorraadmodellen voor één product alsook de onafhankelijke multi-product modellen. Hoofdstuk 3 gaat nader in op de fouten die gemaakt worden ten gevolge van de normale distributie benaderingen. We introduceren hier ook de fout-reducerende functies. In hoofdstuk 4 creëren we de gesloten-vorm benaderingen voor de standaard normale eerste en tweede order verlies-functies en ook voor hun inverse functies. Hoofdstuk 5 biedt drie methodes om multi-product problemen op te lossen. In hoofdstuk 6 stellen we twee cases voor en tonen ook de voordelen aan van een multi-product aanpak. Tenslotte geven we een overzicht in hoofdstuk 7 van de contributies die gerealiseerd werden binnen deze studie en geven ook mee wat er in toekomstig onderzoek kan behandeld worden.

English summary

Inventory management is omnipresent, whether you are in an industrial environment, at a service company or at home. Everywhere you will find an inventory of items: the stored products in a warehouse, the printer paper in the service company or the food products and other consumables at home. To illustrate the goal of this dissertation we will use an example which we will all easily recognize. Consider the cupboard at home for storing goods: coffee, cans of food, beverages, cleaning products, toothpaste, potatoes... We can imagine having 50 products or more in there. We can replenish the shelves with new goods by going to the store, but this requires effort and as such has a cost. This 'reordering' cost is part of the total cost. We could assume to make very large orders to minimize this ordering cost, and thus minimize the number of visits to the local store. But this has also some disadvantages: we need to have the space to store all this, some of the goods we need may be perishable (fresh fruit or vegetables), we need to have the money to buy all these goods... Another factor that comes in place is the service provided by our cupboard, how often will we go and look for a product and notice that it is no longer in stock? We want to have a high service level, let's assume that we want to find the product we need 98 times out of 100, this could be defined as a service level of 98%. This dissertation describes the models and algorithms necessary to solve this type of questions. In the end it will give for each item we have on the shelves an order quantity and also a reorder point. The reorder point indicates when a new order needs to be placed, the order quantity defines the size of this order.

Classic inventory management equations focus on single item problems. We will work towards a multi-item approach with at least one aggregate constraint. There are also two types of multi-item models that will not be considered in this work: network of items (multi-echelon models) and shared supply chain processes where one resource needs to be shared over several items and its maximum capacity cannot be passed. The aggregate constraints we consider are on the inventory key performance indicators (KPI's) such as average inventory, average back-orders, average order frequency, fill rate service level, ready rate service level...

These are problems that inventory managers encounter daily in practice: they are given system-wide goals and constraints on service level, costs or other resources. As such the company can have for example a strategy to achieve an overall fill rate service level of 97% for this year. This service level

may be part of a service contract which has a financial impact in terms of costly penalties if this pre-set target service level is not achieved. In practice managers also need to find solutions for the limited available capacity of several resources. The warehouse has a limited available space that is not easily surmountable without extra costs. The money available to invest in inventory also has its limitations. In practice we see that the order line service level (number of order lines delivered in full out of stock) is very popular. This is approximated in calculations with a fill rate (percentage of volume directly delivered out of stock). Order line service level has only briefly been dealt with in textbooks or in scientific papers. Our analysis shows that the fill rate is often severely higher than the order line service level. An order line service level can only be calculated when making use of exact discrete distributions such as Poisson or compound Poisson, but these distributions are less easy to use than the normal distribution. So we created an approximation based on the normal distribution that reduces the error on each of the existing KPI's, especially on the order line service level. These error reduction functions are applicable for the (r, Q) policy, where an order of size Q is placed as soon as the reorder level r is reached, and for the (s, S) policy, where an order is placed as soon as the inventory reaches s to increase inventory to the maximum inventory level S. The computation of optimal values for an (r, Q) policy in case of normal demand makes ample use of the standard normal first and second order loss functions. These statistical functions do not have a closed-form expression. We develop closed-form double precision rational approximations for these functions and their inverse functions making use of the Remez algorithm. These rational approximation functions enable us to reduce the computation time.

A multi-item approach has two benefits: it allows, on the one hand, for diversification over the different items within given boundaries leading to lower overall costs and, on the other hand, it makes it possible to integrate system limitations such as limited warehouse space or maximum investment. Through examples and cases we see that the cost reduction due to a system approach compared with a single item approach goes from 5% up to 34% and higher.

The purpose of this dissertation is to solve multi-item inventory problems, making use of the normal demand and the newly created error reduction functions to approximate closely the exact discrete demand models. We also foresee the use of the very popular order line service level.

This dissertation is structured as follows. In Chapter 1 we introduce the problem, its assumptions and we also outline the contributions. Chapter 2 gives a literature review on the considered statistical distributions, best approximations, different single item models and on the independent multi-item models. Chapter 3 analyzes the errors due to normal approximation and we introduce our error reduction functions. In Chapter 4 we work out a closed-form approximation for the standard normal first and second order loss functions and their inverse functions. Chapter 5 offers three methods

for solving multi-item inventory problems. In Chapter 6 we present two real life cases and also demonstrate the significant benefits of a multi-item approach: large cost benefits and the possibility to include warehouse space boundaries. Finally we give an overview in Chapter 7 of the contributions created within this study and some possible future research.

1 Introduction

1.1 Introduction

The issues addressed in this dissertation are concerns and problems encountered in practice by managers who are confronted with system wide goals on service level, costs or other resources. As such the company can have for example a strategy to achieve an overall fill rate service level of 97% for this year. This service level may be part of a service contract which has a financial impact in form of costly penalties if this pre-set target service level is not achieved. In practice managers need to find solutions for the limited available capacity of several resources. The warehouse has a limited available space that is not easily surmountable without extra costs. The money available to invest in inventory also has its boundaries and is sometimes used as a direct key performance indicator (KPI). Throughout this dissertation we will focus on the following set of inventory model KPI's:

- Average inventory: time weighted number of items in stock
- Average backorders: time weighted number of items in backorder
- Average stockout frequency: % of time there is no stock
- Average amount of new backorders: rate at which new backorders are generated
- Order frequency: rate at which replenishment orders are placed
- Fill rate service level: % of items that can be delivered directly out of stock

- Ready rate service level: % of time there is one or more items on stock
- Order line service level: % of order lines that can be delivered immediately and completely out of stock

The limited available workforce capacity can be a reason to limit the number of orders, as each order requires a set of activities: administer, perform quality control, receive and put away the goods. So inventory managers have system wide limitations (space, money or workforce) or goals (service levels or costs), while the majority of classic inventory formulas focus on single items and are unable or inefficient to deal with system wide limitations.

Applying a single item approach to attain these goals is not a best practice, neither is it effective to satisfy the system's constraints. Nevertheless we see it being applied too often within companies, without realizing the loss in efficiency or in money this has as a consequence. An IT system that lacks the support for a system wide approach may however be another significant obstacle. We believe that it is the unawareness of possible system approaches, by a large number of managers, or the assumed insurmountable complexity of these approaches that prevents their widespread use. Within this dissertation we want to work on both these aspects. As a first example to value these system approaches, we want to refer to Sherbrooke (2004) who reports using a system approach on 1 414 spare parts resulting in a 46 % reduction of inventory investment without a decrease in performance. We believe that a better understanding and insight of multi-product inventory problems with aggregate constraints should become common knowledge for the inventory manager, knowing that the first papers on these topics date back to the sixties and seventies. This will certainly help them achieve their system goals and will have a positive impact on the key performance indicators.

An optimal policy surface, see Gardner and Dannenbring (1979), is a practical tool to deduct the optimal link between system cost and system service, while fulfilling the system constraints. An optimal policy surface can be generated for each system based on its specific characteristics. In this dissertation we will also provide an overview of the relevant references and investigate and refine the models and algorithms for the considered policies. The usefulness in practice requires the possibility of handling large data sets and easy implementation, e.g. closed-form expressions or the use of familiar software packages. This is the field where we want to have a significant contribution.

Zipkin (2000) gives a broad overview of multi-product inventory management and its several aspects. An important observation is that multi-product systems and multi-location systems are fundamentally identical. We observe the following three categories of multi-product inventory problems:

1. Independent items with aggregate constraints on average KPI values
2. Network of items
3. Shared supply chain processes (constraint on maximum KPI values)

We will focus in this dissertation on the first category. The first category of independent items describes problems with distinct supply and demand processes and no supply-demand links between the items. Of course when there are no links at all between the items, each item can be treated individually. This is where we introduce one or multiple aggregate constraints on the whole set of items. These constraints are not network or supply chain process related but focus on average KPI values of available resources (average used space, average investment and average workforce needed over time) or system result (service level and cost).

A second multi-product inventory category is a network of items with a supply-demand relationship such as: a series system, an assembly system, a distribution system, a tree system or a general system. Axsäter (2003) offers a good overview of multi-echelon serial and distribution inventory systems in supply chains. Song and Zipkin (2003) give a detailed review on assembly-to-order systems, this is a system with last minute assembly. Desmet et al. (2009) present an approximation model for the retailer replenishment lead-times in a two-echelon distribution system, and discusses its implementation for safety stock optimization in a one-warehouse and N-identical retailers system. Desmet et al. (2010) tackle the problem of optimizing safety stocks in a two-echelon assembly system and present several approximation models for the assembly lead-time under the assumption of normality of the assembly demand and normality of components nominal lead times.

Finally there is a multi-product problem category where the items share the supply chain processes themselves. Two well known problems in this area are the joint-replenishment problem and the economic lot scheduling problem (ELSP). Axsäter (2006) discusses extensively both problems. In case of joint replenishment, a group of items should be replenished jointly as much as possible due to many reasons: joint setup costs, quantity discounts or coordinated transports. The ELSP on the opposite tries to spread the cyclic schedules for a number of items with constant demand and no backordering, due to a finite production rate and a minimized holding and ordering cost.

We focus on the first category: independent items with one or multiple aggregate constraints on the average KPI values. This is especially relevant in the last tier of a supply chain where there is no longer any dependency on later steps in the supply chain. Here we are also confronted with the uncertainty of the customer demand. Within a retail and spare parts environment this model has its direct benefits. In a pure end-consumer (retail) setting we can assume there is no association among the items included in an order.

1.2 Assumptions

We consider inventory problems under the following assumptions:

- Backorders (no lost sales): if a demand cannot be delivered directly out of stock, we assume that the client is prepared to wait until replenishment has occurred. The backorders will then be the first to be delivered. So there is no lost sales.
- Constant replenishment leadtime: we assume a constant replenishment leadtime throughout our models. So we do not consider a variation on the lead time.
- Continuous review: we assume a continuous review of the inventory position. So we do not consider a periodic review.
- Independent items with aggregate constraint(s). As explained in section 1.1 we will only focus on multi-item models with independent items with one or multiple aggregate constraints on average KPI values. This means we will not consider network of items or shared supply chain processes.

1.3 How to read

We have written this dissertation to be a self-standing and self-sustaining document. We used the necessary references, but in order to give the reader the opportunity to fully understand the contributions of this work, we also provided brief introductions on the basics of the considered statistical distributions (section 2.1) and the single item inventory model (sections 2.3.1 - 2.3.7). To highlight our own contributions we have created a specific indication: a frame with a light-gray background:

Contribution xx: Multi-item ...
We analyze some...

The list of these contributions can also be found in the beginning of this dissertation, see page xv. So based on the reader's background in inventory management, he or she can decide to walk through the dissertation from chapter to chapter, or from contribution to contribution, in case of a broad inventory background.

Throughout this dissertation we have added ample examples. These are here to support and further explain the equations and algorithms, but these can be skipped when reading the text. They though can give the necessary clarification when needed. An overview of all the examples is given in the

beginning of this document, see page xviii. An example can be recognized as it always indicated in bold as follows:
Example x.x Constant leadtime demand.

1.4 Contributions

Throughout the dissertation we create new contributions on the following topics:

1. Comprehensive annotated literature review on multi-item inventory models
    - Contribution 1: Multi-item inventory models literature review. We analyze and annotate some recent and relevant references grouped into five categories: deterministic constant leadtime demand, news vendor, base-stock, (r, Q) and (s, S) policy.
2. Order line service level
    - Contribution 2: Order line service level for a base-stock policy: We work out an explicit equation for an order line service level in case of a base-stock compound Poisson demand.
    - Contribution 3: Order line service level for an (r, Q) policy: We work out an explicit equation for an order line service level in case of an (r, Q) policy with compound Poisson demand.
    - Contribution 4: Order line service level for an (s, S) policy: We work out an explicit equation for an order line service level in case of an (s, S) policy with compound Poisson demand.
3. Corrected and simplified (r, Q) KPI equations
    - Contribution 5: Corrected (r, Q) normal demand KPI equations: We work out a corrected set of normal demand (r, Q) policy KPI equations
    - Contribution 6: Conditions for simplified normal demand (r, Q) KPI's: We develop a set of conditions that allow simpler normal demand (r, Q) KPI equations
4. (r, Q) and (s, S) KPI error analysis and error reduction functions
    - Contribution 7: Normal demand (r, Q) & (s, S) KPI approximation error analysis: We provide an analysis of the approximation errors while using a normal demand approximation for a (compound) Poisson demand
    - Contribution 8: KPI error reduction functions for (r, Q) and (s, S): We create significantly improved approximation functions

for the compound Poisson KPI's based upon normal demand equations

5. Closed form approximations for standard normal loss functions
    - Contribution 9: SN1OLF algorithm: a closed-form double precision rational approximation for the standard normal first order loss function.
    - Contribution 10: SN2OLF algorithm: a closed-form double precision rational approximation for the standard normal second order loss function.
    - Contribution 11: ISN1OLF algorithm: a closed-form double precision rational approximation for the inverse standard normal first order loss function.
    - Contribution 12: ISN2OLF algorithm: a closed-form double precision rational approximation for the inverse standard normal second order loss function.
6. Multi-item aggregate constrained inventory solution methods
    - Contribution 13: MIIAC algorithm: We developed a general usable algorithm for multi-item inventory problem with aggregate constraint(s) and optional also individual constraint(s) (MIIAC).
    - Contribution 14: MIISSC: Multi-item heuristic: We developed a heuristic for a specific multi-item inventory problem with one aggregate constraint on a system service level (MIISSC).
    - Contribution 15: MIINLP: non-linear programming: We show that real life, large and complex multi-item inventory problems can be solved in non-linear mathematical programming engines (MIINLP).

1.5 Outline

The remainder of this dissertation is organized as follows. **Chapter 2** gives a literature review. First we give an overview of the statistical functions used throughout this dissertation: Poisson, compound Poisson and normal distribution. Then we provide an overview on the best approximation function techniques and the necessary links to the body of knowledge of numerical methodologies with special attention for the Remez approximation algorithm. Next we revisit the foundation of inventory for the single item model. We consider five single item inventory models: constant demand, base-stock, newsvendor, (r, Q) and (s, S) and define for each of these models the relevant KPI's for Poisson, compound Poisson and normal demand.

Finally we give an overview of the current scientific body of knowledge on independent multi-item inventory models with aggregate constraints.

Chapter 3 analyzes and refines some single item inventory model errors. The relevance of the replenishment rate is discussed and for the popular order line service level we provide exact equations. We analyze the errors in case a Poisson or compound Poisson demand is approximated with normal demand, as is done very often in practice. We finalize this chapter with error reduction functions that allow reducing the error significantly while still making use of the easier normal distribution.

In **Chapter 4** we work out the necessary closed-form statistical approximations for the standard normal first and second order loss function and their inverse functions. We want to have fast approximation functions so we aim for a one pass calculation, allowing an evaluation in milliseconds. The other two requirements we set for these functions is that they are valid in the full range and in full accuracy of double precision numbers.

Chapter 5 builds the bridge from a single to a multi-item inventory model with one or multiple constraints. We develop a general solution algorithm MIIAC. Next we provide a heuristic MIISSC for a more specific situation with one aggregate service constraint and individual service level constraints. We conclude with a non-linear programming model (MIINLP) for a real life complex multi-item situation.

As it is our clear intention to enable solutions for real life problems, we also investigate a set of real life case studies in **Chapter 6**. We first deal with a pharmaceutical wholesaler situation that needs to increase its fill rate service level, but is confronted simultaneously with a warehouse storage limitation. We also present a spare parts case with a high percentage of slow movers, here we focus on cost reduction. In both we apply first a single item approach followed by a multi-item approach.

Finally we give an overview in **Chapter 7** of our contributions and give some possible future research paths.

2

Literature review and discussion

In this chapter we discuss the literature review on:

1. Considered statistical distributions
2. Best approximation functions
3. Single-item inventory models
4. Multi-item inventory models

We analyze two discrete statistic distribution functions (Poisson and compound Poisson), also the continuous normal distribution and we compare them. Next we give an introduction to best approximation functions and focus on the Remez algorithm. We revisit the foundation of inventory for the single item model and consider five single item inventory models.

1. Deterministic constant demand model
2. Newsvendor model
3. Base-stock
4. (r, Q) model
5. (s, S) model

In the last part we give an overview of the current state and latest developments in the field of multi-item inventory models, see also De Schrijver et al. (2011a).

2.1 Independent demand and statistical distributions

The focus of this work is on independent demand, demand without relations between the items. A network of items is an example of dependent demand. In the final tier of the supply chain, we are directly confronted with the customer's unpredictable demand and most often also independent demand. The demand process will be described by the use of a statistical distribution. We will focus on three distributions: Poisson, compound Poisson and the normal distribution. The Poisson distribution represents an arrival pattern of customers. In a compound Poisson process we have a Poisson arrival pattern, but each customer also has an identical order quantity pattern. Although both of these distributions can describe a whole set of situations and real-time demand processes, they are not always easy to use or to calculate. That is why we introduce a third statistical distribution, the normal distribution. This is an approximation distribution and we will approximate the Poisson and the compound Poisson distribution. We will show that the normal distribution can be a very good approximation.

2.1.1 Inventory demand models

The basic demand model is the **Poisson process**, the simplest model of random events, demands occur one at a time. A Poisson process is widely used for several reasons, see Zipkin (2000):

- It is easy and mathematically simple: only one parameter, demand rate λ
- It is fairly accurate: demand really behaves like a Poisson process, often demand comes from many, small, nearly independent sources

A Poisson process often closely approximates real demands, but it is not perfect. Sometimes demand processes behave clearly non-Poisson due to an external world factor that is not influenced by ourselves, these can be represented by **world demand models**. Some influences are: the weather, the economy, competition, customer status, ... Zipkin (2000) showed that these world demand models, where demands also occur one at at a time, still have the same performance formulas as for a Poisson process, only the leadtime demand distribution needs to be changed. These world demand models are not considered in this dissertation.

It is a common assumption in stochastic inventory models that the cumulative demand can be modeled by a non-decreasing stochastic process with stationary and mutually independent increments. Such a process can always be represented as a limit of an appropriate sequence of **compound**

Poisson processes, see Axsäter (2006). If we know the distribution of the demand, but the parameters can change over time, we have a **time-varying demand**. We might apply forecasting systems to estimate these parameters for the future periods. Also here Zipkin (2000) showed that the qualitative results remain valid.

If the variance-to-mean ratio is between 0.9 and 1.1 a Poisson process is acceptable. If the ratio is larger, a compound Poisson process is to be used. The demand size distribution histogram can then be adapted to align the compound Poisson distribution variance-to-mean ratio with the real demand. There also exist goodness-of-fit tests to test the hypothesis of the used model making use of the χ^2-test, see Sherbrooke (2004).

There are several reasons why the **normal distribution** is widely used, see Hadley and Within (1963):

- The normal distribution is easy to work with
- Empirical studies have shown that quite often the normal distribution approximates very well demand distributions encountered in practice

So we will use an inventory model based upon a Poisson or compound Poisson process, this is a first approximation of the real world. Next we make an approximation based upon the normal distribution, this is a second approximation. The first approximation error can be reduced by choosing a good inventory process definition: which distribution and which parameters, based on goodness-of-fit tests. This topic is not handled in this dissertation. The dissertation focuses on minimizing the second error, approximation of the Poisson or compound Poisson inventory process KPI's, into a normal distribution model that can be globally optimized. We prefer an approximation of inventory KPI's based on a normal distribution demand over a direct Poisson distribution approximation for the following reasons:

- The normal distribution is versatile, it can easily handle different variation-to-mean ratios and thus can approximate both the Poisson and compound Poisson distribution
- The Poisson and compound Poisson distributions merge towards the normal distribution (central limit theorem)
- The normal distribution is a continuous distribution, allowing necessary and sufficient conditions for optimality with reasonable complexity
- The stable Remez algorithm can be used to develop highly accurate Chebyshev approximations for continuous functions
- We can specifically focus directly on the inventory KPI's in the regions of importance in practice for each specific replenishment policy

2.1.2 Poisson process and distribution

The Poisson process is one of the most important models as it describes the arrival process of customers. The Poisson process is a viable model when the customers originate from a large and independent population. Mathematical models always are simplifications of reality, one such simplification is the assumption of an exponential distribution as an inter-arrival time. The exponential distribution does not deteriorate with time and the exponential distribution is the only distribution that has this property, this is also called the memoryless property. The Poisson process is a counting process, representing the total number of events (e.g. customers arriving), that occur over a period of time. The process increments are independent and the number of events in a time period has a Poisson distribution. The Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time. We will use the abbreviation 'Pn' to refer to the Poisson distribution. These events occur with a known average rate and independently of the time since the last event, see also Ross (1996) and Ross (2009). For the stochastic process description we refer to a random variable X, most often this will represent the leadtime demand, D_L. The fixed period considered here is the leadtime L and the known demand rate is λ, the average demand during leadtime is $\nu = \lambda L$. In (2.1) we describe $g(k)$ the mass function of the random variable D_L. This Poisson mass function gives the probability of a demand k, a non-negative integer value, during leadtime L if average demand during leadtime is ν. The mean, $E[g(k)]$ (2.2), and variance, $V[g(k)]$ (2.3), are both equal to ν.

$$g(k) = Pr(D_L = k) = \frac{\nu^k e^{-\nu}}{k!} \qquad k \geq 0, k \in \mathbb{Z} \tag{2.1}$$

$$E[D_L] = \sum_{k \geq 0} k g(k) = \nu \tag{2.2}$$

$$\begin{aligned} V[D_L] &= \sigma^2 \\ &= E\left[g\left(k\right)^2\right] - \left(E\left[g\left(k\right)\right]\right)^2 \\ &= E\left[\left(g\left(k\right) - E\left[g\left(k\right)\right]\right)^2\right] \\ &= \nu \end{aligned} \tag{2.3}$$

Figure 2.1 shows three Poisson mass functions with ν equal to respectively 2, 5 and 8. The $g(k)$ function only exists for integer k values, the lines are added just to easily interpret this chart.

In (2.4) and (2.5) we give the discrete cumulative Poisson distribution and its complementary cumulative. (2.7) and (2.8) represent the first order and

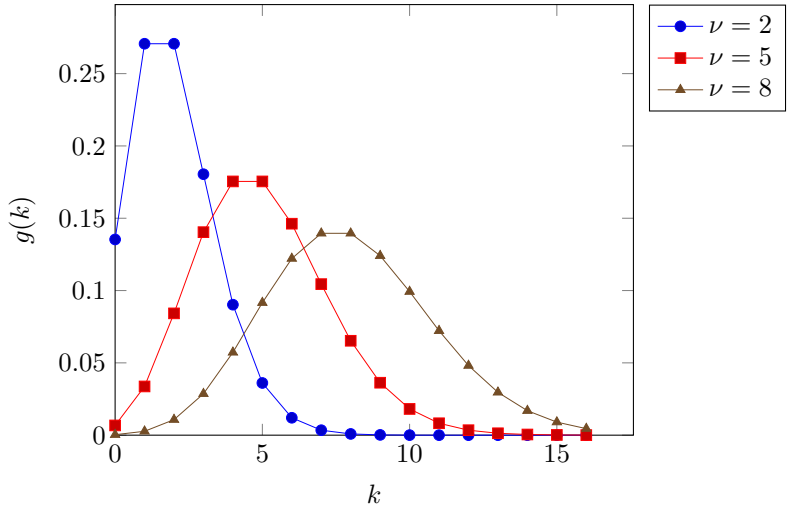

Figure 2.1: Poisson mass function

the second order Poisson loss function, here we use the expression $[D_L - k]^+$, this is explained in (2.6). These loss functions will be often used in the following sections when we describe the inventory performance indicators. In Figure 2.2 we plot the cumulative Poisson distribution for ν equal to respectively 2, 5 and 8.

$$G(k) = Pr(D_L \leq k) = \sum_{y=0}^{k} g\left(y\right) \tag{2.4}$$

$$G^0(k) = Pr(D_L > k) = \sum_{y>k} g\left(y\right) = 1 - G(k) \tag{2.5}$$

$$[D_L - k]^+ = \begin{cases} D_L - k & \text{if } D_L - k \geq 0 \\ 0 & \text{if } D_L - k < 0 \end{cases} \tag{2.6}$$

$$\begin{aligned} G^1(k) &= E\left[[D_L - k]^+\right] \\ &= \sum_{y \geq k} (y-k) g(y) \\ &= \sum_{y \geq k} G^0\left(y\right) \\ &= -(k-\nu) G^0(k) + \nu g(k) \end{aligned} \tag{2.7}$$

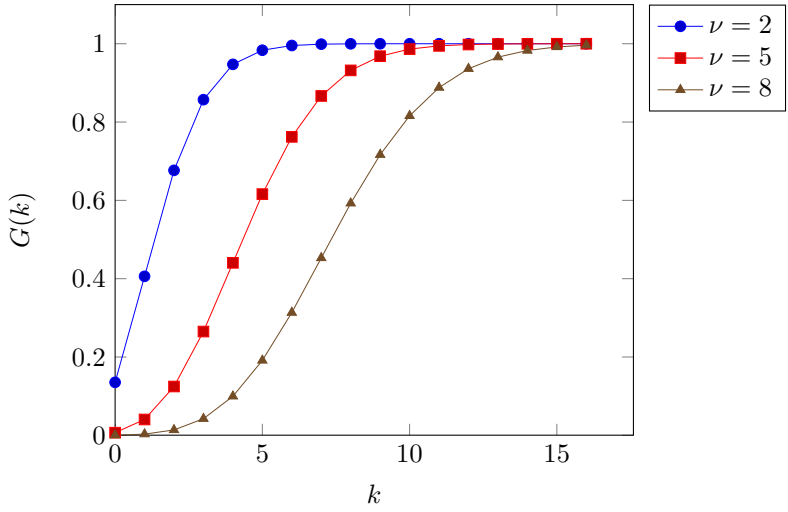

Figure 2.2: Discrete cumulative Poisson distribution, various ν

$$\begin{aligned} G^2(k) &= \frac{1}{2}[[D_L - k]^+[D_L - k - 1]^+] \\ &= \frac{1}{2}\sum_{y \geq k}(y-k)(y-k-1)g(y) \\ &= \sum_{y \geq k}(y-k)G^0(y) \\ &= \sum_{y > k}\left[G^1(y)\right] \\ &= \frac{1}{2}\left\{\left[(k-\nu)^2 + k\right]G^0(k) - \nu(k-\nu)g(k)\right\} \end{aligned} \tag{2.8}$$

As we will also need the inverse of the diverse Poisson functions, we will define them here: the inverse cumulative Poisson distribution (2.9), the inverse complementary cumulative Poisson distribution (2.10), the inverse Poisson first order loss function (2.11) and the inverse Poisson second order loss function (2.12).

$$G^{inv}(p) = k_p, \text{ where } k_p \text{ is smallest k where } G(k) \geq p \tag{2.9}$$

$$G^{0inv}(p_0) = k_{p0}, \text{ where } k_{p0} \text{ is smallest k where } G^0(k) < p_0 \tag{2.10}$$

$$G^{1inv}(p_1) = k_{p1}, \text{ where } k_{p1} \text{ is smallest k where } G^1(k) < p_1 \tag{2.11}$$

$$G^{2inv}(p_2) = k_{p2}, \text{ where } k_{p2} \text{ is smallest k where } G^2(k) < p_2 \tag{2.12}$$

Example 2.1 Poisson demand $\nu = 2$

In this Example 2.1, see Table 2.1 and Figure 2.3, we assume a client demand rate of 20 pieces per year ($\lambda = 20$) and a leadtime of 0.1 years, $L = 0.1$.

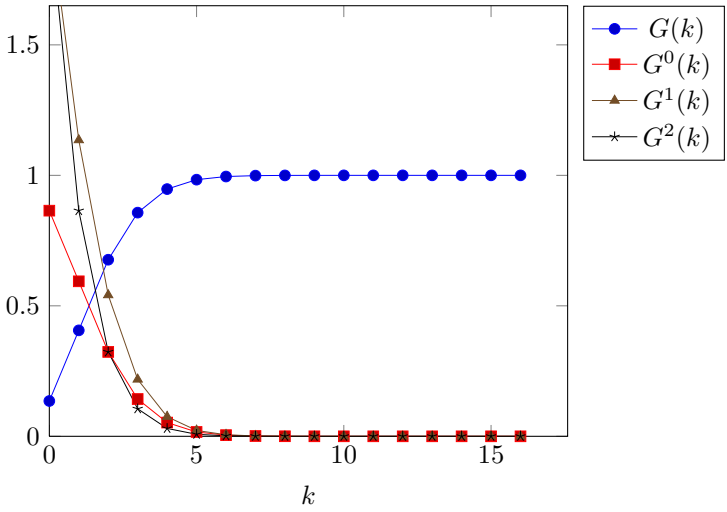

Figure 2.3: Example 2.1: Poisson functions $\nu = 2$

If the client only orders one piece on each visit, it is appropriate to apply a Poisson distribution as demand process. The average demand during leadtime is $\nu = 2$, as is the demand during leadtime variance, $\sigma^2 = 2$. Later on we will use this example in an inventory context, so we provide here some values of the different statistical distributions, see Table 2.1.

Input	Value	Poisson	Value
λ	20	$g(2)$	0.270671
L	0.1	$G(2)$	0.676676
ν	2	$G^0(1)$	0.593994
σ^2	2	$G^1(2)$	0.541341
		$G^2(2)$	0.323324

Table 2.1: Example 2.1: Poisson demand $\nu = 2$

Example 2.2 Poisson demand $\nu = 8$

Example 2.2 in Table 2.2 gives a set of Poisson functions for a larger demand during leadtime, $\nu = 8$. This example has a leadtime demand of on average 8 pieces. In table 2.2 we indicate the demand process parameters and also the different Poisson function values for $k = 9$. The Poisson functions for $\nu = 8$ are also shown in Figure 2.4.

Input	Value	Poisson	Value
λ	80	$g(9)$	0.124077
L	0.1	$G(9)$	0.716624
ν	8	$G^0(9)$	0.283376
σ^2	8	$G^1(9)$	0.709240
		$G^2(9)$	0.920571

Table 2.2: Example 2.2: Poisson demand $\nu = 8$

Comparing Figure 2.3 and 2.4 clearly reveals major differences. We also see that Figure 2.4 already shows the same likeliness and appearance of the normal distribution.

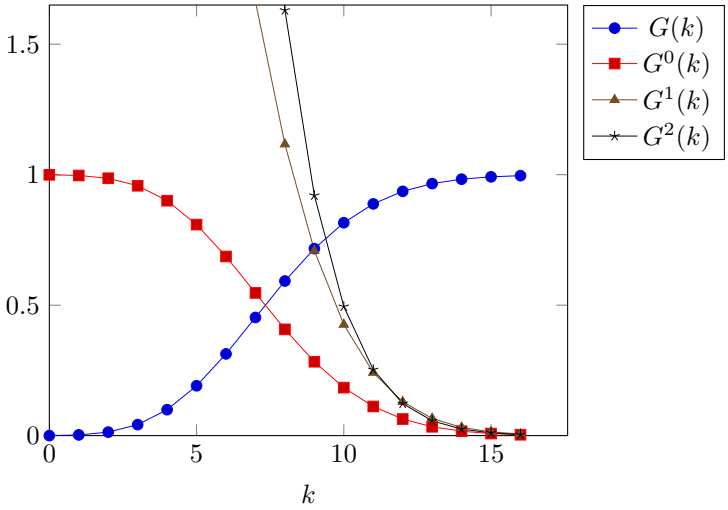

Figure 2.4: Example 2.2: Poisson functions $\nu = 8$

2.1.3 Compound Poisson distribution

In a Poisson process each customer orders exactly one item, so the total demand equals the total number of customers arrived. In a compound Poisson (cP) process, see Ross (1996), a customer arrives according a Poisson process with intensity τ, but he can demand a quantity larger than one. The quantities ordered by the different customers are independent but they all are identically distributed. Let's consider two examples. If buses arrive at a sport event according a Poisson process and the number of fans on each bus are independent and identically distributed, then the number of fans arriving on a bus over a period of time is a compound Poisson process. Another example: if the customers leaving a supermarket is assumed to be according a Poisson process and the amount spent by each customer is supposed to be independent and identically distributed, then the amount spent over a period of time is also a compound Poisson process. So in a time interval t the chance that k customers arrive is expressed by (2.13).

$$g(k) = Pr(D = k) = \frac{(\tau t)^k e^{-\tau t}}{k!} \qquad k \geq 0, k \in \mathbb{Z} \tag{2.13}$$

The size of each customer demand is another stochastic variable f_d, (2.14), that expresses the probability of demand size $= d$ in a compound Poisson process, see Axsäter (2006). Each customer arriving has the same stochastic distribution f_d determining his quantity ordered, but the f_d values of each customer are independent of each other. We do assume that not all demand are multiples of some integer larger than one. As soon as $f_1 > 0$ this condition is already met. If the demand sizes would be multiples of e.g. 5, we could then easily convert to a new unit size of 5, this would then satisfy our condition.

$$f_d = Pr(customer\ demand = d), j = 1, 2, ... \tag{2.14}$$

Example 2.3 Compound Poisson demand pattern

Example 2.3 in Figure 2.5 shows f_d where a customer orders 1, 2, 3 or 4 pieces when he places an order. Most likely a customer will order 1 piece in 40 % of the time.

The maximum quantity ordered by a client is η (2.15). Based upon the format of the stochastic variable f_d we can calculate the average quantity ordered by a customer, χ, see (2.16).

$$\eta = max(d)\ where\ f_d > 0 \tag{2.15}$$

$$\chi = \sum_{d=1}^{\eta} d f_d \tag{2.16}$$

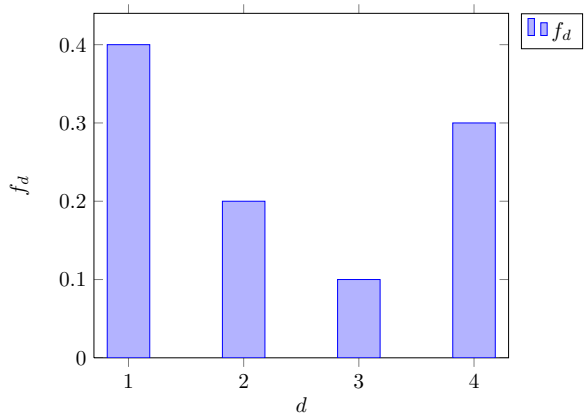

Figure 2.5: Example 2.3: compound Poisson customer demand size

f_d^k is the probability of total demand size $= d$ by k customers in a compound Poisson process, this is expressed recursively by (2.17).

$$f_d^k = \sum_{i=k-1}^{d-1} f_i^{k-1} f_{d-i} \ , \ where \ f_0^0 = 1, f_d^1 = f_d \tag{2.17}$$

So the probability that the total quantity demanded over a period of time t equals d is given by

$$Pr(D = d) = \sum_{k=0}^{\infty} \left[\frac{(\tau t)^k e^{-\tau t}}{k!} f_d^k \right] \tag{2.18}$$

The average demand per unit of time λ (2.19) and the variation of the demand ψ^2 (2.20) are for a compound Poisson demand:

$$\lambda = E[D] = \tau \sum_{d=1}^{\eta} d f_d = \tau \chi \tag{2.19}$$

$$\psi^2 = V[D] = \tau \sum_{d=1}^{\eta} d^2 f_d \tag{2.20}$$

The compound Poisson mass function $g_Y(d)$ for demand during leadtime becomes (2.21).

$$g_Y(d) = Pr(D_L = d) = \sum_{k=0}^{\infty} \left[\frac{(\tau L)^k e^{-\tau L}}{k!} f_d^k \right] \tag{2.21}$$

The average demand during leadtime and the variation of demand leadtime are simply:

$$\nu = E[D_L] = \lambda L = \tau\chi L \tag{2.22}$$

$$\sigma^2 = V[D_L] = \psi^2 L \tag{2.23}$$

Just like for the Poisson distribution we can now define for the compound Poisson function the cumulative distribution (2.24), the complementary cumulative distribution (2.25) and the first (2.26) and second order loss (2.27) functions.

$$G_Y(k) = Pr(D_L \leq d) = \sum_{y=0}^{d} g_Y(y) \tag{2.24}$$

$$G_Y^0(k) = Pr(D_L > d) = \sum_{y>d} g_Y(y) = 1 - G_Y(d) \tag{2.25}$$

$$\begin{aligned} G_Y^1(d) &= \sum_{y\geq d}(y-d)g_Y(y) \\ &= \sum_{y\geq d} G_Y^0(y) \\ &= \nu - \sum_{0\leq y<d} G_Y^0(y) \end{aligned} \tag{2.26}$$

$$\begin{aligned} G_Y^2(d) &= \sum_{y\geq d}(y-d)G_Y^0(y) \\ &= \sum_{y>d} G_Y^1(y) \end{aligned} \tag{2.27}$$

As we will also need the inverse of the diverse compound Poisson functions, we will define them here.

$$G_Y^{inv}(p) = d_p, \text{ where } d_p \text{ is smallest d where } G_Y(d) \geq p \tag{2.28}$$

$$G_Y^{0inv}(p_0) = d_{p0}, \text{ where } d_{p0} \text{ is smallest d where } G_Y^0(d) < p_0 \tag{2.29}$$

$$G_Y^{1inv}(p_1) = d_{p1}, \text{ where } d_{p1} \text{ is smallest d where } G_Y^1(d) < p_1 \tag{2.30}$$

$$G_Y^{2inv}(p_2) = d_{p2}, \text{ where } d_{p2} \text{ is smallest d where } G_Y^2(d) < p_2 \tag{2.31}$$

Example 2.4 Compound Poisson demand $\nu = 2$

Input	Value	Input	Value	Output	Value
λ	20	χ	2.3	$g_Y(2)$	0.098247
L	0.1	f_1	0.4	$G_Y(2)$	0.663166
ν	2	f_2	0.2	$G_Y^0(2)$	0.336834
σ^2	6	f_3	0.1	$G_Y^1(2)$	0.984053
τ	8.695652	f_4	0.3	$G_Y^2(2)$	1.596813
		η	4		

Table 2.3: Example 2.4: Compound Poisson $\nu = 2$

In Table 2.3 we present Example 2.4, this continues on Example 2.1 (Table 2.1). We still have $\lambda = 20, L = 0.1$ and $\nu = 2$, but in the compound Poisson case our customers order a quantity varying between 1 and 4. In Figure 2.6 we show the compound Poisson mass function from Example 2.4. The quantity ordered by the customers, f_d, is the same as previously defined in Figure 2.5. The average ordered quantity per customer is $\chi = 2.3$. We have lowered the arrival pattern to rate $\tau = 8.7$, in order to have the same $\lambda = 20$ and $\nu = 2$ as in Example 1. In Figure 2.6 we show the different compound Poisson functions for Example 2.4. If we compare it with Example 2.1 in Figure 2.3, we can see some distinct differences. G_Y and G_Y^0 cross earlier and also G_Y^1 and G_Y^2 relate differently close to $k = 0$.

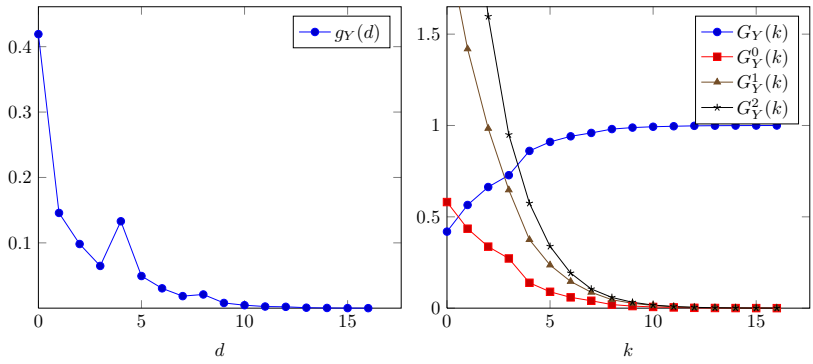

Figure 2.6: Example 2.4: Compound Poisson functions

Example 2.5 Compound Poisson demand $\nu = 8$

The compound Poisson Example 2.5 in Table 2.4 is built in line with the Poisson Example 2.2 in Table 2.2, where we also had a large $\lambda = 80$ and $\nu = 8$. We keep the same f_d customer demand distribution as in Example 2.4. This yields a $\tau = 34.8$ for the customer arrival process. Just like in Example 2.4 we can see the resemblance of the distribution function with a normal distribution.

Input	Value	Input	Value	Compound Poisson	Value
λ	80	χ	2.3	$g_Y(9)$	0.073121
L	0.1	f_1	0.4	$G_Y(9)$	0.658679
ν	8	f_2	0.2	$G^0_Y(9)$	0.341321
σ^2	24	f_3	0.1	$G^1_Y(9)$	1.530409
τ	34.78261	f_4	0.3	$G^2_Y(9)$	4.627767
ψ^2	240	η	4		

Table 2.4: Example 2.5: Compound Poisson $\nu = 8$

Comparing the Example 2.5 (compound Poisson) distributions in Figure 2.7 with the Example 2.2 (Poisson distribution) in Figure 2.4 we also clearly see the impact of the larger variation, $\sigma^2 = 24$ in Example 2.5 (compound Poisson) versus $\sigma^2 = 8$ in Example 2.2 (Poisson). This is because the customers in the compound Poisson example order a quantity between 1 and 4 , while in the Poisson example each customer orders exactly a quantity of one.

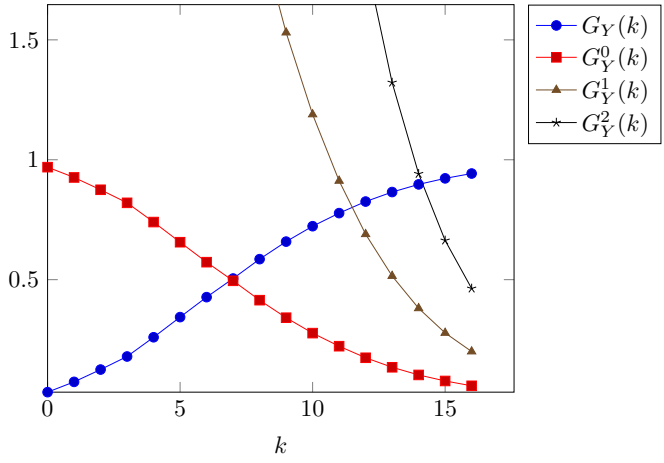

Figure 2.7: Example 2.5: Compound Poisson functions $\nu = 8$

2.1.4 Normal distribution

Instead of using exact performance measures based upon the Poisson demand distribution or compound Poisson distribution we can also apply an approximation distribution. This allows easier calculations. The normal distribution is one of the most important approximation distributions that works very well in a lot of situations, except for some cases with a stochastic leadtime, see section 2.3.8. In probability theory the normal or Gaussian distribution is a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value, see Ross (2009). The normal distribution has an average $E[D_L] = \nu$ and a variance $V[D_L] = \sigma^2$. The probability density function for the normal distribution is (2.32).

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\nu)^2}{2\sigma^2}\right) \tag{2.32}$$

The special case where $\nu = 0$ and $\sigma = 1$ is called the standard normal distribution. Each normal distribution can be translated via a simple transformation to a standard normal distribution. If D_L, the random variable leadtime demand, has a normal distribution with average ν and variance σ^2, then Z, (2.33), has a standard normal distribution.

$$Z = \frac{D_L - \nu}{\sigma} \tag{2.33}$$

Later we will use the following notations (2.34) and (2.35) to express the standardized values of respectively r and $r + Q$.

$$z_{(r)} = \frac{(r-\nu)}{\sigma} \tag{2.34}$$

$$z_{(r+Q)} = \frac{(r+Q-\nu)}{\sigma} \tag{2.35}$$

The standard normal probability density function φ is given by (2.36) and visualized in Figure 2.8.

$$\varphi(z) = \frac{exp\left(-z^2/2\right)}{\sqrt{2\pi}} \tag{2.36}$$

$$\Phi(z) = \int_{-\infty}^{z} \varphi(x)\, dx \tag{2.37}$$

There is no closed-form expression for the normal cumulative distribution function (2.37). It can be expressed in terms of the special error function, *erf*, see (2.38) and (2.39).

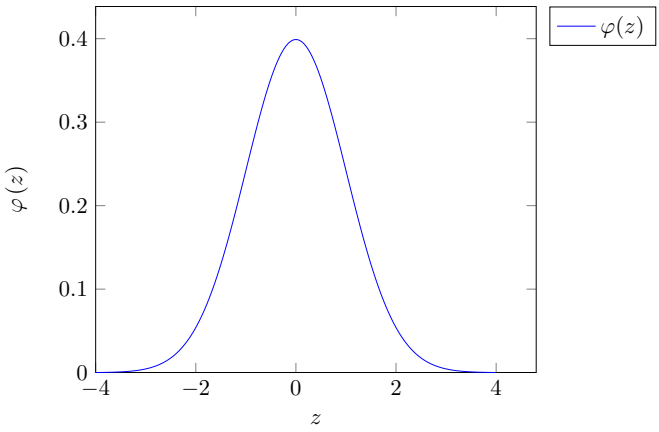

Figure 2.8: Standard normal probability density function

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{2.38}$$

$$\Phi\left(\frac{x-\nu}{\sigma}\right) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\nu}{\sigma\sqrt{2}}\right)\right] \tag{2.39}$$

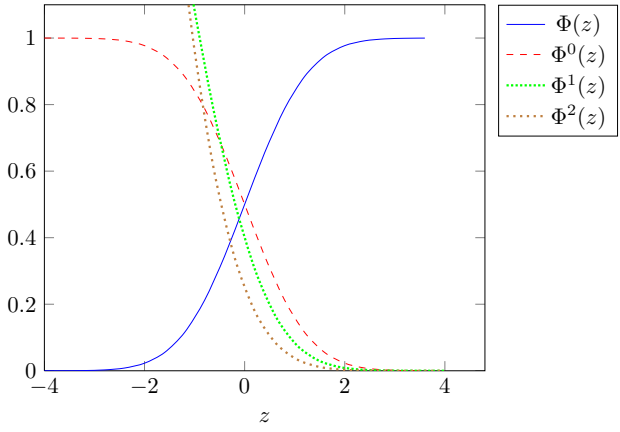

Figure 2.9: Standard normal statistical functions

Within the formulations used later, it is often useful to directly use the standard normal complementary distribution function $\Phi^0(z)$ (2.40). Both $\Phi(z)$ and $\Phi^0(z)$ are visualized in Figure 2.9.

$$\Phi^0(z) = \int_z^\infty \varphi(x)\, dx \tag{2.40}$$

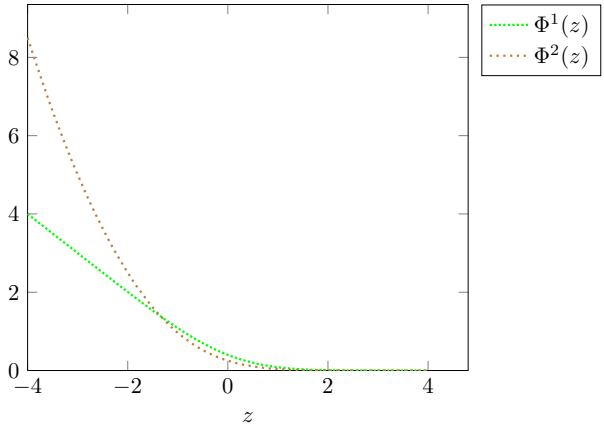

Figure 2.10: Standard normal first and second order loss functions

While optimizing some of the performance indicators included in inventory cost functions we will make ample use of the standard normal first order loss function (2.41) and the standard normal second order loss function (2.42), see Figure 2.9 and 2.10.

$$\Phi^1(z) = \int_z^\infty (x-z)\varphi(x)\,dx \tag{2.41a}$$

$$= \int_z^\infty \Phi^0(x)\,dx \tag{2.41b}$$

$$= \varphi(z) - z\Phi^0(z) \tag{2.41c}$$

$$\Phi^2(z) = \int_z^\infty (x-z)\Phi^0(x)\,dx \tag{2.42a}$$

$$= \int_z^\infty \Phi^1(x)\,dx \tag{2.42b}$$

$$= \frac{1}{2}\left[(z^2+1)\Phi^0(z) - z\varphi(z)\right] \tag{2.42c}$$

Within optimizations we will also need the inverse of these normal distribution functions: the inverse standard complementary cumulative distribution Φ^{0inv} (2.43), the inverse standard normal first order loss function Φ^{1inv} (2.44) and the inverse standard normal second order loss function Φ^{2inv} (2.45).

$$\Phi^{0inv}(p_0) = z_{p0}, \text{ where } p_0 = \Phi^0(z_{p0}) \tag{2.43}$$

$$\Phi^{1inv}(p_1) = z_{p1}, \text{ where } p_1 = \Phi^1(z_{p1}) \tag{2.44}$$

$$\Phi^{2inv}(p_2) = z_{p2}, \text{ where } p_2 = \Phi^2(z_{p2}) \tag{2.45}$$

2.1.4.1 Comparison Poisson and normal distribution

As it is our goal to work out easy to use inventory functions, we want to know where we can use the normal distribution as approximation. We can make use of the central limit theorem to prove that Poisson converges towards a normal distribution: the distribution of the sum of a large number of independent, identically distributed variables has an approximately normal distribution, whatever the underlying distribution we are looking at, so also for the Poisson distribution. Let's assume we have a Poisson distribution Y with mean ν, then Y can also be expressed by (2.46), where X_i are Poisson distributions with mean = 1. So the central limit theorem holds for a Poisson distribution.

$$Y = \sum_{i=1}^{\nu} X_i \tag{2.46}$$

We will discuss three Poisson approximations based on the normal distribution: the direct normal distribution $\phi(z_{(k)})$, a corrected normal distribution $\phi(z_{(k+0.5)})$ and finally the Wilson-Hilferty approximation. In Figure 2.11 we give 4 comparisons between the cumulative distribution functions for Poisson $G(k)$ and the normal $\Phi(z_{(k)})$. We compare 4 situations: $\nu = 2$, $\nu = 8$, $\nu = 20$ and $\nu = 40$. We clearly see the increase in resemblance between both curves as ν increases. The Poisson distribution is a discrete distribution and only exists at the discrete values. The normal distribution is a continuous distribution and also exists for negative values, while $G(k)$ only exists for $k \geq 0, k \in \mathbb{Z}$. In figure 2.11A and 2.11B we also plot $\Phi(z_{(k+0.5)})$, that holds a 'continuity correction' to improve the approximation. In figure 2.11C and 2.11D we did not plot the $\Phi(z_{(k+0.5)})$ correction as $\Phi(z_{(k)})$ is already a good approximation, due to the larger ν and the central limit theorem. Peizer and Pratt (1968) discuss various types of normal-based approximations to the Poisson distribution in more detail. Lesch and Jeske (2009) analyze the Wilson-Hilferty normal based Poisson approximation, which is easy to calculate and also accurate, it is based on the χ^2-approximation from Wilson and Hilferty (1931), see (2.47). (2.47), (2.7) and (2.8) enables us to calculate the Poisson distribution and its loss functions. Unfortunately the Wilson-Hilferty approximation would lead to optimization models that have a very complex formulation and that are not tractable.

$$G(k) \approx 1 - \Phi((c - \nu)/\sigma) \tag{2.47a}$$

$$c{=}(\nu/(1+k))^{1/3}),\ \nu{=}1 - 1/(9(1+k)) \text{ and } \sigma{=}1/(3\sqrt{(1+k)} \tag{2.47b}$$

When approximating $G^1(k)$ with $\sigma\Phi^1(z_{(k)})$ there is no need for a +0.5 correction, see Figure 2.12.

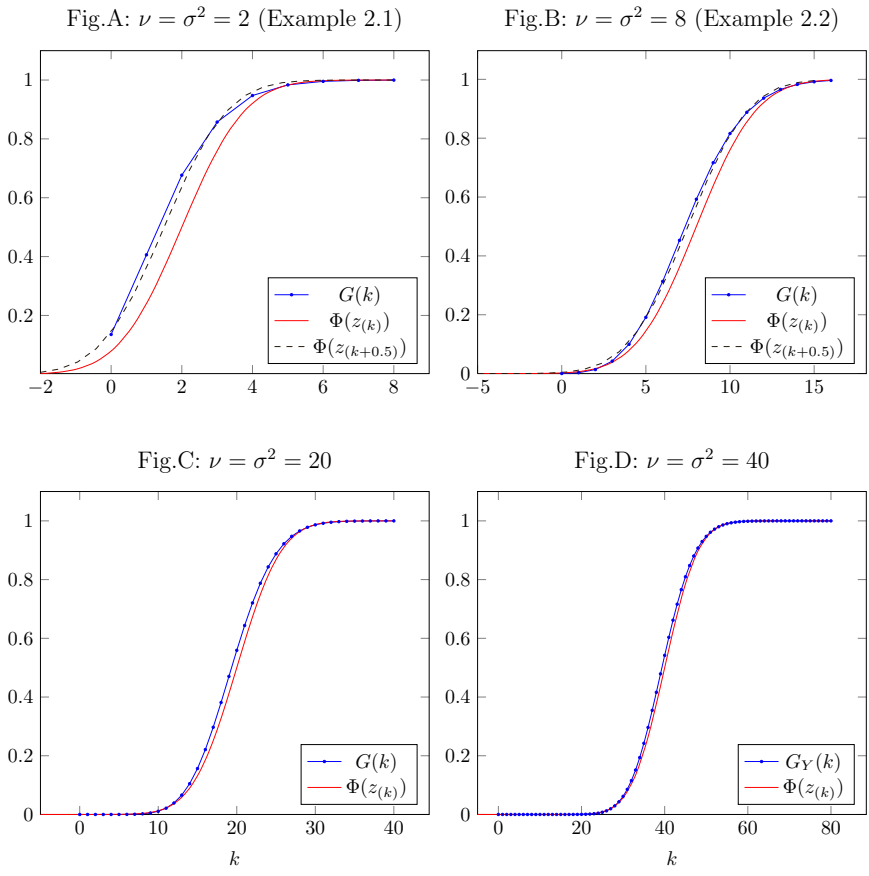

Figure 2.11: Comparison Poisson and Normal, varying ν and σ^2

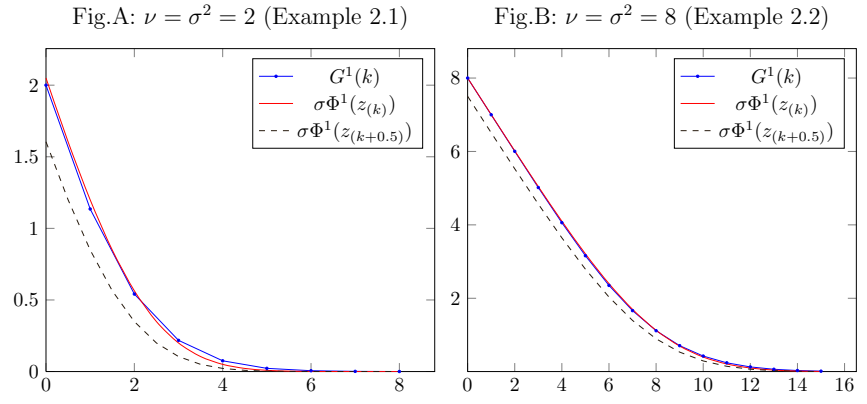

Figure 2.12: Comparison $G^1(k)$ and $\Phi^1(z_{(k)})$, varying ν and σ^2

2.1.4.2 Comparison compound Poisson and normal distribution

We also make a normal approximation for the compound Poisson distribution. Using a customer demand pattern f_d as defined in Figure 2.5, $\sigma^2/\nu = 3$. We now compare 4 different situations $\nu = 2$, $\nu = 8$, $\nu = 20$ and $\nu = 40$ with customer demand pattern f_d. As with the Poisson case we see an increasing resemblance between G_Y and Φ as ν increases due to the central limit theorem. When σ/ν increases, $\Phi(x = 0)$ also increases, which means that Φ for negative values becomes more significant. This is clearly visible if we compare the first two charts of Figure 2.11 with the first two charts of Figure 2.13. So larger σ/ν leads to larger deviation between the compound Poisson and its normal approximation. When approximating $G_Y(k)$ with $\Phi(z_{(k+0.5)})$, applying a continuity correction, see Figure 2.5A and 2.5B, approves the approximation, but less than with the Poisson case.

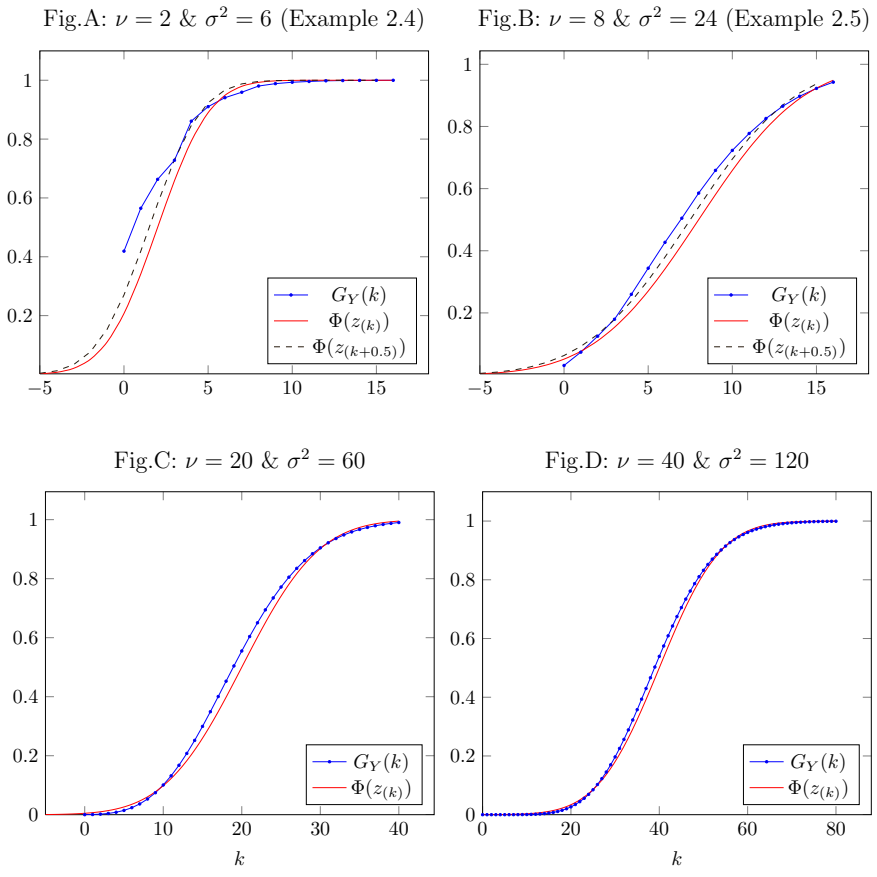

Figure 2.13: Comparison compound Poisson and Normal, varying ν and σ^2

2.2 Best approximation functions

To solve inventory optimization questions we will be required to have approximations of several statistical distribution functions. As no closed-form expressions exist for these statistical functions we will create a set of highly accurate approximations of normal statistical loss functions and their inverse functions. So we will create a simpler function for the standard normal first and second order loss function. As we want to evaluate these loss functions very often at several points, there is a direct gain in performance if this evaluation can be performed on the simpler and faster approximation functions. Polynomial expansions such as Taylor and Maclaurin series represent a function as an infinite sum of terms based upon the derivatives of the specific function we want to approximate. While they are often used in theory, for practical work they can be less useful. In practice an algorithm is required that is fast and accurate, as it may be required several thousands of times to solve a single specific question. We will use a minimax approximation, it minimizes the maximum error (absolute or relative) between the approximation and the considered function.

Burden and Faires (2005) give a survey on approximation theory. Two types of approximations are considered: discrete and continuous. Discrete approximations are applied when there is only a limited set of data points for the function we want to approximate. A least squares technique can be applied here to form a linear, polynomial or trigonometric polynomial function. For the latter an efficient method is the Fourier transform.

Continuous approximations can be used if the approximated function is fully known. Then we can minimize the integral of the errors instead of the sum. An efficient continuous least squares polynomial approximation leads to orthonormal sets of polynomials, such as Legendre and Chebyshev polynomials. Chebyshev polynomials are a sequence of orthogonal polynomials which can be defined recursively. Chebyshev polynomials are important in approximation theory because the roots of the Chebyshev polynomials of the first kind, which are also called Chebyshev nodes, are used as nodes in polynomial interpolation. Rational approximations, a ratio of two polynomials, allow a more uniform method of approximation than polynomials. An example is the Padé approximation and a further extension is the Chebyshev rational approximation. Evgeny Remez developed general computational methods of Chebyshev approximation for polynomials. Later he developed a similar algorithm that allowed rational approximations of continuous functions defined on an interval with a prescribed degree of accuracy, see Remez (1934a), Remez (1934b) and Remez (1934c).

Fraser and Hart (1962) already indicated that for computer programs that require as nearly as possible approximations the Chebyshev approxi-

mations are better than others. The Remez algorithm has been developed into a stable method for finding the best polynomial approximations. Barrar and Loeb (1970) gave proof of the convergence for the classic Remez algorithm when applied in certain non-linear approximating families. Cody (1970) provided a survey with methods for generating rational or polynomial approximations to continuous functions. Dunham (1975) investigated the convergence of the Fraser-Hart variant of the Remez algorithm, which is used to determine the best rational Chebyshev approximation to a continuous function. Litinov (1993) describes several construction methods for rational approximations to functions of one real variable and he focuses on error auto correction, so that significant errors in the coefficients do not affect the accuracy of the approximation. Elbarbary et al. (2003) construct a restrictive type of Chebyshev rational approximation to approximate the exponential function, this approximation yields more accurate results and exact values at selected points.

2.2.1 Remez method

We use the Boost (2011) C++ algorithm to develop approximations. This C++ code is based upon the Remez algorithm. We approximate a function $f(x)$ by way of a function $R(x)$, where $R(x)$ may be either a polynomial $P(x)$ or a ratio of two polynomials $P(x)/Q(x)$ (a rational function). We want to find the "best" rational approximation, where "best" is defined to be the approximation that has the least deviation from $f(x)$. We can measure the deviation by way of an error function which is expressed in terms of absolute error ϵ_{abs} (2.48), but we can equally use relative error ϵ_{rel} (2.49).

$$\epsilon_{abs}(x) = f(x) - R(x) \tag{2.48}$$

$$\epsilon_{rel} = \frac{f(x) - R(x)}{|f(x)|} \tag{2.49}$$

The Remez algorithm is briefly explained in more detail in Appendix A.1.

Example 2.6 Rational minimax Remez approximation

We give a small example to show the benefits of a rational minimax Remez approximation compared with a more classic equidistant polynomial approximation. We consider the complementary normal cumulative distribution Φ^0 in the range $z \in [0, 3]$ and we will make four approximations:

1. $P_{equidist}$: an equidistant polynomial approximation of order 4
2. P_{remez}: a minimax polynomial Remez approximation of order = 4

3. R_{remez}: a minimax rational Remez approximation of order = 2/2
4. R_{remez}: a minimax rational Remez approximation of order = 2/2 based upon a dominant function

For each of these three functions we have five parameters, see (2.50), (2.51) and (2.52).

$$P_{equidist} = -0.0071z^4 + 0.028z^3 + 0.0672z^2 - 0.4305z + 0.5026 \tag{2.50}$$

$$P_{remez} = -0.0071z^4 + 0.028z^3 + 0.0641z^2 - 0.4272z + 0.5018 \tag{2.51}$$

$$R_{remez} = \frac{0.0594z^2 - 0.3417z + 0.4991}{0.2931z^2 + 0.0781z + 1} \tag{2.52}$$

In Figure 2.14 we can see the absolute error ϵ_{abs} for each of the three approximation functions. For $P_{equidist}$ we can see a limited Runge effect near the borders of the considered range, especially near $z = 2$ we see a steep increase of the absolute error to a value $> 4e-3$. This Runge phenomenon is an oscillation at the interval borders in case of high degree polynomial interpolation, see Ralston and Rabinowitz (2001). The Remez polynomial approximation does not use equidistant nodes and here we can establish an absolute error $\epsilon_{abs} < 2e-3$. The third approximation is a Remez rational function with $\epsilon_{abs} < 8e-4$.

To explain the strength of dominant functions in the process to find the best Remez approximation, we will make use of the dominant function $exp(-0.5z^2)$, see (2.53).

$$R_{remez2} = exp(-0.5z^2)\frac{0.00209z^2 - 0.09749z + 0.49999}{0.29995z^2 + 0.99215z + 1} \tag{2.53}$$

Using this dominant function, which we can calculate exact, will yield far better results. Although we are still using five parameters, as with the previous approximations, the error has further decreased: $\epsilon_{abs} < 9.6e-6$ and as such is nearly 100 times smaller than R_{remez} and 500 times smaller than $P_{equidist}$. Making use of these aspects in a creative way we will be able to develop novel and highly accurate approximations for the standard normal first and second order loss function and their inverse in the full range of double precision values.

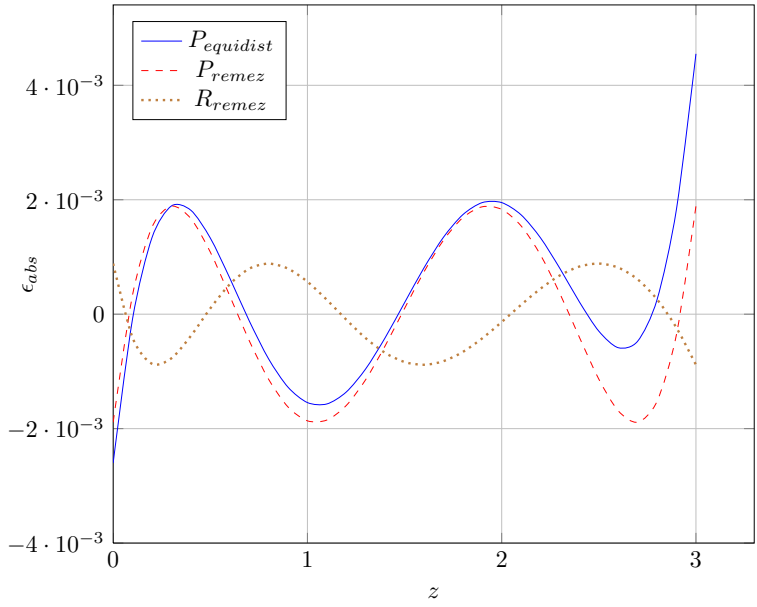

Figure 2.14: Example 2.6 Function approximation absolute error ϵ_{abs}

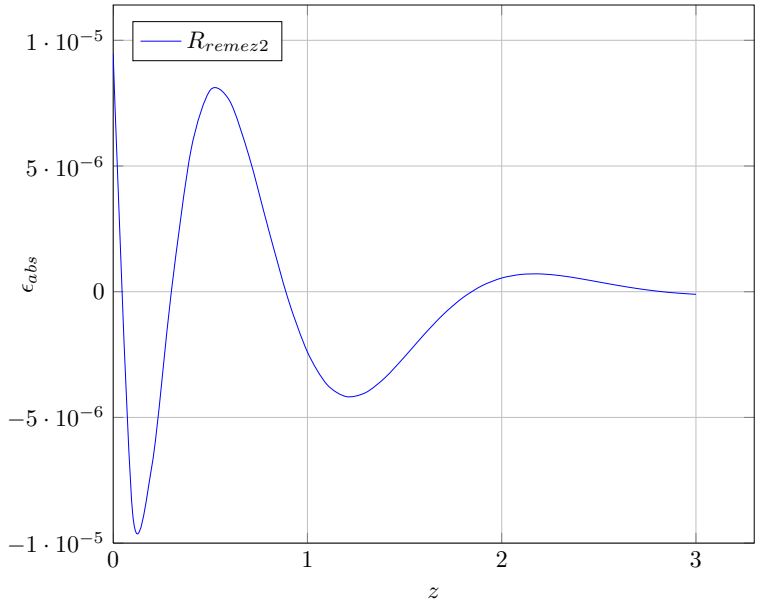

Figure 2.15: Example 2.6 Function approximation absolute error ϵ_{abs}

2.3 Single-item inventory models literature review

We review the inventory foundations for single item inventory models. We use the constant demand model for making the elementary definitions. The base-stock model is dealt with in great details as it is the foundation of two other stochastic inventory models we will consider. (r, Q) and (s, S) can be seen as an average and weighted average of the base-stock model. Illustrative examples and enlightening figures are used throughout the chapter. We consider several instances of this single-item inventory problem:

- Constant leadtime demand
- Base-stock: an (r,Q) policy with $Q = 1$, this is relevant when ordering costs are negligible compared with other costs
- Newsvendor: a single period model with a stochastic demand and penalty costs for ordering too much or too little
- (r,Q) policy: an order of size Q is placed as soon as the inventory position falls to or below the reorder point r
- (s,S) policy: an order is placed to reach the stock maximum level S as soon as stock falls to or below reorder point s

Each of these models and much more are extensively described in Hadley and Within (1963), Silver et al. (1998), Zipkin (2000) and Axsäter (2006). Here we will summarize the relevant and essential elements. This single-item analysis enables us to make the step towards the multi-item models.

2.3.1 Constant leadtime demand

In the case with constant demand during leadtime the following assumptions are made: a constant demand rate λ *(quantity-units / time-units)* and a constant demand leadtime L *(time-units)*. Under these conditions there are two questions that need to be answered: When to order? How much to order? In Figure 2.16 we visualize this perfect world inventory example. We visualize two lines, $I(t)$ the inventory at time t and also $IP(t)$, the inventory position at time t. $IO(t)$ is the inventory on order, which is the amount ordered but not yet arrived at time t. The relation between $IP(t)$, $I(t)$ and $IO(t)$ is given by (2.54) if there are no backorders or lost sales. As we have a constant demand leadtime L and a constant client demand rate λ, we can calculate the demand during leadtime ν and place an order as soon as the inventory $I(t)$ reaches the reorder point r. In this simple case $r = \nu$, see (2.55). As such the order will arrive at the moment when the inventory $I(t)$ would become zero. Using this relation we can express the inventory at time

$t+L$ based upon the current inventory position $IP(t)$ and the reorder point r, see (2.56). Setting the reorder point like this will prevent backorders or lost sales.

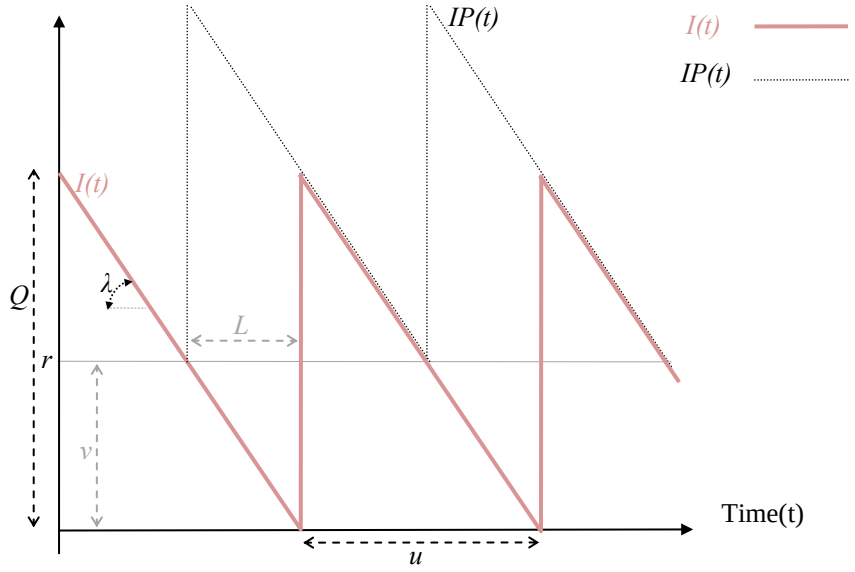

Figure 2.16: Inventory functions: I and IP

$$IP(t) = I(t) + IO(t) \tag{2.54}$$

$$r = \nu = \lambda L \tag{2.55}$$

$$I(t+L) = I(t) + IO(t) - \nu = IP(t) - \nu \tag{2.56}$$

So an order needs to be placed as soon as the inventory position function $IP(t)$ reaches the reorder point $r = \nu$. In order to determine the optimal order quantity, Q^*, we have to know the related costs. For this model we have three relevant cost components: reorder cost, unit cost and holding cost, each having its own parameter:

- k = fixed cost to place an order ($moneys$)
- c = the purchase cost ($moneys/quantity - unit$)
- h = holding cost, cost to hold one unit in inventory for one unit of time ($moneys/[quantity - unit * time - unit]$)

In order to easily express the cost we first define two performance indicators: the average inventory ($\overline{I}$), see (2.57), and the average order frequency ($\overline{OF}$), see (2.58). Both of these performance criteria will depend on the order quantity Q we will choose. The time between two orders, the cycle time u, is also directly related to the order quantity Q (2.59).

$$\overline{I} = Q/2 \tag{2.57}$$

$$\overline{OF} = \lambda/Q \tag{2.58}$$

$$u = \frac{Q}{\lambda} \tag{2.59}$$

The total average cost of this system, C, depends on Q and can be expressed by (2.60). Very often we are only interested in the total variable cost C_v, see (2.61), this part of the cost we can influence.

The optimization of this cost function was firstly discovered and published by Harris (1913), later reprinted as Harris (1990), but most commonly known as the Wilson formula, see Wilson (1934). This total average cost can be minimized by the EOQ (economic order quantity) Q^*, see (2.62). This yields the optimal average total cost C^*, see (2.63).

$$C(Q) = k\overline{OF} + c\lambda + h\overline{I} \tag{2.60}$$

$$C_v(Q) = k\overline{OF} + h\overline{I} \tag{2.61}$$

$$EOQ = Q^* = \sqrt{\frac{2k\lambda}{h}} \tag{2.62}$$

$$C^* = c\lambda + \sqrt{2k\lambda h} \tag{2.63}$$

Example 2.7 Constant leadtime demand

Example 2.7 in Table 2.5 describes a model with a demand rate $\lambda = 20$, a holding cost $h = 32$ and an ordering cost $k = 80$. To be complete we also add the leadtime $L = 0.1$ and the demand during leadtime $\nu = 2$, but these do not impact the optimal order quantity.

Input	Value	KPI	Value
λ	20	Q^*	10
h	32	C_v^*	320
k	80	$\overline{I}$	5
L	0.1	$\overline{OF}$	0.5
ν	2		

Table 2.5: Example 2.7: Constant leadtime demand

In Figure 2.17 we give a visual overview of the two variable cost components $h\overline{I}$ and $k\overline{OF}$ and also of the total variable cost C_v. The minimum value of the variable cost indicates the economic order quantity $Q^* = 10$.

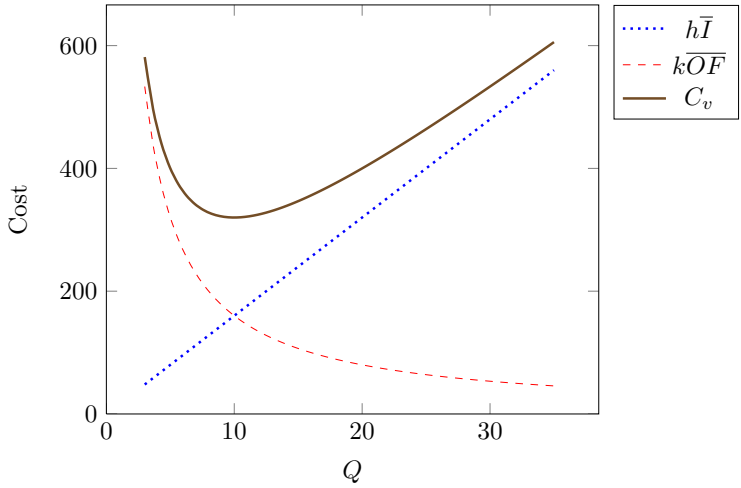

Figure 2.17: Example 2.7: Economic order quantity (EOQ)

2.3.2 Service levels and backorders

In the previous policy we prevented backorders by ordering as soon as $I(t)$ reached the reorder level $r = \nu$, to prevent stock breaks. In certain occasions it can be interesting to allow backorders, so sometimes the orders are not delivered out of stock immediately. We assume here that clients are prepared to wait for a certain time and that the orders will be delivered, but later than the moment of ordering. In other circumstances we might have stock breaks due to stochastic effects in leadtime demand. This has some implications on the previously defined inventory functions. There will be moments in time when the actual inventory $I(t)$ is zero, but there are some outstanding orders, these are indicated by $B(t)$, the amount of backorders at time t. A new function, the net inventory at time t, $IN(t)$, is defined. In some text books $IN(t)$ is named 'inventory level' instead of 'net inventory'. This function $IN(t)$ is equal to $I(t)$, when the inventory is positive, and equal to $-B(t)$, when there are backorders. These new functions will require us to redefine the inventory position $IP(t)$ as (2.64). These relations are visualized in Figure 2.18.

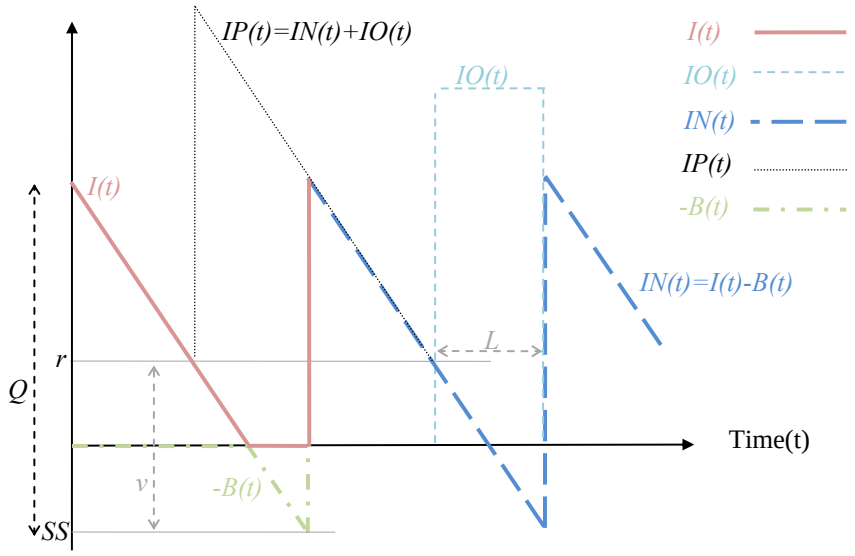

Figure 2.18: Inventory functions: I(t), IP(t), IO(t), IN(t) and B(t)

This system has two decision variables: the order quantity Q and the reorder point r. The reorder point r equals the sum of the safety stock SS and the demand during leadtime ν, see (2.66). In Figure 2.18 we have a negative safety stock SS, as it is beneficial to make some customers wait, under the assumption they are prepared to wait.

$$IP(t) = IN(t) + IO(t) \tag{2.64}$$

$$IN(t+L) = IN(t) + IO(t) - \nu = IP(t) - \nu \tag{2.65}$$

$$r = SS + \nu \tag{2.66}$$

Throughout this dissertation we will make use of the following indicators to express the service level, backorders or stockout indicators:

- S_1: Replenishment rate: probability of no stockout per replenishment cycle
- S_2: Fill rate service level: fraction of demand that can be satisfied immediately from stock on hand
- S_3: Ready rate service level: fraction of time with positive net inventory $IN(t)$
- S_{OL}: Percentage of order lines delivered out of stock
- $\overline{A}$: Stockout frequency, $\overline{A} = 1 - S_3$
- $\overline{B}$: Average backorders
- $\overline{P}$: Average amount new backlogs incurred, $\overline{P} = \lambda(1 - S_2)$

S_1 is the specified probability of no stockout per replenishment cycle, see Silver et al. (1998) and Waters (2003). We will see later that this service level definition has no real value in practice, see section 3.1.

$$S_1 = \frac{\textit{Number of replenishments where } IN(t) > 0 \textit{ at replenishment arrival}}{\textit{Total number of replenishment cycles}} \tag{2.67}$$

S_2 is the fill rate service level that equals the fraction of demand that can be satisfied immediately from stock on hand.

$$S_2 = \frac{\textit{Units of demand delivered directly out of stock}}{\textit{Total units demanded}} \tag{2.68}$$

S_3 is the ready rate service level which is the fraction of time with positive net inventory $IN(t)$. Due to the linearity between time and quantities of the simple model considered here, the fill rate S_2 and the ready rate S_3 are

equal to each other. So if $IN(t) > 0$ in 90 % of the time, the fill rate and the ready rate are both 90%.

$$S_3 = \frac{Time\ when\ IN(t) > 0}{Total\ time} \tag{2.69}$$

S_{OL} is seen very often in practice, it is the % of order lines that can be delivered out of stock. An order line refers to one SKU (stock keeping unit) of which one or multiple items can be ordered. If 5 units are demanded and only 4 can be delivered out of stock, this leads to $S_{OL} = 0\%$ for this order line. So this service level definition is definitely more severe than the fill rate S_2 definition. In case of a Poisson demand S_{OL} and S_2 are equal. Within this dissertation we will develop order line service level equations in case of compound Poisson demand for several replenishment policies, see section 3.2.

$$S_{OL} = \frac{Number\ of\ order\ lines\ delivered\ out\ of\ stock\ in\ full}{Total\ number\ of\ order\ lines} \tag{2.70}$$

$\overline{P}$ is the average new backlog, this is related to the fill rate, see (2.71).

$$\overline{P} = \lambda(1 - S_2) \tag{2.71}$$

To clarify visually the different performance indicators we define four intermediate variables in Figure 2.19: S_{2i}, S_{3i}, I^i and B^i. The formulations for these new performance indicators are given by respectively (2.72), (2.73), (2.74) and (2.75). Later we will refine (2.72) for the cases where we do not have a constant leadtime demand.

$$S_{2i} = \begin{cases} Q - \nu + r & \text{if } r < \nu \\ Q & \text{if } r \geq \nu \end{cases} \tag{2.72a}$$

$$S_2 = \frac{S^{2i}}{Q} \tag{2.72b}$$

$$A(t) = \begin{cases} 1 & \text{if } IN(t) \leq 0 \\ 0 & \text{if } IN(t) > 0 \end{cases} \tag{2.73a}$$

$$S_3 = lim_{T\to\infty}\left[\frac{\int_0^T (1 - A(t))\, dt}{T}\right] = \frac{S_{3i}}{u} = 1 - \overline{A} \tag{2.73b}$$

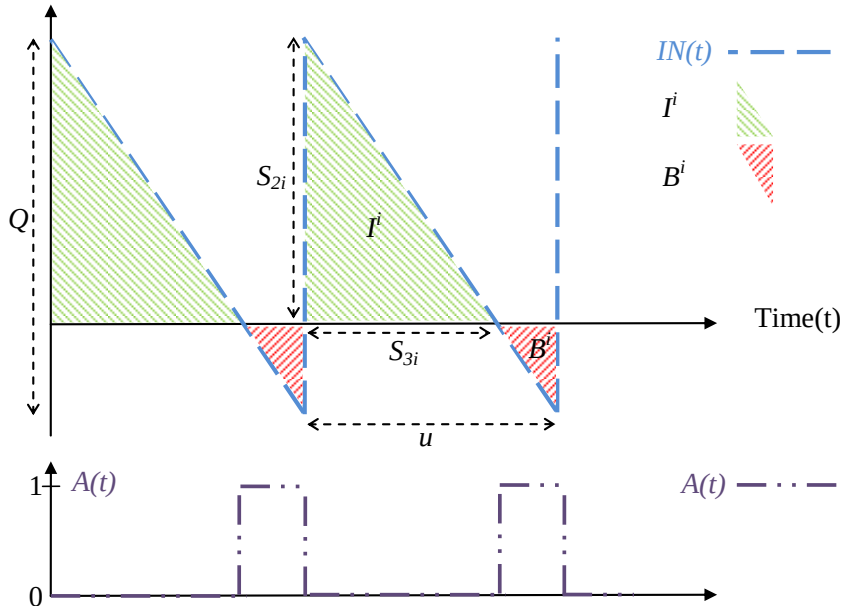

Figure 2.19: Inventory performance indicators

$$\overline{I} = lim_{T\to\infty}\left[\frac{\int_0^T I(t)dt}{T}\right] = \frac{I^i}{u} \tag{2.74}$$

$$\overline{B} = lim_{T\to\infty}\left[\frac{\int_0^T B(t)dt}{T}\right] = \frac{B^i}{u} \tag{2.75}$$

2.3.3 Constant leadtime demand & backorders

When $\nu \geq r$, thus having a backlog, results in the simplified KPI's (2.76)-(2.80):

$$S_2 = S_3 = 1 - \frac{\nu - r}{Q} \tag{2.76}$$

$$\overline{I} = \frac{(Q + SS)^2}{2Q} \tag{2.77}$$

$$\overline{B} = \frac{(SS)^2}{2Q} \tag{2.78}$$

$$\overline{A} = \frac{\nu - r}{Q} \tag{2.79}$$

$$\overline{P} = \lambda \overline{A} \tag{2.80}$$

To express the cost we use a set of additional marginal cost parameters b, a and p. b is the shortage cost per unit and per time unit *(moneys/[quantity-unit*time-unit])*. a is the shortage cost per time unit (*moneys/time-unit*) and p is the shortage cost per unit (*moneys/quantity-unit*). So for each unit not delivered on time, there is a cost b for each day it is in backlog, a cost a per time unit there is a stockout and a cost p for each unit not delivered directly out of stock. The total average cost is (2.81), in practice only one, or sometimes none, of the three shortage costs is applied: b, a or p.

$$C(r, Q) = k\overline{OF} + c\lambda + h\overline{I} + b\overline{B} + a\overline{A} + p\overline{P} \tag{2.81}$$

For the time being we will consider the case where $a = p = 0$. We define a cost ratio ω (2.82) so we can immediately see the resemblance with the optimal order quantity in case there is no backlog. By equalizing the partial derivatives equal to zero we become the conditions for Q^* (2.83) and r^* (2.84) in order to minimize the cost.

$$\omega = \frac{b}{b + h} \tag{2.82}$$

$$Q^* = \sqrt{\frac{2k\lambda}{h\omega}} \tag{2.83}$$

$$r^* = \nu - (1 - \omega)\, Q^* \tag{2.84}$$

This yields the optimal cost (2.85) for an inventory system with a constant leadtime demand and backorders. The optimal variable cost is (2.86).

$$C^* = c\lambda + \sqrt{2k\lambda h\omega} \tag{2.85}$$

$$C_v^* = \sqrt{2k\lambda h\omega} \tag{2.86}$$

Example 2.8 Constant leadtime demand with backorders

In Example 2.8, Table 2.6, we retake Example 2.7 (Table 2.5). We now allow backorders with $b = 50$ and have a leadtime $L = 0.1$.

Input	Value	KPI	Value
λ	20	Q^*	12.81
b	50	r^*	-3.00
h	32	C_v^*	249.88
k	80	$\overline{I}$	2.38
L	0.1	$\overline{B}$	0.98
ν	2	$\overline{OF}$	1.56
ω	0.61	$\overline{A}$	0.39
a	0	$\overline{P}$	7.8
p	0	$S_2 = S_3$	61%

Table 2.6: Example 2.8: Constant leadtime demand with backorders

This results in $r^* = -3$ and $Q^* = 12.80$, so we have an increase of the order quantity compared to Example 2.7 due to the allowed backorders, see Figure 2.20.

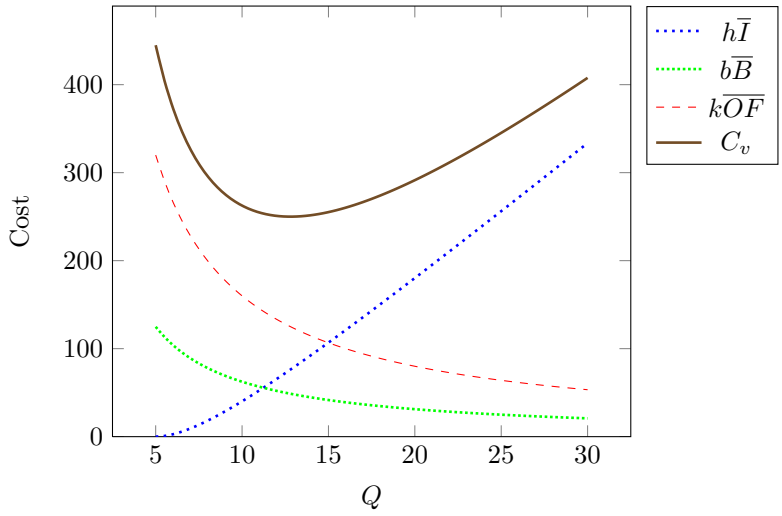

Figure 2.20: Example 2.8: Constant demand with backorders costs

In Figure 2.21 we plot three cost curves where we have set the reorder point to three different values: $r = 2$, $r = -3$ and $r = -8$. Here we see clearly that the curve with $r = -3$ has the lowest minimum.

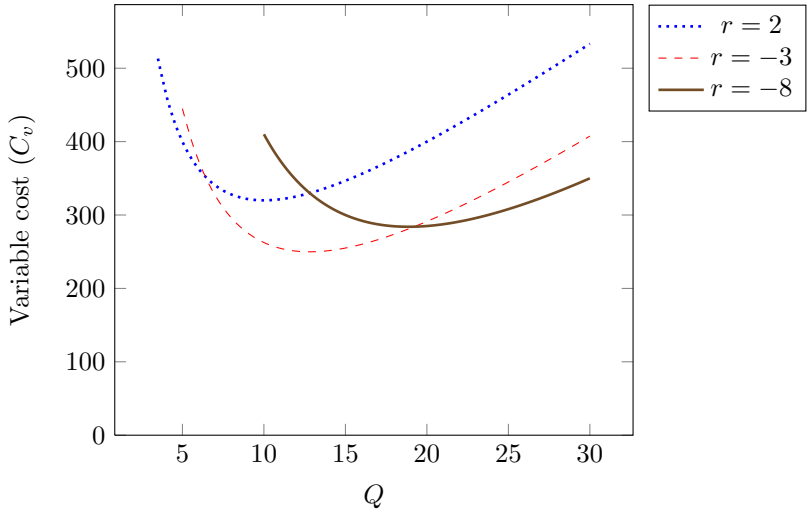

Figure 2.21: Example 2.8: Constant demand with backorders costs

In Figure 2.22 a three dimensional representation is given of the variable cost function C_v, depending on the order quantity Q and the reorder point r.

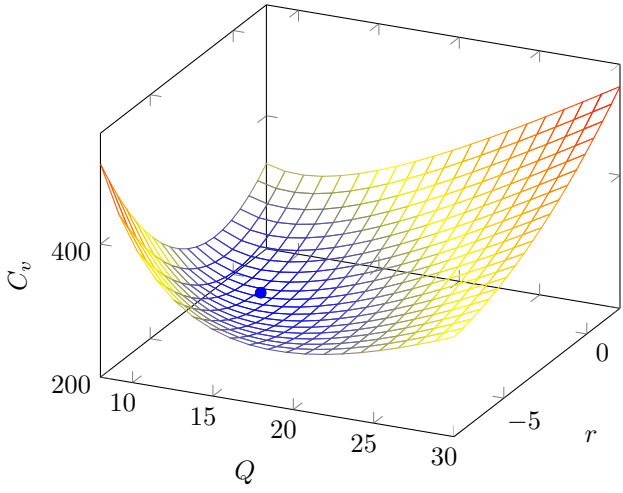

Figure 2.22: Example 2.8: Constant demand with backorders costs (3D)

2.3.4 Base-stock

Within a spare parts environment it is assumed to have small demand rates and a high unit purchase costs c, inducing a base-stock policy (bS). An (r, Q) model is an inventory model where an order of size Q is placed as soon as the net inventory *IN(t)* reaches the reorder point r. A base-stock model is an (r, Q) model where $Q = 1$. As the unit purchase cost (c) is much larger than the ordering cost (k), the ordering cost is considered negligible and not withheld in the formulation. The target stock level of a base-stock model is s, this is linked to the (r, Q) model through (2.87).

$$s = r + 1 = \textit{base-stock target level} \tag{2.87}$$

Within a base-stock level the inventory position *IP(t)* will always be kept at the target stock level s. So each customer order will trigger an order for the supplier. We now no longer consider a constant demand, but a stochastic demand.

Example 2.9 Base-stock policy

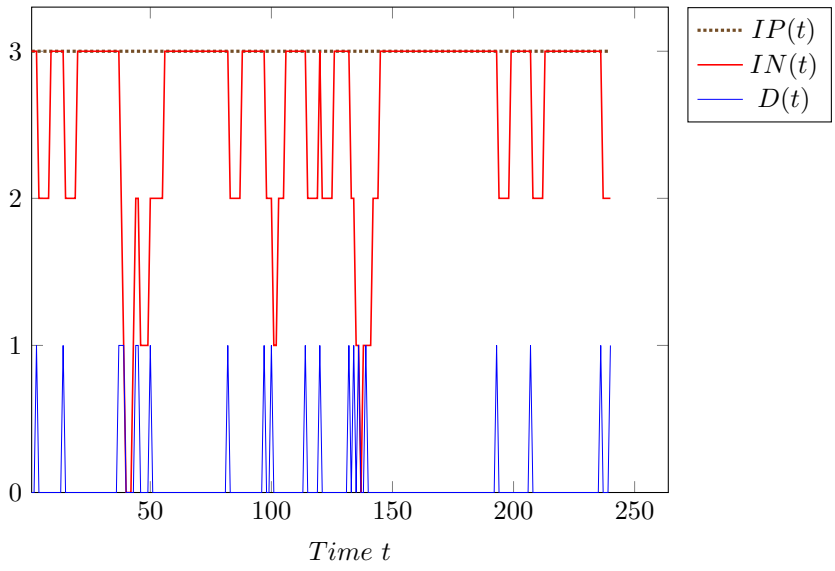

Figure 2.23: Example 2.9: base-stock policy

In Figure 2.23 we give an evolution of $IP(t)$, $IN(t)$ and $D(t)$. We oversee a period of 240 working days, the year demand is $\lambda = 25$, leadtime $L = 5$ days and the target stock level is $s = 3$. Each time when a customer arrives, we notice that $IN(t)$ decreases with one. The inventory position is constant at value 3, the target stock level s in this example.

The previously defined inventory functions $I(t)$, $B(t)$, $IP(t)$, $IO(t)$ and $IN(t)$ also exist. For each of these functions we can define an equilibrium function, which is a random variable having the limiting distribution of the inventory process. So there exists an equilibrium for each of the previously defined inventory processes.

- I= equilibrium inventory, a random variable having the limiting distribution of the stochastic process **I**= $\{I(t) : t \geq 0\}$
- IN= equilibrium net inventory, a random variable having the limiting distribution of the stochastic process **IN**= $\{IN(t) : t \geq 0\}$
- IP= equilibrium inventory position, a random variable having the limiting distribution of the stochastic process **IP**= $\{IP(t) : t \geq 0\}$
- IO= equilibrium inventory on order, a random variable having the limiting distribution of the stochastic process **IO**= $\{IO(t) : t \geq 0\}$
- B= equilibrium backorders, a random variable having the limiting distribution of the stochastic process **B**= $\{B(t) : t \geq 0\}$
- A= equilibrium stockout indicator, a random variable having the limiting distribution of the stochastic process **A**= $\{A(t) : t \geq 0\}$
- D_L= equilibrium leadtime demand, a random variable having the limiting distribution of the stochastic demand process during leadtime **D**= $\{D(t, t + L) : t \geq 0\}$

With these new definitions we can rephrase the relations between the net inventory equilibrium IN and the inventory on order equilibrium IO as follows for a stochastic demand base-stock policy (2.88):

$$IN = s - IO \tag{2.88}$$

As the inventory on order IO equals the demand during leadtime in case of a base-stock policy, (2.88) can be reformulated as (2.89) and (2.90):

$$IN = s - D_L \tag{2.89}$$

$$Pr(IN = k) = Pr(D_L = s - k) \tag{2.90}$$

We use (2.90) to formulate the performance indicators. $\overline{A}$ (2.91) is the stockout frequency, being the probability that the net inventory is non-positive. $\overline{B}$ (2.93) is the average number of backorders or $-\overline{IN^-}$, where IN^- is given by 2.92. For calculating the average inventory (2.95) we make use of the average number of backorders and state that the average inventory equals the target stock level minus the average demand during leadtime, but we need to increase this with the average number of backorders. The rate at which new shortages are incurred is the derivative of $\overline{B}$ with respect to time,

see (2.94). As long as there is a continuous demand or a unit demand (Poisson), there is a direct link between $\overline{A}$ and $\overline{P}$, namely $\overline{P} = \lambda\overline{A}$. If there is a bulky demand, like compound Poisson, this relationship between $\overline{A}$ and $\overline{P}$ no longer holds. The order frequency (2.96) remains the same as previously defined.

$$\overline{A} = Pr(IN \leq 0) = Pr(D_L \geq s) \tag{2.91}$$

$$IN^{-} = \begin{cases} 0 & \text{if } IN > 0 \\ IN & \text{if } IN(t) \leq 0 \end{cases} \tag{2.92}$$

$$\begin{aligned} \overline{B} &= E\left[IN^{-}\right] \\ &= E\left[[D_L - s]^{+}\right] \end{aligned} \tag{2.93}$$

$$\overline{P} = \frac{d(\overline{B})}{dL} \tag{2.94}$$

$$\begin{aligned} \overline{I} &= E\left[IN + [IN]^{-}\right] \\ &= s - \nu + \overline{B} \end{aligned} \tag{2.95}$$

$$\overline{OF} = \lambda/Q = \lambda \tag{2.96}$$

For the base-stock model we will assume that $k = 0$ and as such the average total cost (2.81) can be reduced to (2.97) and the average variable cost becomes (2.98):

$$C(s) = c\lambda + h\overline{I} + b\overline{B} + a\overline{A} + p\overline{P} \tag{2.97}$$

$$C_v(s) = h\overline{I} + b\overline{B} + a\overline{A} + p\overline{P} \tag{2.98}$$

2.3.4.1 Base-stock Poisson demand

In case of a demand process with a Poisson distribution we can use the performance indicators formulations (2.91), (2.93) and (2.95) together with the Poisson formulations (2.1) - (2.97) to reformulate them. We start by defining the demand during leadtime distribution D_L (2.99) and net inventory IN probability (2.100).

$$Pr(D_L = k) = g(k) = \frac{\nu^k e^{-\nu}}{k!} \qquad k \geq 0, k \in \mathbb{Z} \tag{2.99}$$

$$Pr(IN = k) = Pr(D_L = s - k) = g(s - k) \tag{2.100}$$

$$\overline{A} = 1 - G(s - 1) = G^0(s - 1) \tag{2.101}$$

$$\overline{B} = G^1(s) \tag{2.102}$$

$$\overline{P} = \lambda \overline{A} \tag{2.103}$$

$$\overline{I} = s - \nu + \overline{B} \tag{2.104}$$

In case of Poisson demand $\overline{A}$ can be used to express the fill rate S_2. This fact reflects a fundamental property of Poisson processes, know by the PASTA acronym (Poisson Arrivals See Time Averages), see Zipkin (2000). The time average in question here is $\overline{A}$, and PASTA asserts that a typical arriving customer finds no inventory with frequency $\overline{A}$. Also in the next sections we will use PASTA to move from $S_3 = 1 - \overline{A}$ to fill rate or order line in case of Poisson or compound Poisson. As the PASTA property does not hold for non-Poisson demands, we cannot always state that $S_2 = 1 - \overline{A}$.

$$S_1 = S_3 = S_{OL} = S_2 = 1 - \overline{A} = G(s - 1) \tag{2.105}$$

$$\overline{OF} = \lambda \tag{2.106}$$

To find the optimal target stock level s^*, when $a = 0$ and $p = 0$, we also can ignore the purchasing cost $c\lambda$ as this is a fixed cost. Setting the first derivative of the cost equal to zero and making use of the cost ratio ω (2.82) we find the following condition for the optimal base-stock target level s^*, (2.107) and (2.108):

$$\frac{\partial C_v(s)}{\partial s} = h + (h + b)\frac{\partial G^1(s)}{\partial s} = h - (h + b)G^0(s) = 0 \tag{2.107}$$

$$G^0(s^* - 1) > \frac{h}{b + h} = 1 - \omega \geq G^0(s^*) \tag{2.108}$$

Example 2.10 Base-stock Poisson demand

Input	Value	KPI	Value
λ	20	s^*	2
h	32	C_v^*	44.39
k	0	$\overline{I}$	0.54
b	50	$\overline{B}$	0.54
L	0.1	$\overline{A}$	0.59
ν	2	$\overline{OF}$	20
σ^2	2	$S_1 = S_2 = S_3$	41%
ω	0.61	$\overline{P}$	11.88
$a = p$	0		

Table 2.7: Example 2.10: Base-stock Poisson demand

In Example 2.10, see Table 2.7 we continue on Example 2.8, Table 2.6. We now have a stochastic demand with a Poisson distribution, so $\nu = \sigma^2 = 2$. We set $k = 0$, as we are in a base-stock policy.

We can use Figure 2.3 and Table 2.1 from Example 2.1 on the Poisson distribution to calculate the base-stock Poisson demand KPI's. As the shortage cost, $b = 50$, is not set very high in comparison with the holding cost, $h = 32$, we see that only a low service level S_2 and S_3 of 41 % is reached. This is because it is cheaper to have backlogs than to hold extra stock. This can be understood visually by looking at Figure 2.19, the I^i surface would increase much more than would decrease the surface B^i by increasing the reorder point r.

In Figure 2.24 we visualize the cost curves of Example 2.10. The minimum cost is realized when $s^* = 2$ and the minimum variable cost $C_v = 44.39$.

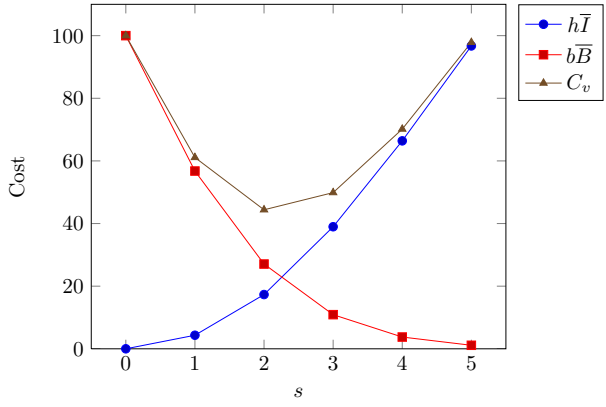

Figure 2.24: Example 2.10: Base-stock cost, Poisson demand

2.3.4.2 Base-stock compound Poisson demand

The performance indicators in case of a base-stock policy, see (2.91)-(2.97), can be transformed to the equations (2.109)-(2.118). Here we have used the compound Poisson distribution functions: (2.21)-(2.27).

$$Pr(D_L = d) = g_Y(d) = \sum_{k=0}^{\infty} \left[\frac{(\tau L)^k e^{-\tau L}}{k!} f_d^k \right] \tag{2.109}$$

$$Pr(IN = k) = Pr(D_L = s - k) = g_Y(s - k) \tag{2.110}$$

$$\overline{A} = 1 - G_Y(s - 1) = G_Y^0(s - 1) \tag{2.111}$$

$$\overline{B} = G_Y^1(s) \tag{2.112}$$

$$\overline{I} = s - \nu + \overline{B} \tag{2.113}$$

$$S_3 = 1 - \overline{A} = 1 - G_Y^0(s - 1) = G_Y(s - 1) \tag{2.114}$$

$$\overline{OF} = \lambda \tag{2.115}$$

Axsäter (2006) discussed the fill rate service level S_2 for an (r, Q) policy and a compound Poisson demand. We have reformed and simplified it to an explicit base-stock policy, (2.116). S_2 is the ratio between the expected quantity delivered to the customer and the total quantity demanded. The delivered quantity is the minimum between the quantity demanded d and the net inventory level $(IN = k)$, see (2.116). We express the expected new shortages $\overline{P}$ using the fill rate S_2, see (2.117).

$$\begin{aligned} S_2 &= \frac{\sum_{d=1}^{\eta} \sum_{k=1}^{s} min(d, k) f_d \, Pr(IN = k)}{\chi} \\ &= \frac{\sum_{d=1}^{\eta} \sum_{k=1}^{s} min(d, k) f_d \, g_Y(s - k)}{\chi} \end{aligned} \tag{2.116}$$

$$\overline{P} = \lambda(1 - S_2) \tag{2.117}$$

The order frequency performance indicator as defined by (2.118) is only valid if each basic batch size will be a separate order. In case multiple batch quantities can be grouped into one order, a more complex definition of the order frequency than (2.118) is necessary.

$$\overline{OF} = \lambda / Q \tag{2.118}$$

Just like for the Poisson demand we can find the optimal target stock level, if $a = p = 0$, through (2.119):

$$G_Y^0(s^* - 1) > \frac{h}{b+h} = 1 - \omega \geq G_Y^0(s^*) \tag{2.119}$$

2.3.4.3 Base-stock normal demand

The performance indicators in case of a base-stock policy, see (2.91)-(2.97), can be transformed to the equations (2.120)-(2.129) if there is a normal demand distribution. Here we have used the normal distribution functions: (2.36)-(2.42). S_{OL} is not defined here, as it only makes sense for discrete demand. As an approximation and upper bound we can use the fill rate S_2.

$$\frac{Pr(x < D_L \leq x + dx)}{dx} = \nu + \sigma\varphi(z_{(x)}) \tag{2.120}$$

$$Pr(IN \leq x) = \Phi^0(z_{(s-x)}) \tag{2.121}$$

$$\overline{A} = \Phi^0(z_{(s)}) \tag{2.122}$$

$$\overline{B} = \Phi^1(z_{(s)})\sigma \tag{2.123}$$

$$\overline{P} = \lambda\overline{A} \tag{2.124}$$

$$\overline{I} = s - \nu + \overline{B} = [\frac{s-\nu}{\sigma} + \Phi^1(\frac{s-\nu}{\sigma})]\sigma = \Phi^1(\frac{-s+\nu}{\sigma})\sigma = \Phi^1(-z_{(s)})\sigma \tag{2.125}$$

$$S_2 = S_3 = 1 - \overline{A} = 1 - \Phi^0(z_{(s)}) = \Phi(z_{(s)}) \tag{2.126}$$

$$S_1 = Pr(D_L < s) = \Phi(z_{(s)}) \tag{2.127}$$

$$\overline{OF} = \lambda \tag{2.128}$$

The optimal target stock level, when $a = p = 0$, is found by:

$$z_{(s^*)} = \Phi^{0inv}(1 - \omega) \tag{2.129}$$

2.3.5 Newsvendor

A newsvendor problem is a single period model with a stochastic demand and penalty costs for ordering too much (c_o for each item ordered but not sold) or too little (c_u for each demand that cannot be satisfied). So we have one period where we need to decide the order quantity Q. There is a cost for each product not sold at the end of the period and a cost for each demand not met. We assume a stochastic demand that is normally distributed, φ is the standard normal probability density function (2.36). The total average cost $C(Q)$ is (2.85).

$$C(Q) = c_o \int_{-\infty}^{Q} (Q - x)\, \varphi(z_{(x)}) dx + c_u \int_{Q}^{\infty} (x - Q)\, \varphi(z_{(x)}) dx \tag{2.130}$$

Setting the first derivative from $C(Q)$ equal to zero gives (2.131) and (2.132).

$$\Phi\left(z_{(Q)}\right) = \frac{c_u}{c_u + c_o} \tag{2.131}$$

$$z_{(Q)} = \Phi^{inv}\left(\frac{c_u}{c_u + c_o}\right) \tag{2.132}$$

2.3.6 (r, Q) policy

In an (r, Q) policy an order of size Q is placed as soon as the inventory position would fall to or below the reorder point r. In case of Poisson demand the **IP** process is cyclic: $r+Q \rightarrow r+Q-1 \ ... \ \rightarrow r+1 \rightarrow r+Q \ ...$ In case of bulky demand (compound Poisson), we define $IP_c(t) = r + Q - IP(t)$. $\mathbf{IP}_c$ is a continuous time Markov chain and is irreducible. The sequence of the state is no longer sequential, but changes at demand epochs. We must assume for compound Poisson demand that not all demands can be multiples of one integer larger than one, see section 2.1.3. The stationary frequency distribution of the stochastic process **IP** is uniform in the range $[r+1, r+Q]$, see (2.133). Just like in the previous base-stock policy we can describe the relation between IN, IP and D_L, see the inventory functions in (2.134) and their equilibrium random variables (2.135).

$$Pr(IP = i) = 1/Q \quad i \in [r+1, r+Q] \tag{2.133}$$

$$IN(t+L) = IP(t) - D(t, t+L) \tag{2.134}$$

$$IN = IP - D_L \tag{2.135}$$

Example 2.11 (r, Q) **policy**

In Figure 2.25 we visualize the inventory functions $IP(t)$, $IN(t)$ and the demand $D(t)$. The year demand is $\lambda = 9.500$, the leadtime $L = 20$ days, the order quantity $Q = 780$ and the reorder point $r = 1.561$.

As IP and D_L are independent, we can reuse the characteristics from the base-stock policy to formulate the performance indicators for an (r, Q) policy. The distribution of IN can be considered as the average of a set of base-stock policies with the target stock level s at values in the range of $[r+1, r+Q]$. As the other performance indicators are deducted from IN we can also reuse the base-stock formulations (2.91)-(2.97) and define them as an average from the base-stock performance indicators, see (2.136)-(2.139).

$$\overline{A}(r, Q) = \frac{1}{Q} \sum_{s=r+1}^{r+Q} \overline{A}(s) \tag{2.136}$$

$$\overline{B}(r, Q) = \frac{1}{Q} \sum_{s=r+1}^{r+Q} \overline{B}(s) \tag{2.137}$$

$$\overline{P}(r, Q) = \frac{1}{Q} \sum_{s=r+1}^{r+Q} \overline{P}(s) \tag{2.138}$$

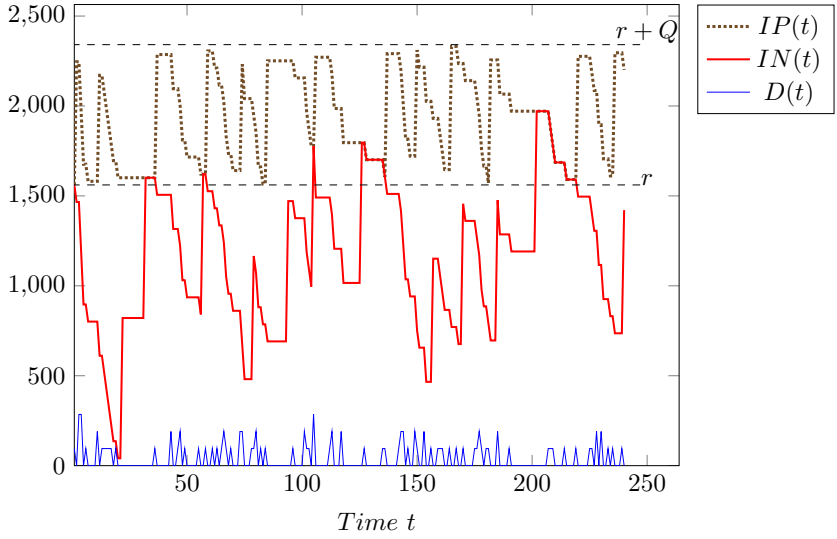

Figure 2.25: Example 2.11: (r, Q) policy

$$\overline{I}(r,Q) = \frac{1}{Q}\sum_{s=r+1}^{r+Q} \overline{I}(s) \tag{2.139}$$

The order frequency (2.140) still follows the formulation as previously defined.

$$\overline{OF} = \lambda/Q \tag{2.140}$$

$$S_1 = Pr(D_L \leq r) \tag{2.141}$$

$$C_v(r,Q) = k\overline{OF} + h\overline{I} + b\overline{B} + a\overline{A} + p\overline{P} \tag{2.142}$$

$$C(r,Q) = c\lambda + C_v(r,Q) \tag{2.143}$$

2.3.6.1 (r, Q) Poisson demand

In case of a normal demand and an (r, Q) we also assume that the inventory position IP is uniformly distributed on the range $[r, r + Q]$. Under these conditions it is allowed to use the base-stock performance indicators for a Poisson demand (2.99-2.106) to reformulate these for an (r, Q) policy: (2.144) - (2.152).

$$Pr(D_L = k) = g(k) = \frac{\nu^k e^{-\nu}}{k!} \qquad k \geq 0, k \in \mathbb{Z} \tag{2.144}$$

$$Pr(IN = j) = \frac{1}{Q} \sum_{k=max(r+1,j)}^{r+Q} g(k-j) \text{ where } j \leq r + Q \tag{2.145}$$

$$\overline{A} = \frac{1}{Q} \left[G^1(r) - G^1(r+Q) \right] \tag{2.146}$$

$$\overline{B} = \frac{1}{Q} \left[G^2(r) - G^2(r+Q) \right] \tag{2.147}$$

$$\overline{P} = \lambda \overline{A} \tag{2.148}$$

$$\overline{I} = \frac{Q+1}{2} + r - \nu + \overline{B} \tag{2.149}$$

$$S_2 = S_3 = S_{OL} = 1 - \overline{A} \tag{2.150}$$

$$S_1 = Pr(D_L \leq r) = G(r) \tag{2.151}$$

$$\overline{OF} = \lambda / Q \tag{2.152}$$

Example 2.12 (r, Q) **Poisson demand**

Input	Value	KPI	Value
λ	20	r^*	1
h	32	Q^*	12
k	80	C_v^*	300.13
b	100	$\overline{I}$	3.92
L	0.2	$\overline{B}$	0.42
ν	4	$\overline{A}$	0.25
σ^2	4	$\overline{OF}$	1.67
a	0	$\overline{P}$	5.03
p	0	$S_2 = S_3 = S_{OL}$	75%
		S_1	9%

Table 2.8: Example 2.12: (r, Q) *Poisson demand*

In Example 2.12 in Table 2.8 we reuse the input from the base-stock Poisson Example 2.10, Table 2.7, but now there is an ordering cost $k = 80$.

We have also increased the shortage cost $b = 100$ and have set the leadtime $L = 0.2$. For an (r, Q) policy replenishment rate S_1 is no longer equal to the fill rate and other service definitions, the $S_1 = 9\%$ is remarkably lower than the other service levels $S_2 = S_3 = S_{OL} = 75\%$. The low S_1 service level can be explained because in nearly each replenishment cycle there is a stockout situation, but as the order quantity Q is quite large, $Q = 12$ is 60% of the year demand, the backorder size when a new order arrives is on average 3 pieces or 25% $(1 - S_2)$ of the order quantity.

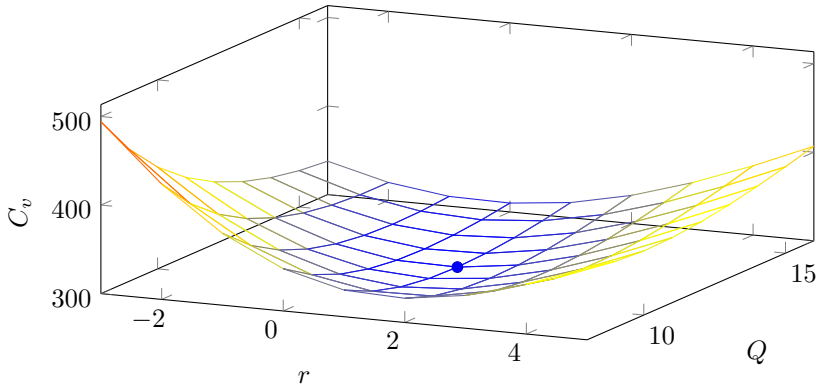

Figure 2.26: Example 2.12: (r, Q) cost, Poisson demand (3D)

Federgruen and Zheng (1992) have worked out a surprisingly simple and efficient algorithm for the determination of an optimal (r, Q) policy. This algorithm is developed for compound (Poisson) demand, and as such it is also applicable for Poisson demand. The cost of Example 2.12 is plotted in three dimensions in Figure 2.26. Figure 2.27 gives a representation of three cost curves for respectively $Q = 8$, $Q = 12$ and $Q = 16$, while the reorder point r ranges from -3 up to 5.

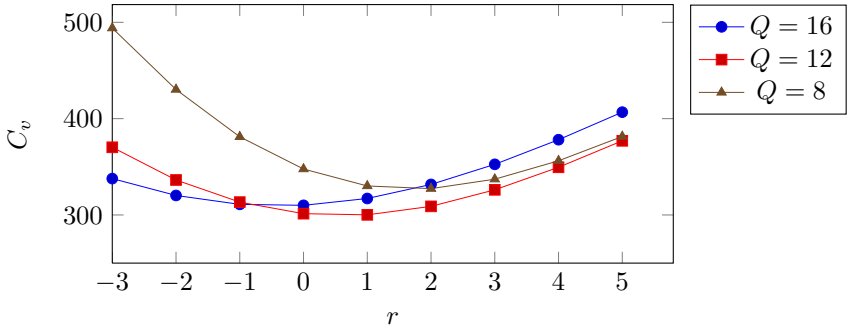

Figure 2.27: Example 2.12: (r, Q) cost, Poisson demand

2.3.6.2 (r, Q) compound Poisson demand

Also for the compound Poisson demand we can reuse the base-stock definitions and take the average over the range $[r + 1, r + Q]$. For compound demand the S_2 and $\overline{P}$ definition require another calculation due to bulky demand.

$$Pr(D_L = d) = g_Y(d) = \sum_{k=0}^{\infty} \left[\frac{(\tau L)^k e^{-\tau L}}{k!} f_d^k \right] \tag{2.153}$$

$$Pr(IN = j) = \frac{1}{Q} \sum_{k=max(r+1,j)}^{r+Q} g_y(k-j) \ \textit{where} \ j \leq r + Q \tag{2.154}$$

$$\overline{A} = \frac{1}{Q} \left[G_Y^1(r) - G_Y^1(r+Q) \right] \tag{2.155}$$

$$\overline{B} = \frac{1}{Q} \left[G_Y^2(r) - G_Y^2(r+Q) \right] \tag{2.156}$$

$$\overline{P} = \lambda(1 - S_2) \tag{2.157}$$

$$\overline{I} = \frac{Q+1}{2} + r - \nu + \overline{B} \tag{2.158}$$

$$S_3 = 1 - \overline{A} \tag{2.159}$$

$$\overline{OF} = \lambda / Q \tag{2.160}$$

In (2.161) we have reformulated the fill rate service level as it was defined by Axsäter (2006). We limited the upper bounds of the summations to respectively η and $r + Q$. This allows a more rapid calculation, instead of setting them equal to $+\infty$.

$$S_2 = \frac{\sum_{d=1}^{\eta} \sum_{k=1}^{r+Q} min(d, k) f_d \, Pr(IN = k)}{\chi} \tag{2.161}$$

2.3.6.3 (r, Q) Normal demand

The principle that (r, Q) can be seen as the average of a range of base-stock policies where the target stock level varies over the range $[r, Q]$ also stands for a continuous normal demand. So we can reuse the normal demand base-stock policy KPI's: (2.120)-(2.125). Before going to the full definition, we first want to point out that in practice very often simplifications are used, as defined by (2.162)-(2.163). In section 3.3.3 we will give the conditions when simplifications can be applied without jeopardizing the quality of the result.

$$\overline{A} = \frac{\sigma}{Q}\Phi^1(z_{(r)}) \tag{2.162}$$

$$\overline{B} = \frac{\sigma^2}{Q}\Phi^2(z_{(r)}) \tag{2.163}$$

Without simplifications, the KPI equations are (2.164)-(2.171).

$$\frac{Pr(x < D_L \leq x + dx)}{dx} = \nu + \sigma\varphi(z_{(x)}) \tag{2.164}$$

$$\begin{aligned} Pr(IN \leq x) &= \frac{1}{Q}\int_r^{r+Q} \Phi^0(z_{(u-x)})du \\ &= \frac{\sigma}{Q}\left[\Phi^1(z_{(r-x)}) - \Phi^1(z_{(r+Q-x)})\right] \end{aligned} \tag{2.165}$$

$$\overline{A} = \frac{\sigma}{Q}\left[\Phi^1(z_{(r)}) - \Phi^1(z_{(r+Q)})\right] \tag{2.166}$$

$$\overline{B} = \frac{\sigma^2}{Q}\left[\Phi^2(z_{(r)}) - \Phi^2(z_{(r+Q)})\right] \tag{2.167}$$

$$\overline{I} = \frac{Q}{2} + r - \nu + \overline{B} \tag{2.168}$$

$$\overline{P} = \lambda\overline{A} \tag{2.169}$$

$$S_2 = S_3 = S_{OL} = 1 - \overline{A} \tag{2.170}$$

$$\overline{OF} = \lambda/Q \tag{2.171}$$

The replenishment rate S_1 service level is not an average, as it only looks to the probability of no stockout per replenishment cycle, we only need to look at the reorder point r and it is not influenced by the order quantity Q. This is the major reason why it should not be used in practice, as it does not represent a good service for the client.

$$S_1 = Pr(D_L \leq r) = \Phi(z_{(r)}) \tag{2.172}$$

2.3.7 (s, S) policy

In an (s, S) policy an order is placed to reach the stock maximum level S as soon as stock falls to or below reorder point s. The order quantity is not fixed here, but is set in a way that the inventory position IP reaches the value S. There is no difference between an (r, Q) or (s, S) policy if the order is placed immediately if the reorder point is reached. This is the case if there is continuous review and continuous demand or Poisson demand. Here we will always assume a continuous review. So only in case of a demand that can be greater than 1, for example the compound Poisson demand, the (s, S) will be different from the (r, Q) policy. If the inventory position IP drops to $s - 2$, an order of size $S - s + 2$ will be placed.

Example 2.13 (s, S) **policy**

In Figure 2.28 we visualize the inventory functions $IP(t)$, $IN(t)$ and the demand $D(t)$. The year demand is $\lambda = 9.500$, the leadtime $L = 20$ days, the maximum stock value is $S = 2.341$ and the reorder point $s = 1.561$. We used the same parameters as applied for Figure 2.25, we can clearly see that in case of an (s, S) policy the IP reaches the maximum stock value S each time an order is placed. This was definitely not the case in the (r, Q) policy, see Figure 2.25.

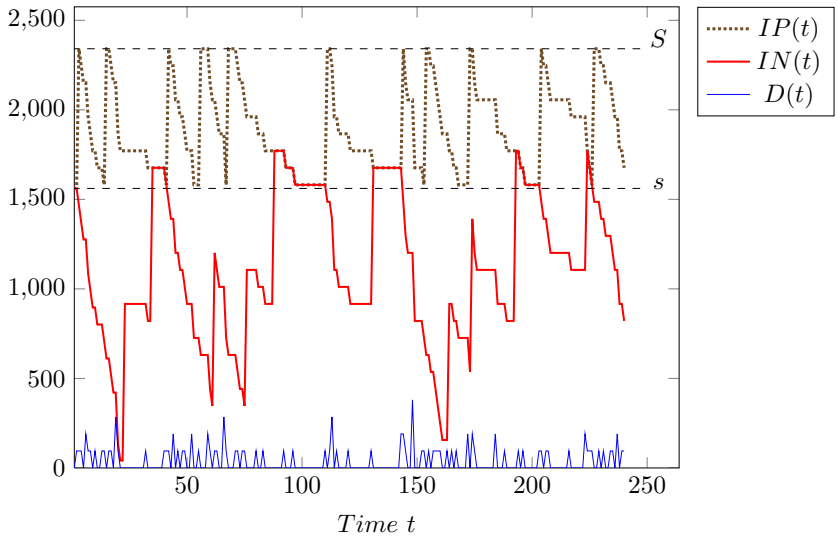

Figure 2.28: Example 2.13: (s, S) *policy*

As a consequence the inventory position is no longer uniformly distributed as was the case for the (r, Q) policy. Instead we are here confronted

with a renewal process for each of the transitions of the inventory position IP, this process is triggered when an order is placed. We want to determine the number of customers that will invoke regeneration (a new order being placed). We shall express the probability for each value that the inventory position can take as m_k, (2.173). So m_k is the probability to reach $IP = k$ during an order cycle ($s + 1 \leq k \leq S$), so $m_S = 1$, see also Axsäter (2006). We use the expression f_d again, also used to define the compound Poisson distribution. f_d describes the probability that a customer will place an order of quantity d.

$$m_k = \sum_{i=k+1}^{S} m_i f_{(i-k)}, \text{ where } k \in [s+1, S-1] \tag{2.173}$$

Once we know m_k, that needs to be calculated recursively, we have the number of expected visits to a certain inventory position k. If we divide this by the average total number of customers during an order cycle we can express the steady-state distribution of the inventory position IP: ι_k, see (2.174).

$$\iota_k = Pr(IP = k) = \frac{m_k}{\sum_{i=s+1}^{S} m_i}, \text{ where } k \in [s+1, S] \tag{2.174}$$

Where we had a uniform distribution of IP in case of (r, Q), it was possible to simply divide by the order size Q to compute the net inventory IN. As $IN = IP - D_L$ and both IP and D_L have stochastic distributions, we need to calculate a weighted average to compute the net inventory for an (s, S) policy, using the probability for each of the IP values, see (2.175).

$$Pr(IN = k) = \sum_{i=max(s+1,k)}^{S} Pr(IP = i)\ Pr(D_L = i - k), \text{ where } k \leq S \tag{2.175}$$

In case of compound Poisson demand this can be restated as (2.176):

$$Pr(IN = k) = \sum_{i=max(s+1,k)}^{S} \iota_i\ g_Y(i - k) \tag{2.176}$$

Just like previously, starting from the net inventory function we can define the relevant performance indicators (2.177)-(2.181). In (2.179) it is important to use the proper definitions for ν, see (2.19) and (2.22) as defined for compound Poisson demand.

$$\overline{A} = \sum_{i=s+1}^{S} \iota_i G_Y^0(i-1) \tag{2.177}$$

$$\overline{B} = \sum_{i=s+1}^{S} \iota_i G_Y^1(i-1) \tag{2.178}$$

$$\overline{I} = \sum_{i=s+1}^{S} i\iota_i + r - \nu + \overline{B} \tag{2.179}$$

$$\overline{OF} = \frac{\tau}{\sum_{i=s+1}^{S} m_i} \tag{2.180}$$

$$S_2 = \frac{\sum_{d=1}^{\eta} \sum_{k=1}^{S} min(d,k) f_d \, Pr(IN = k)}{\chi} \tag{2.181}$$

2.3.8 Stochastic leadtimes and periodic review

All of the equations seen in the previous sections can be extended towards stochastic leadtimes and periodic review. Some additional insights are given in Appendix B.1 and B.2.

2.4 Multi-item inventory models annotated literature review

Contribution 1: Multi-item inventory models annotated literature review

The inventory problems we discuss in this section can be formulated by the following equations (2.182a), (2.182b), (2.182c) and (2.182d):

$$Minimize\ f(x) = \sum_{j=1}^{J} f_j(x_j) \tag{2.182a}$$

$$Subject\ to\ g_n(x) = \sum_{j=1}^{J} g_{nj}(x_j) \leq e_n,\ n = 1, .., N \tag{2.182b}$$

$$x_j \in \mathbb{R}^m\ or\ \mathbb{Z}^m,\ j = 1, .., J,\ m = 1\ or\ 2 \tag{2.182c}$$

$$x_j \geq l_j, \quad l_j \in \mathbb{R}^m\ or\ \mathbb{Z}^m \tag{2.182d}$$

There are J different inventory items. Each item has m (1 or 2) variables with lower bound(s) l_j. The decision variable values are real but can in some cases be integer. The functions f_j, g_{1j}, ..., g_{zj} are defined on $\mathbb{R}^m$ or $\mathbb{Z}^m$. We will only consider items with independent demand subject to at least one aggregate constraint ($N \geq 1$). The inventory cost of these items cannot be optimized independently due to the active aggregate constraint(s) (2.182b).

We will discuss seven instances of the inventory problem (2.182), see De Schrijver et al. (2011a). They are grouped according the following five categories: deterministic constant leadtime demand, newsvendor problem, base-stock policy, (r, Q) policy and (s, S) policy. For the newsvendor problem we have two categories: a problem with one aggregate constraint and a problem with multiple aggregate constraints. The (r, Q) policy is also split in two categories: the general approach and one without marginal costs. For each of the considered problems we give one basic model formulation, we present some extensions to this basic model and finally we discuss its practical contribution.

2.4.1 Deterministic constant leadtime demand

Let $J \geq 2$, $N = 1$, $m = 1$, $x_j \in \mathbb{R}$. The sole decision variable ($m = 1$) in this model is Q_j, the order quantity for each item. See section 2.3.1 for details on the single item model. The most simple inventory model is known as the model with the deterministic constant leadtime. In case of a single item the order quantity to realize the lowest cost is achieved through the economic

order quantity (EOQ), see (2.62). When confronted with a binding capacity constraint, e.g. limited average investment, the widespread EOQ formula is no longer applicable. In an aggregate approach the order quantities can also minimize the investment cost, where c_j is the purchase cost of item j, see (2.183a), but without violating the total number of orders per year, see (2.183b), where λ_j is the demand rate of item j. Starr and Miller (1962) determine for each item the optimal order quantity and create an 'optimal policy curve' expressing the optimal total average inventory cost for each total number of orders and vice versa, see Figure 2.29. Using Lagrange multipliers a closed-form expression is determined for this curve, see (2.184). For more information on Lagrange multipliers see section 5.2.

$$Minimize\ f(x) = \sum_{j=1}^{J} \left(\frac{c_j Q_j}{2} \right) \tag{2.183a}$$

$$Subject\ to\ g(x) = \sum_{j=1}^{J} \left(\frac{\lambda_j}{Q_j} \right) \leq e \tag{2.183b}$$

$$f(x)g(x) = \frac{1}{2} \left(\sum_{j=1}^{J} \sqrt{\lambda_j c_j} \right)^2 \tag{2.184}$$

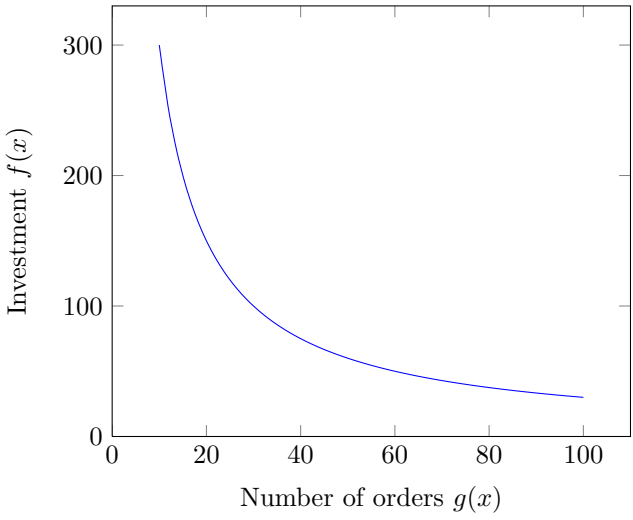

Figure 2.29: Optimal policy curve

Zipkin (2000) performs a sensitivity analysis on this 'optimal policy curve' using a 'variety index' that embodies the effective variety of the sys-

tem. He proves that the inventory turnover, ratio of demand rate and average inventory, can even decline while having rapid increasing sales. In his case study most subsidiaries of the conglomerate had an increased 'variety index' as the number of items grew faster than the revenue. Hadley and Within (1963) indicate that the Lagrangian principle can also handle multiple constraints such as average floor space and average number of orders or a lower bound (l_j) on the order quantity, but it is much more difficult to solve.

When the aggregate constraint is no longer an average but a maximum performance measurement aggregate constraint, the problem category tends to go towards a multi-product inventory problem of the shared supply process category, as defined in the introduction. When the maximum investment or the maximum warehouse space must be limited, instead of the average, Page and Paul (1976) propose a grouping procedure. 'The equal order interval method' assures that all the orders are not replenished at the same time and then the initial composition is maintained. It outperforms the Lagrangian approach. If the Lagrangian method is used to limit the maximum use of a resource, a 'normalizing factor' should be applied, this factor lies between 0.5 and 1 and the maximum allowed is multiplied with this factor. Goyal (1978) improves this heuristic making use of order phasing and a basic replenishment cycle. Rosenblatt (1981) explains that neither of these methods, Lagrange or grouping, really finds the optimal value due to simplifications in both formulations. Rosenblatt and Rothblum (1990) shift towards a penalty like method, where extra capacity can be bought. Puerto and Fernandez (1998) construct a multi-objective problem using a Pareto-optimal approach. This is a way of doing global sensitivity analysis on the solution space. Haksever and Moussourakis (2005) propose a mixed integer programming model to deal with multiple linear constraints while making use of piecewise linear approximations. The model chooses between an independent or a fixed cycle approach. Test problems with up to 30 items are solved. Boctor (2010) introduces a new mathematical formulation and an efficient heuristic for this inventory replenishment staggering problem, which is NP-hard (non-deterministic polynomial-time hard). The replenishment cycles must be integer multiples of a basic cycle. Examples with up to 200 items can be solved approximately within seconds while outperforming previous heuristics with 11% better results.

Conclusion: closed-form expressions with sensitivity analysis can be created for the multi-product inventory problem with deterministic constant demand and an aggregate average resource limitation. Transforming the aggregate average resource limitation into an aggregate maximum resource limitation yields a more complex problem. During the last four decades approximation algorithms were developed and are now able to find high

quality solutions for this staggering problem with a maximum capacity limit.

2.4.2 Newsvendor model with a single constraint

Let $J \geq 2$, $N = 1$, $m = 1$, $l_j = 0$, $x_j \in \mathbb{R}$. The decision variable in this model is Q_j, the order quantity for each item. A newsvendor problem is a single period model with a stochastic demand and penalty costs for ordering too much (c_{oj} for each item ordered but not sold) or too little (c_{uj} for each demand that cannot be satisfied). See section 2.3.5 for the single item model. The stochastic demand has a normal distribution $\varphi(z)$, see (2.185a). Here we consider the newsvendor problem in a multi-product environment with one volume capacity constraint, w_j is the volume of one unit of item j and e is the total available volume, see (2.185b). In the work of Hadley and Within (1963) one can end up with negative order quantities and service levels in case of a very tight capacity constraint for the problem (2.185). Hence a lower bound on the order quantity needs to be added: $l_j = 0$, as was discovered by Lau and Lau (1996). They extend the method so it can handle general demand distributions making use of the Lagrangian method. The procedure to solve this problem is rather complicated.

$$Minimize\, f(x) = \sum_{j=1}^{J} \left(c_{oj} \int_{-\infty}^{Q_j} (Q_j - x)\, \varphi(z_{(x)j}) dx + c_{uj} \int_{Q_j}^{\infty} (x - Q_j)\, \varphi(z_{(x)j}) dx \right) \tag{2.185a}$$

$$Subject\ to\ g(x) = \sum_{j=1}^{J} w_j Q_j \leq e \tag{2.185b}$$

In an alternative approach using deterministic optimization by Vairaktarakis (2000) uncertainty is described using interval and discrete demand scenarios. Algorithms are applied for min-max regret objectives to obtain optimal solutions under the defined conditions. Abdel-Malek and Montanari (2005b) further analyze the phenomenon of the lower bounds and divide the solution space in three regions: a non binding constraint region, a binding constraint region where each product can be bought and finally a region with a very strict constraint resulting in zero order sizes for some products. An iterative Lagrangian based method is used with an approximation of the cumulative distribution. Zhang et al. (2009) continue on this work and develop a solution algorithm using a binary search procedure which provides near optimal solutions for a continuous demand distribution and a good approximate solution for discrete demand. Multiple aggregate constraints are

considered as future research. In a 5 item discrete demand test the reached solution has a gap of 2.2% with the optimal cost, while budget constraint violation is on average 5%, in case of violation. Abdel-Malek et al. (2008) expand the scope of this problem by integrating a random yield. They refer to this problem as the Gardener Problem: a gardener has a limited acreage and must divide this over several possible crops, while demand and yield of the crops is uncertain. An exact solution is reached in case of uniform distribution and an approximate solution in case of other distributions. A 5 item example is demonstrated and validated through simulation.

Conclusion: It is possible to incorporate a capacity constraint and find optimal solutions for a multi-product newsvendor problem. The pitfall to end up with negative order quantities or service levels must be prevented in case of a very binding capacity limit. Adding additional uncertainties or working with discrete demand complicates the problem, but approximate solutions are available for small sized problems.

2.4.3 Newsvendor model with multiple constraints

Let $J \geq 2$, $N \geq 2$, $m = 1$, $l_j = 0$, $x_j \in \mathbb{R}$. Having more than one ($N \geq 2$) aggregate constraint converts our multi-product newsvendor problem in a considerable more difficult problem, see (2.186). In a two constraint example it is assumed that w_{1j} is the volume of one unit of item j and e_1 is the total available volume, w_{2j} is the purchase cost of one unit of item j and e_2 is the available budget. The primary purpose of Lau and Lau (1996) was to deal with this multi-constraint problem. Using a Lagrange method is only possible if one knows the active constraints, the ones that are binding. Only these constraints can be withheld in the Lagrange function. During the iterations for finding the correct values for the Lagrange variables, some constraints may be activated while others are deactivated. They develop a procedure to manage the pool of active constraints and the primal problem is converted into a dual problem, because typically there are a huge number of items but only a small number of constraints. The algorithm performance is linked to the number of constraints and the tightness of these constraints. A problem with 1.000 items and 20 constraints is solved within seconds.

$$Minimize\ f(x) = \sum_{j=1}^{J} \left(c_{oj} \int_{-\infty}^{Q_j} (Q_j - x)\, \varphi(z_{(x)j}) dx + c_{uj} \int_{Q_j}^{\infty} (x - Q_j)\, \varphi(z_{(x)j}) dx \right) \tag{2.186a}$$

$$Subject\ to\ g_n(x) = \sum_{j=1}^{J} w_{nj} Q_j \leq e_n, \quad n = 1, .., N \tag{2.186b}$$

Abdel-Malek and Montanari (2005a) examine the dual of the solution space with two constraints. An important feature of their approach is its applicability to general probability distribution functions, while it yields an optimum or near optimum solution with a known pre-set error. A 4 item example is solved within 4 iterations with an error on the constraints of less than 0.05%. Abdel-Malek and Areeratchakul (2007) propose a quadratic programming approach, enabling the use of available software packages so that lower bounds and multiple constraints are no longer an issue. This software can also work with large data sets and offers sensitivity analysis. An example from Lau and Lau (1996) with 7 products and 5 constraints, but now solved with the quadratic programming approach, gives nearly the same cost while using familiar software instead of a specific algorithm. Niederhoff (2007) uses separable programming. The simplex method is used to find solutions for nonlinear programs where the objective function and the constraint functions are the sum of functions and each function involves only one variable. A 10 product example is given, but larger problems form no issue for the software. Zhou et al. (2008) introduce a risk factor in this problem defining a CVaR (conditional value at risk) aggregate constraint that represents a loss function of a portfolio. It is shown that the CVaR model can be represented as a linear program through approximation of the demand density function. A 10 item example is solved and analyzed. Özler et al. (2009) use VaR (value at risk) to limit the risk of earning less than a desired target. The VaR constraint is an approximation of the total profit of different products with independent demand and a Normal distribution. A non-linear solver is used to solve the case with up to 50 items.

Conclusion: although the multi-product newsvendor problem with multiple aggregate constraints is much harder to deal with than the problem with one aggregate constraint, it better represents practical situations. Real life problems can be solved with specific, rather complex, algorithms. Approximations in problem formulation can incorporate risk factors but enable the use of familiar linear, quadratic or non-linear software packages. Here we only have encountered small examples and are still missing real life cases.

2.4.4 Base-stock models

Let $J \geq 2$, $N = 1$, $m = 1$, $x_j \in \mathbb{Z}$. The decision variable is the target stock level s_j. Within a spare parts environment it is assumed to have small demand rates and high unit purchase costs, inducing a base-stock policy. This is an (r, Q) model where $Q = 1$ and $r = s - 1$. See section 2.3.4 for

the single item model. As the unit purchase cost (c_j) is much larger than the ordering cost (k_j), the ordering cost is considered negligible and not withheld in the formulation. Sherbrooke (2004) uses a system approach on a set of 1 414 spare parts, resulting in a 46% reduction in inventory investment without a decrease in performance. The goal is to minimize the investment in spare parts, see (2.187a), while the time-weighted backorders should be smaller than a predefined aggregate non-service level (2.187b). G_j^1 is the first order Poisson loss function, see (2.7) and the time weighted backorders are as defined for a single item model, see (2.102). The problem with a Poisson demand distribution is solved using marginal analysis. It considers the decrease in backorders by adding one unit to the target stock level, while comparing with the cost of adding one unit for each item.

$$Minimize\ f(x) = \sum_{j=1}^{J} c_j s_j \tag{2.187a}$$

$$Subject\ to\ g(x) = \sum_{j=1}^{J} G_j^1(s_j) \leq e \tag{2.187b}$$

Thonemann et al. (2002) quantifies the expected improvement in case of a system approach using only a single parameter representation of the unit cost and average demand skewness over all parts. He integrates a time weighted fill rate constraint. Systems with high unit cost skewness profit most from a system approach. Using a 400 item data set it is shown that a high cost skewness, typical in spare parts, gives an improvement between 13% and 25%.

Hill and Pakkala (2007) minimize the cost that includes holding, backorder and order fill rate costs. The order fill rate is the probability that a customer order can be satisfied entirely and immediately from stock, this is relevant in a retail system to prevent extra shipping costs. Through an iterative procedure an approximate solution is reached. A problem with 2.187 items is solved in seconds. Future research to this work can focus on a compound Poisson demand process. Kranenburg and Houtum (2007) diversify the target aggregate fill rate over groups of items while commonality exists between groups and a shared stock is used. A heuristic provides a lower bound and an approximate solution. In a case study with 2 groups of 700 items on average 6% can be saved in spare parts provisioning costs and it takes 13 seconds to run this model.

Conclusion: a base stock policy is typically applied in spare parts environments. A system wide service level constraint of e.g. 97% can be reached by attaining this service level with each item. Differentiating the service level

over the items can reduce the costs significantly, with cost benefits ranging from 6% up to 46% in the given real life cases.

2.4.5 (r, Q) inventory system

Let $J \geq 2$, $N = 1$, $m = 2$, $x_j \in \mathbb{R}^2$. The two decision variables ($m = 2$) for each item are the reorder point r_j and its order quantity Q_j. In an (r, Q) policy an order of size Q is placed as soon as the inventory position falls to or below the reorder point r. See section 2.3.6 for the single item model. Hadley and Within (1963) touch this multi-product problem with aggregate constraint and stochastic demand. An iterative procedure is proposed to find the appropriate Lagrange multiplier. The goal is to minimize the cost, see (2.188a). The cost has four components in this case: the ordering cost (k_j), a holding cost (h_j), a shortage cost per unit and time unit (b_j) and a shortage cost per unit (p_j). The demand rate is λ_j and the average demand during leadtime is ν_j. Φ^1 and Φ^2 are the first and second order standard normal loss functions, see also (2.41) and (2.42) for the definitions. In section 2.3.6.3 the single-item (r, Q) model with normal demand is discussed in more detail. The expected number of orders, with a direct relation to the necessary workforce, has an upper limit, see (2.188b). Although a general principle is provided, they do not provide an efficient procedure to deal with the inter-dependencies between r and Q or how to add lower bounds for the reorder points ($r_j \geq l_j \geq 0$). As such it is too time consuming and simplifications are needed for the problem to be tractable.

$$\begin{aligned} Minimize\, f(x) = \sum_{j=1}^{J} \Bigg[& \frac{\lambda_j k_j}{Q_j} + h_j \left(\frac{Q_j}{2} + r_j - \nu_j \right) \\ & + \left(\frac{h_j + b_j}{Q_j} \right) \left(\Phi^2(z_{(r)j}) - \Phi^2(z_{(r+Q)j}) \right) \\ & + \frac{\lambda_j p_j}{Q_j} \left(\Phi^1(z_{(r)j}) - \Phi^1(z_{(r+Q)j}) \right) \Bigg] \end{aligned} \tag{2.188a}$$

$$Subject\, to\, g(x) = \sum_{j=1}^{J} \frac{\lambda_j}{Q_j} \leq e \tag{2.188b}$$

Ghalebsaz-Jeddi et al. (2004) extend this model and explore the impact of paying purchasing costs (k_j) when orders arrive and not when the order is placed. They assume a Normal distribution for the total budget and use an approximate formulation for the expected shortage. It must be pointed out that the approximation neglecting $\Phi^2(z_{(r+Q)j})$ or $\Phi^1(z_{(r+Q)j})$ may perform poorly in many situations, see Zipkin (1986a). Unless leadtime demand is

quite regular and Q is known to be large, neglecting the specified terms may give misleading results, for more details on this see section 3.3.3. Next they introduce linear and quadratic piecewise approximations. A 2 item example gives approximate results, but needing less iterations than the Hadley and Within (1963) method. Additional constraints are considered as future work. Bera et al. (2009) transform this problem in a multi-objective optimization format using a fuzzy chance-constrained technique and surprise function. A minimax distribution free procedure is applied to solve this problem and a 2 item numerical example is also solved. Zhao et al. (2007) consider the problem where demand is according a renewal process with customer demands of one unit, e.g. Poisson. The aggregate constraint is the sum of maximum storage for each item separately, as each item will have a fixed location space. An algorithm with polynomial time computation complexity finds the optimal solution and it is tested on an example with 30 items. The extension towards aggregate constraints with commonly resources, such as budget, is seen as future research.

Conclusion: even with the immense popular (r, Q) policy, with two decision variables per item, it is possible to imply aggregate constraints. The Lagrangian principle is still applicable, but the interdependency of r and Q increase the computing complexity during each iteration. We must notify that simplifications, to reduce the inter-dependencies between r and Q, may give misleading results. Specific techniques are applied to cases with reduced complexity or to small sized problems. We do not see real life cases solved in full precision yet.

2.4.6 (r, Q) systems without marginal costs

Let $J \geq 2$, $N \geq 1$, $m = 2$, $x_j \in \mathbb{R}^2$. Marginal cost information for inventory models is not likely available in practice. So here we look into models that do not use the following marginal cost components: order cost (k_j), holding cost (h_j) or shortage costs (a_j, b_j or p_j). The demand rate is λ_j and the average demand during leadtime is ν_j. Φ^1 is the first order standard normal loss function, see also (2.41) for the definition. Gardner and Dannenbring (1979) minimize the non-service (2.189a), while satisfying the aggregate budget (2.189b) and workforce constraints (2.189c), see section 2.3.6.3 for the definition of these KPI's. This is visualized as an optimal policy surface. The formulations for service level and average inventory are approximate formulations, here as well $\Phi^2(z_{(r+Q)j})$ and $\Phi^1(z_{(r+Q)j})$ are neglected to simplify the solving algorithm. For the problem with one constraint an iterative procedure is used to find the Lagrange variable. With two constraints, (2.189b) and (2.189c), the iterative search for the Lagrange variables is more complex as there is interdependency between r_j and Q_j. An application of this

technique on a sample of 78 180 items shows that workload can go down 25% and service increases with 1% to 6% without increasing the necessary investment.

$$Minimize\, f(x) = \sum_{j=1}^{J} \frac{\lambda_j}{Q_j} \left(\Phi^1(z_{(r)j})\right) \tag{2.189a}$$

$$Subject\, to\, g_1(x) = \sum_{j=1}^{J} \left(\frac{Q_j}{2} + r_j - \nu_j\right) \leq e_1 \tag{2.189b}$$

$$g_2(x) = \sum_{j=1}^{J} \left(\frac{\lambda_j}{Q_j}\right) \leq e_2 \tag{2.189c}$$

Schrady and Choe (1971) minimize the long term time weighted backorders while respecting a system budget constraint, although a simplification was made to both formulations. A first solution approach finds the Wilson order sizes and iteratively calculates the reorder levels making use of a Lagrangian function. For multiple constraints an exterior penalty function method is proposed, see Bazaraa et al. (2006). Lenard and Roy (1995) group the different items in families. For each family an aggregate item is chosen drawing efficient policy surfaces, based upon simulation. This approach prevents unacceptable shortage levels for a number of items, which is a possible result in case the aggregate service level is formulated as an arithmetic mean of the individual service levels. Hopp et al. (1997) aim at minimizing the aggregate inventory investment while satisfying a maximum order frequency and a minimum service level in case of batch demands, thus implying integer variables. They present three Lagrangian heuristics for approximating the inventory performance measures. The two simpler heuristics are closed-form expressions, but can perform poorer in cases of low service level and low order frequency. A practical case proves the necessity for lower bounds on the reorder level. An implementation of this heuristic in a 30 000 item system gives a 20% inventory investment reduction for comparable service levels.

Conclusion: in case of using non-marginal costs in the popular (r, Q) policy for multi-product problems we see the use of simplifications for service level and inventory performance measures, with known risks. Even with these limited accuracy simplifications a significant monetary benefit speaks in favor of a system approach. In the mentioned examples the cost reductions range from 20% till 25% with a system approach versus an item approach. It should be highlighted that additional lower bounds might be needed to assure minimal service levels for all the products and to prevent negative reorder points.

2.4.7 (s, S) inventory systems

Let $J \geq 2$, $N = 1$, $m = 2$, $x_j \in \mathbb{R}^2$. The two ($m = 2$) decision variables are the reorder point level (s_j) and the order up to level (S_j). In an (s, S) policy an order is placed to reach the stock maximum level S as soon as stock falls to or below reorder point s. See section 2.3.7 for the single item model. Mittchel (1988) developed an algorithm based upon an approximation of demand for the periodic review (s, S) problem in a multi-product environment with a service constraint. The goal is to minimize the cost that has two components: holding cost (h_j) and ordering cost (k_j), see (2.190a). M_j is the Renewal function of Φ_j, the standard normal distribution, and Φ_j^* is the (L+1)-fold convolution of Φ_j with itself, L is leadtime. There is an aggregate constraint to achieve a system wide ready rate service level, the fraction of periods without stock-outs, see (2.190b). The service constraint is not weighted by demand. Operating costs can be reduced significantly when a uniform service model, where each item has the same service level, is no longer used. For a 32 item example with a system service level of 85%, cost reductions between 20% and 39% are achieved.

$$Minimize\ f(x) = \sum_{j=1}^{J} \left[h_j \left(\int_0^{S_j - s_j} \int_{-\infty}^{S_j - y} (S_j - y - x)\, d\Phi_j^*(x) dM_j(y) + \int_{-\infty}^{S_j} (S_j - x)\, d\Phi_j^*(x) \right) + k_j \right] / [1 + M_j(S_j - s_j)] \tag{2.190a}$$

$$Subject\ to\ g(x) = \sum_{j=1}^{J} \left[\int_0^{S_j - s_j} \Phi_j^*(S_j - y) dM_j + \Phi_j^*(S_j) \right] / [N\,(1 + M_j(S_j - s_j))] \geq e \tag{2.190b}$$

Schneider and Rinks (1989) use asymptotic properties from the renewal theory to approximate the optimal solution. Besides the service constraint, two other aggregate constraints are added: one on maximum workload and another on maximum storage room. An iterative grid search is performed for the Lagrange multiplier values. For a 100 item system the results are visualized in an optimal policy surface chart that shows the tradeoff between cost and service level, while satisfying the workload and storage constraint. Cohen et al. (1992) add shortage costs and use a demand weighted fill rate constraint. There is no backlogging, so unmet demand is lost. A greedy algorithm is developed to find near-optimal solutions for the approximate

problem formulation. It is tested on a 4 item part and shows a small error of 3% in case of low service levels, but larger errors up to 17% in case of high service levels compared to a lower bound value.

Conclusion: an order up policy (s, S) is also often used in real life and goes often under the name of a min-max system. Here as well we see a significant cost reduction potential when a system approach is chosen over an item approach. So although exact solutions cannot be reached, one should not wait to already profit from a system approach. Cost benefits lie between 20% and 39% in the examples given. Allowing some approximations it is possible to integrate multiple constraints or shortage costs, but the errors can no longer be neglected then.

2.5 New contribution

We analyzed and annotated some recent and relevant multi-item inventory problem references grouped into five categories: deterministic constant lead-time demand, news vendor, base-stock, (r, Q) and (s, S) policy. We investigated the proposed model formulations, the algorithmic approaches and benefits of a system approach versus an item approach. We also highlighted the limitations from a practical viewpoint of these models, see contribution 1.

2.6 Conclusions

Poisson and compound Poisson processes form a very good representation of inventory demand. However in practice these are not used often and are replaced frequently by the normal distribution. Due to the central limit theorem Poisson and compound Poisson merge towards the normal distribution in case of large volumes. For smaller demands or for larger variances we saw that the differences cannot be neglected.

When optimizing an inventory problem we need to compute thousands of times the first and second order loss functions of the demand distribution and also their inverse functions. The Poisson, compound Poisson and normal distribution have no closed-form expressions for these functions. So there is a need for accurate and efficient approximation functions. The Remez algorithm is a stable minimax approach to approximate a continuous function as the ratio of two polynomials.

Although the knowledge on single-item inventory models is vast and the foundations have been laid in the sixties and before, we establish a great unawareness in practice. The value of the revision of these fundamentals cannot be overestimated.

We see a practical need for a system approach, rather than an item approach. This would enable managers to realize their goals with an optimal mix between cost and service, while confronted with limited resources such as workspace, workforce or investment. In the last decades we find some research on this topic, although limited and not suited yet for use in practice. It is our aim to close this gap in the next chapters.

3

Single item inventory models and approximation error analysis

We revisit two service levels. We discuss the relevance of replenishment rate, S_1. Next we provide formulations for the order line service level S_{OL} in case of compound Poisson demand. We highlight that the normal demand (r, Q) policy performance measure definitions are not completely correct. We also point out that simplified formulations may yield huge errors, here we provide the conditions to safely use these simplifications.

We demonstrate and analyze the errors when the normal demand model is used to approximate the Poisson or compound Poisson demand inventory model. For each inventory KPI we develop an error reduction function and analyze the improvement and remaining gap.

3.1 Replenishment rate (S_1) relevance

In many textbooks only the very simple service level replenishment rate S_1 (2.67) is explained, because it is very easy to calculate in case of normal demand. S_1 is the specified probability of no stockout per replenishment cycle. Although easy to calculate in case of normal or Poisson demand, it has little value in practice. S_1 is not important for the company holding the stock and even less for the client. S_1 is independent of the order quantity Q, while S_2, S_3 and S_{OL} all increase with increasing Q, if r is unchanged, in case of an (r, Q) policy. From the point of view of completeness we also added it to the key performance indicators for normal and Poisson demand. As it is quite complex to compute in case of compound Poisson and as it has no practical meaning, we did not work it out for compound Poisson demand. Through an example we will demonstrate the uselessness of the S_1 service level.

Example 3.1 S_1 no practical sense as independent from Q

In Example 3.1, Table 3.1, we take two situations with the same demand over a period of 240 days. We apply an (r, Q) policy and set $r = 900$ in both cases, but in the first we have $Q = 800$, while $Q = 1600$ in the second case. We take this example to demonstrate the uselessness of S_1. In both cases we have $S_1 = 50\%$: with $Q = 800$ 6 out of 12 times the net inventory (IN) was negative at the moment of replenishment. With $Q = 1600$ this is 3 out of 6. In table 3.1 we can see how the other service levels (S_2, S_3 and S_{OL}) increase as the order quantity increases. Also in Figure 3.2 we can see there is a better service level, compared with figure 3.1. So we can conclude that S_1, although easy to calculate sometimes, has no real value to express a service level, due to these two reasons:

- S_1 is not depending on the order quantity Q
- The intrinsic definition of S_1 (2.67) has no value for the client or for the company holding the inventory

These two argument are valid for S_1, but not for the other service levels:

- S_2, S_3 and S_{OL} increase with Q, see table 3.1, e.g. S_{OL} increases from 85% up to 91% as Q increases from 800 up to 1600
- In Example 3.1 we see $S_1 = 50\%$, while the three other service levels are all $> 90\%$ in case of $Q = 1600$. In Figure 3.2 we can also see that 50% is not a good representation of the service level as experienced by the client or the company holding the stock

Input	Value	a: $Q = 800$	Value	b: $Q = 1600$	Value
λ	10000	S_1	50%	S_1	50%
ν	833	S_2	87%	S_2	92%
L	0.083	S_3	88%	S_3	94%
r	900	S_{OL}	85%	S_{OL}	91%

Table 3.1: Example 3.1: Service levels with $Q = 800$ and $Q = 1600$

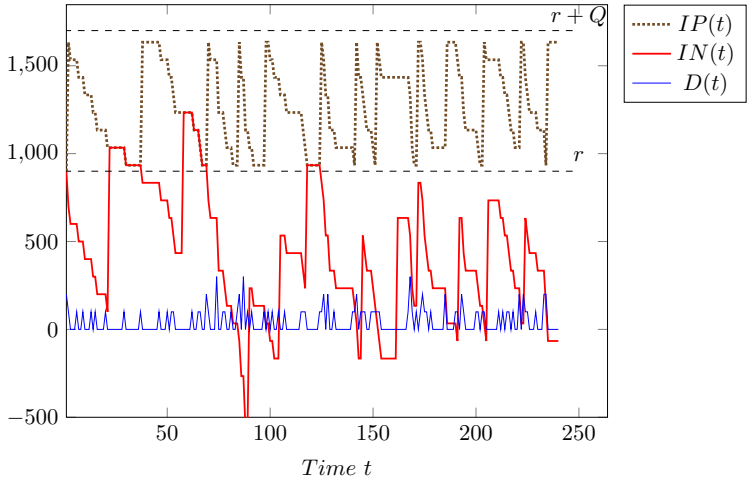

Figure 3.1: Example 3.1a: Stockouts with $Q = 800$

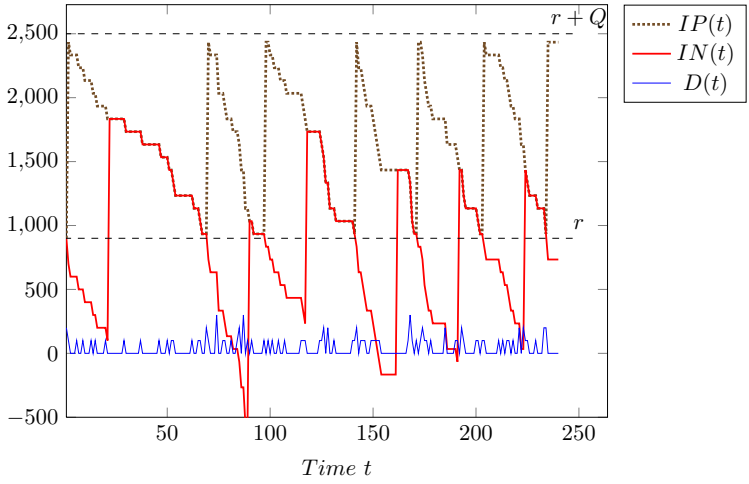

Figure 3.2: Example 3.1b: Stockouts with $Q = 1600$

3.2 S_{OL} order line service level

3.2.1 Base-stock order line service level

Contribution 2: Order line service level for a base-stock policy

Boylan and Johnston (1994) gave insights in the relationships between six different frequently used service measures. Order line service level is one of them, but no direct equations for computation are given. We deduct the formula for the order line service level S_{OL} from the S_2 equation structure. For S_{OL} we are only interested in the order lines that can be fully satisfied. In this definition an order line of quantity $d = 1$ has the same weight as an order line with quantity $d = 100$, which is definitely not the case in the S_2 definition. As such we can drop $min(d, k)$ in the numerator and χ in the denominator of (2.116). As it is not a 'weighted' formula, this reduces the denominator to 1. As we only take into account the fully delivered order lines, we only look at the IN positions where $IN \geq d$. This yields (3.1), see also De Schrijver et al. (2012).

$$\begin{aligned} S_{OL} &= \frac{\textit{Number of order lines delivered out of stock in full}}{\textit{Total number of order lines}} \\ &= \sum_{d=1}^{min(\eta,s)} \sum_{k=d}^{s} f_d \, Pr(IN = k) \\ &= \sum_{d=1}^{min(\eta,s)} \sum_{k=d}^{s} f_d \, g_Y(s-k) \end{aligned} \tag{3.1}$$

Example 3.2 Base-stock compound Poisson demand

Example 3.2 in Table 3.2 is based upon Poisson Example 2.10, see Table 2.7. The only difference is that we have a compound Poisson demand. We used the compound Poisson definition from Example 2.4, Table 2.3, where we have $\chi = 2.3$. In order to keep $\lambda = 20$ and $\nu = 2$, we have set $\tau = 8.7$. The optimal target stock level is $s^* = 2$, see Figure 3.3, just like in the previous Poisson Example 7.

The total variable cost C_v on the other hand is nearly the double, due to the greater demand variability. The set of KPI's is also completely different in Example 2.10 and 3.2. Amongst others there is a clear distinction between the service level definitions: $S_3 = 56\%$, $S_2 = 35\%$ and $S_{OL} = 31\%$. As explained previously the stricter S_{OL} service level has a lower value than the fill rate S_2 in case of compound Poisson demand.

Input	Value	Input	Value	KPI	Value
λ	20	χ	2.3	s^*	2
h	32	f_1	0.4	C_v^*	80.69
k	0	f_2	0.2	$\overline{I}$	0.98
b	50	f_3	0.1	$\overline{B}$	0.98
L	0.1	f_4	0.3	$\overline{A}$	0.44
ν	2	η	4	$\overline{OF}$	20
σ^2	6	a	0	S_2	35%
τ	8.695652	p	0	S_3	56%
				$\overline{P}$	12.90
				S_{OL}	31%

Table 3.2: Example 3.2: Base-stock compound Poisson demand

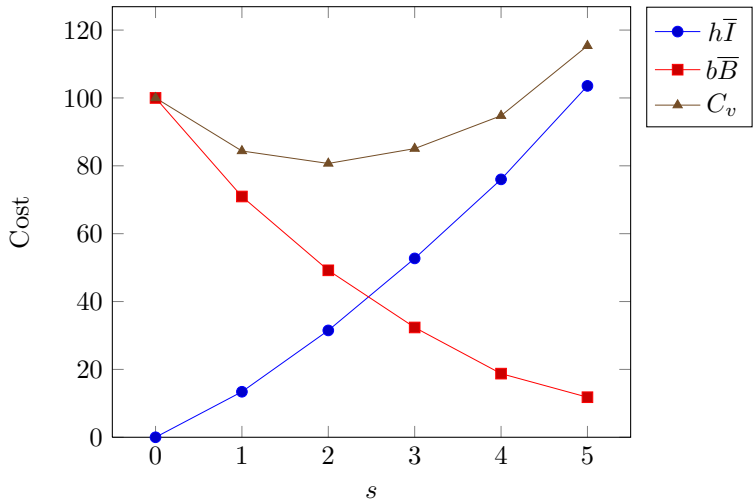

Figure 3.3: Example 3.2: Base-stock cost, compound Poisson demand

3.2.2 (r, Q) order line service level

Contribution 3: Order line service level for an (r, Q) policy

In practice we see that a normal demand fill rate is almost always used to mimic the order line service level. We now show here we can accurately calculate the order line service with (3.2). This definition is deducted from the fill rate definition (2.161), but within an order line definition the complete order needs to be fulfilled, which of course is more severe. So we can remove the denominator, as the size of the average order line has no impact on the definition, in the numerator we thus can also drop $min(d, k)$, the size of this specific order, see also De Schrijver et al. (2012).

$$S_{OL} = \sum_{d=1}^{min(\eta, r+Q)} \sum_{k=d}^{r+Q} f_d \, Pr(IN = k) \tag{3.2}$$

Example 3.3 (r, Q) **compound Poisson demand**

In Example 3.3, see Table 3.3 we continue on Example 3.2 (Table 3.2), but as in the previous example we adjust $k = 80$, $b = 100$ and $L = 0.2$.

Input	Value	Input	Value	KPI	Value
λ	20	χ	2.3	r^*	0
h	32	f_1	0.4	Q^*	14
k	80	f_2	0.2	C_v^*	339.10
b	100	f_3	0.1	$\overline{I}$	4.35
L	0.2	f_4	0.3	$\overline{B}$	0.85
ν	4	η	4	$\overline{A}$	0.28
σ^2	12	a	0	$\overline{P}$	7.08
τ	8.70	p	0	$\overline{OF}$	1.43
				S_2	65%
				S_3	72%
				S_{OL}	63%

Table 3.3: Example 3.3: (r, Q) compound Poisson demand

In Figure 3.4 we give a 3D representation of the variable cost C_v and we also indicate the minimum cost at $r^* = 0$ and $Q^* = 14$. We can see there is a broad area around this minimum where the cost impact is minimal. In Figure 3.5 we show three cost curves C_v for respectively $Q = 10$, $Q = 14$ and $Q = 18$ and where $r \in [-4, 4]$.

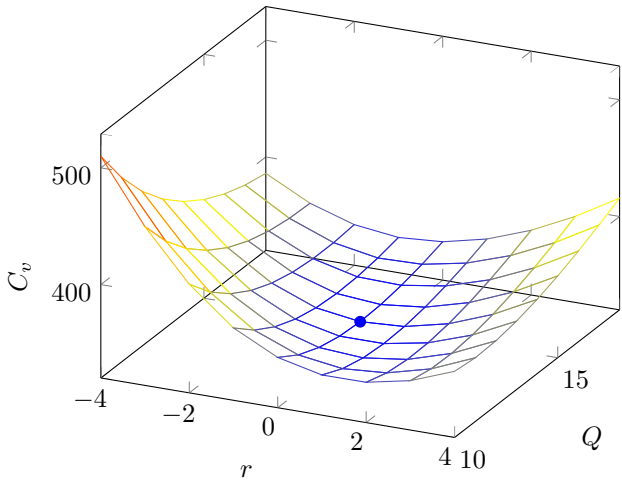

Figure 3.4: Example 3.3: (r, Q) cost, compound Poisson demand (3D)

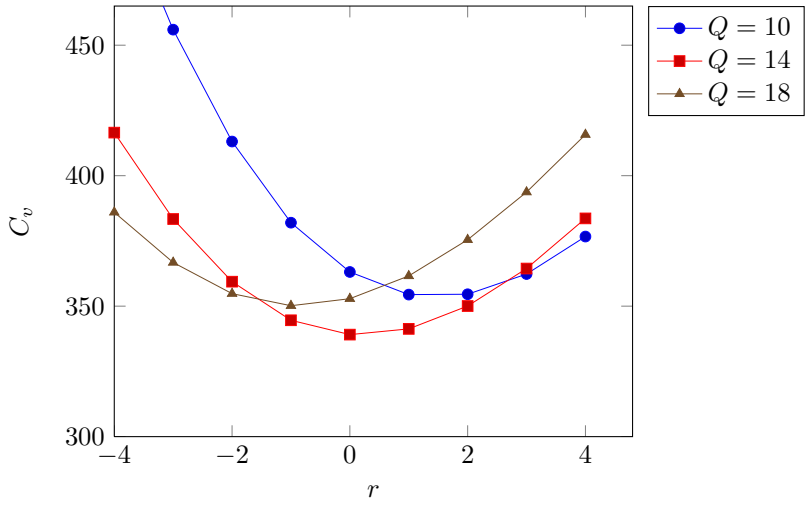

Figure 3.5: Example 3.3: (r, Q) cost, compound Poisson demand

3.2.3 (s, S) order line service level

Contribution 4: Order line service level (s, S)

We can reuse the (r, Q) definition (3.2) to formulate the order line service level in case of an (s, S) policy. The major change is the IN definition, (2.176) in the equation (3.3), see also De Schrijver et al. (2012).

$$S_{OL} = \sum_{d=1}^{min(\eta,S)} \sum_{k=d}^{S} f_d \, Pr(IN = k) \tag{3.3}$$

Example 3.4 (s, S) **compound Poisson**

Example 3.4 in Table 3.4 retakes the input from the (r, Q) compound Poisson Example 3.3 in Table 3.3. The (s, S) optimal cost is slightly ($< 1\%$) better than the (r, Q) optimal cost.

Input	Value	Input	Value	KPI	Value
λ	20	χ	2.3	s^*	0
h	32	f_1	0.4	S^*	13
k	80	f_2	0.2	C_v^*	337.85
b	100	f_3	0.1	$\overline{I}$	4.31
L	0.2	f_4	0.3	$\overline{B}$	0.85
ν	4	η	4	$\overline{A}$	0.28
σ^2	12	a	0	$\overline{P}$	7.09
τ	8.70	p	0	$\overline{OF}$	1.43
				S_2	65%
				S_3	72%
				S_{OL}	62%

Table 3.4: Example 3.4: (s, S) *compound Poisson demand*

3.3 Normal demand (r, Q) corrections

3.3.1 Corrected (r, Q) normal demand KPI equations

Contribution 5: Corrected (r, Q) ***KPI equations***

Comparing the result of the formula for average inventory $\overline{I}$ for base-stock (2.125) and for an (r, Q) policy (2.168), where we set $Q = 1$, resulted in different KPI values, especially for the average inventory. In Zipkin (2000) there is a small reference, although not applied in later formulas, that it is more correct to integrate over the range $[r+0.5, r+0.5+Q]$. If we set $Q = 1$ in equation (2.125) and (2.123), we see we integrate over the range $[r, r+1]$, so the average is $r + 0.5$. In 2.87 we have seen that $s = r + 1$. So if we add 0.5 and integrate over the range $[r + 0.5, r + 0.5 + Q]$ with $Q = 1$ we have an average of $r + 1$ which is equal to s. In Figure 3.6 we can clearly see that the formulas using the '+0.5' correction give a far better match with the Poisson exact KPI.

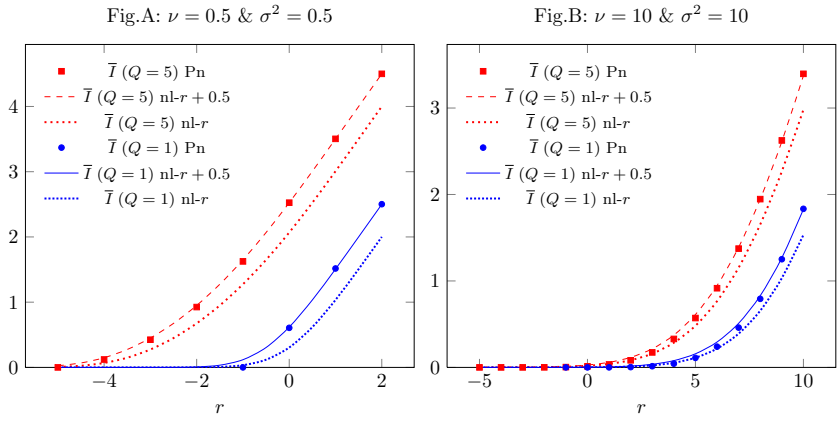

Figure 3.6: Impact of no 0.5 correction on r when approximating (r, Q) $\overline{I}$

In practice and also in most of the text books when using the normal approximation in an (r, Q) policy, integration is over the range $[r, r + Q]$ instead of $[r + 0.5, r + 0.5 + Q]$. This can make a large difference especially for small ν, see Figure 3.6. In this Figure 'nl-r' (normal demand - no 0.5 correction) refers to equation (2.168) without the 0.5 correction, while 'nl-$r + 0.5$' refers to (3.7). We can see that (2.168) leads to an underestimation of the approximated average inventory. The corrected (r, Q) normal demand equations become (3.4)-(3.10).

$$\begin{aligned} Pr(IN \leq x) &= \frac{1}{Q} \int_{r+0.5}^{r+0.5+Q} \Phi^0(z_{(u-x)}) du \\ &= \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5-x)}) - \Phi^1(z_{(r+0.5+Q-x)}) \right] \end{aligned} \tag{3.4}$$

$$\overline{A} = \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right] \tag{3.5}$$

$$\overline{B} = \frac{\sigma^2}{Q} \left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) \right] \tag{3.6}$$

$$\overline{I} = \frac{Q}{2} + r + 0.5 - \nu + \overline{B} \tag{3.7}$$

$$\overline{P} = \lambda \overline{A} \tag{3.8}$$

$$S_2 = S_3 = S_{OL} = 1 - \overline{A} \tag{3.9}$$

$$\overline{OF} = \lambda / Q \tag{3.10}$$

3.3.2 (compound) Poisson $\overline{A}$ approximation correction

In section 2.1.4.1 and 2.1.4.2 we have seen in Figure 2.11 that a better normal approximation is achieved for $G(k)$ and $G_Y(k)$ when using $\Phi(z_{(k+0.5)})$ instead of $\Phi(z_{(k)})$. So if we want to approximate the base-stock Poisson ready rate, $\overline{A} = G^0(s-1)$, see (2.101), with a normal distribution, this leads to $\overline{A} = \Phi^0(z_{(s-1+0.5)}) = \Phi^0(z_{(s-0.5)})$. When comparing with $\overline{A} = \Phi^0(z_{(s)})$, see (2.122), we thus need to make a '-0.5' correction. The reasoning is the same for ready rate in case of compound Poisson. For approximating $G^1(k)$ and $G_Y^1(k)$ this continuity correction is not necessary, see Figure 2.12. This means no correction needs to be made to the base-stock $\overline{B}$ equations. The (r, Q)-policy is an average of base-stock, so combining these findings with section 3.3.1 results in:

$$\overline{A} = \frac{\sigma}{Q} \left[\Phi^1(z_{(r)}) - \Phi^1(z_{(r+Q)}) \right] \tag{3.11}$$

$$\overline{B} = \frac{\sigma^2}{Q} \left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) \right] \tag{3.12}$$

$$\overline{I} = \frac{Q}{2} + r + 0.5 - \nu + \overline{B} \tag{3.13}$$

$$\overline{P} = \lambda \overline{A} \tag{3.14}$$

$$S_2 = S_3 = S_{OL} = 1 - \overline{A} \tag{3.15}$$

$$\overline{OF} = \lambda / Q \tag{3.16}$$

So the '0.5' correction from section 3.3.1 only needs to be made to $\overline{B}$ and not to $\overline{A}$, when the normal distribution is used to approximate the Poison or compound Poisson distribution, as this will give better approximating results. When we want to model a normal distribution demand, we should apply a '+0.5 correction' to $\overline{A}$ as well.

3.3.3 Conditions for simplified KPI normal demand (r, Q) expressions

Contribution 6: Conditions for simplified normal demand (r, Q) ***KPI's***

Within the existing literature we see that simplifications are applied on the KPI expressions. Simpler equations may yield larger errors. The impact of ignoring the second term based on $z_{(r+Q)}$ in equations (3.5) and (3.6) for $\overline{A}$ and $\overline{B}$ is in some cases negligible, but in other cases it can be up to 50% and more, as we will see in example 3.5.

We will investigate these conditions here. A reasonable measure of relative error when applying this simplification can be expressed as ϵ_1 and ϵ_2, see (3.17) and (3.18). The foundations for this analysis have been described by Zipkin (1986a). We use his findings and work them further out for the specific case of an (r, Q) replenishment policy in case of normal demand.

$$\epsilon_1(r,Q) = 1 - \frac{\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)})}{\Phi^1(z_{(r+0.5)})} = \frac{\Phi^1(z_{(r+0.5+Q)})}{\Phi^1(z_{(r+0.5)})} \tag{3.17}$$

$$\epsilon_1(r,Q) = 1 - \frac{\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)})}{\Phi^2(z_{(r+0.5)})} = \frac{\Phi^2(z_{(r+0.5+Q)})}{\Phi^2(z_{(r+0.5)})} \tag{3.18}$$

We state that a maximum relative error of $1e-3$ is acceptable in practice. In Figure 3.7 the region above the line $\epsilon_1 = 1e-3$ is where a simplified formula can be used. In figure 3.7 we can see that the condition for ϵ_1 is stricter than for ϵ_2. So it is sufficient to respect the conditions for ϵ_1. Through linear regression we can approximate the condition for an acceptable error ϵ_1 in two intervals: $z_{(r)} \in [-11, 3]$ and $z_{(r)} \in (3, 5]$. Both linear regressions in (3.19) have a high accuracy, the linear regression for the first interval has $R^2 = 99.99\%$ and in the second interval we have $R^2 = 99.98\%$. R^2 is a coefficient between 0 and 1 in a statistical analysis that allows the observer to see how effective one variable is at forecasting another variable. The higher the R^2 the more accurate the approximation is.

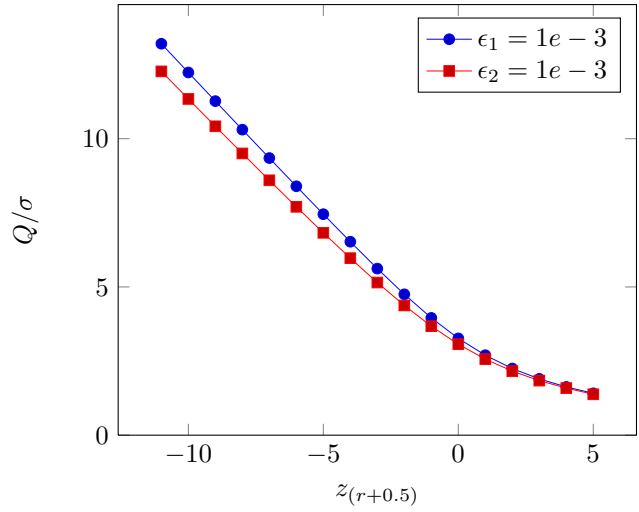

Figure 3.7: Conditions (ϵ_1 & ϵ_2) simplified (r, Q) $\overline{A}$ & $\overline{B}$ equations

$$\frac{Q}{\sigma} \geq -0.95406 z_{(r+0.5)} + 3.16766 \text{ if } z_{(r+0.5)} \text{ in } [-11, -3] \tag{3.19a}$$

$$\frac{Q}{\sigma} \geq 0.05323 z^2_{(r+0.5)} - 0.62941 z_{(r+0.5)} + 3.00697 \quad \text{if } z_{(r+0.5)} \text{ in } (-3, 5] \tag{3.19b}$$

$$\frac{Q}{\sigma} \geq 1.19073 \text{ if } z_{(r+0.5)} \text{ in }]5, +\infty] \tag{3.19c}$$

So when condition (3.19) is fulfilled one can reduce the definition of $\overline{A}$ to (3.20) and $\overline{B}$ to (3.21), while $\epsilon \leq 1e-3$.

$$\overline{A} = \frac{\sigma}{Q} \Phi^1(z_{(r+0.5)}) \tag{3.20}$$

$$\overline{B} = \frac{\sigma^2}{Q} \Phi^2(z_{(r+0.5)}) \tag{3.21}$$

When r and Q are not known yet and we need to include $\overline{A}$ or $\overline{B}$ in an optimization, we need to perform an additional step first to assure whether the simplified formulas can be used. A simple workaround is to determine a $Q = EOQ$ by using (2.62). As the order quantity Q will only increase due to additional penalty costs or service level constraints, we can see this as a safe approach. Establishing whether the simplified formulas can be used, demands a onetime check, if then the simplified equations can be used, this can be a big time saver in an iterative search while optimizing a system. The

relation between Q/σ and $z_{(r+0.5)}$ for $\epsilon_1 = 1e{-}3$ is given by (3.19a) - (3.19c). Each of these equations covers a region, the switching point between the first and second region is at $z_{(r+0.5)} = -3$ or $Q/\sigma = 5.3743$. We reformulate (3.19a) into (3.22) and (3.19b) into (3.23). So $z_{(r)}$ must exceed a minimum value that depends on its Q and σ.

$$\textit{if } \frac{Q}{\sigma} \geq 5.37430 \textit{ then } z_{(r+0.5)} \geq 3.46766 - \frac{Q}{0.95406\sigma} \tag{3.22}$$

$$\textit{if } 1.19073 \leq \frac{Q}{\sigma} < 5.37430 \textit{ then} \tag{3.23a}$$

$$z_{(r+0.5)} \geq \frac{0.62941 - \sqrt{0.21293(Q/\sigma) - 0.24412}}{0.10647} \tag{3.23b}$$

$$\textit{if } \frac{Q}{\sigma} < 1.19073 \textit{ , then no simplification allowed} \tag{3.24}$$

Example 3.5 Error due to simplified equations

In Example 3.5, see table 3.5, we take a case where the error would be extremely large if the simplified equation (3.20) would be used, $\overline{A} > 1$ and $S_2 < 0$, so leading to a negative fill rate service level! The simplified equation gives a fill rate of -27%, while it should be 81%, thus a difference of more than 100%! The ratio $Q/\sigma = 0.09 < 1.19073$, so the condition (3.24) is fulfilled and no simplification is allowed. This extreme example shows the danger of using the simplified equations when it is not allowed. If we respect the conditions (3.22)-(3.24), we are assured that the error will be acceptable in practical cases.

Input	Value	KPI	Simple eq. (2.162)	Full eq. (2.166)
ν	20	$\overline{A}$	127%	19%
σ^2	120	S_2	-27%	81%
r	29			
Q	1			
Q/σ	0.09			

Table 3.5: Example 3.5: (r, Q) Error due to simplifications

As a more general conclusion we can state that the (r, Q) normal demand simplified equations (3.20) and (3.21) lead to large errors in case of small Q, or more specified in case of small Q/σ. The conditions to apply the simplified equations with an acceptable error are given by (3.22)-(3.24). If these conditions are not met the equations (3.5)-(3.10) must be used.

3.4 (compound) Poisson demand normal approximation

In this section we will approximate the inventory models having Poisson, see (2.146)-(2.152), and compound Poisson demand, see (2.155)-(2.160), making use of an inventory model with a normal demand based upon the equations (3.5)-(3.10). We start from examples given previously and analyze the differences.

3.4.1 Poisson Base-stock normal approximation

As the normal distribution is a continuous distribution, we have no guarantee at all that the optimal target stock level s^* is an integer value, it can even be a negative value. In Example 3.6 in Table 3.6 we see that $s^* = 2.39$ with a minimal variable total cost $C_v = 44.59$. Example 3.6 is an approximation of the Poisson Example 2.10, see Table 2.7. In Figure 3.8 we can see there is only a little increase for the cost if $s = 2$ or $s = 3$ in Example 3.6.

Example 3.6 Base-stock normal demand, Poisson approximation

Input	Value	KPI	Value
λ	20	s^*	2.39
h	32	C^*	44.50
k	0	$\overline{I}$	0.78
b	50	$\overline{B}$	0.39
L	0.1	$\overline{A}$	0.39
ν	2	$\overline{OF}$	20
$a = p$	0	$\overline{P}$	13.21
σ^2	2	$S_1 = S_2 = S_3$	61%

Table 3.6: Example 3.6: Base-stock Normal demand, Poisson approximation

In Table 3.7 we compare the KPI's for Example 2.10, Poisson demand, and Example 3.6, an approximation with normal demand. For the normal demand we give the KPI's for the target stock level integer values surrounding the optimal target stock level s^*. We establish a good resemblance between the KPI's for Example 2.10 and Example 3.6 for $s = 2$. Only for the stock out frequency $\overline{A}$ and consequently also for the service levels, we have a larger deviation.

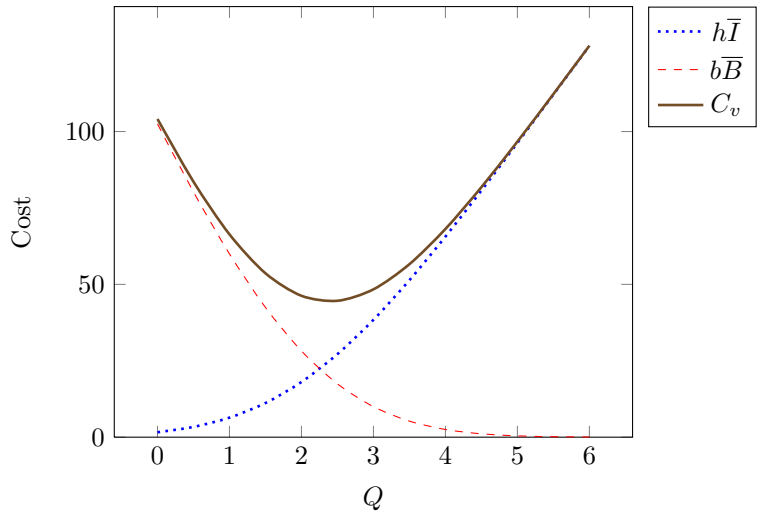

Figure 3.8: Example 3.6: Base-stock normal demand, Poisson approximation

KPI	Poisson Ex 2.10	normal Ex 3.6	normal Ex 3.6
s	2	2	3
C_v	44.39	46.26	48.37
$\overline{I}$	0.54	0.56	1.20
$\overline{B}$	0.54	0.56	0.20
$\overline{A}$	0.59	0.50	0.24
$\overline{P}$	11.88	10	4.80
$\overline{OF}$	20	20	20
$S_1 = S_2 = S_3$	41%	50%	76%

Table 3.7: Example 3.6: Base-stock KPI's Poisson and normal approximation

In the previous examples we have shown the exact inventory models for a Poisson and compound Poisson leadtime demand as well as the normal approximations for both. How accurate is the approximation? This is an important question as it will allow us to determine when to use the less complex normal approximation. We will now compare for a base-stock policy three important KPI's: average inventory $\overline{I}$, average backorders $\overline{B}$ and the stock out frequency $\overline{A}$. Here we did not include the average new shortages $\overline{P}$, because they are directly linked to $\overline{A}$. In Figure 3.9 we compare three situations for a Poisson approximation: $\nu = 0.5$, $\nu = 2$ and $\nu = 10$.

We see that the resemblance between both increases with ν. Only for small ν the normal approximation gives less accurate results, see Figure 3.9A for $\nu = 0.5$, especially for the stock out frequency $\overline{A}$, that directly determines

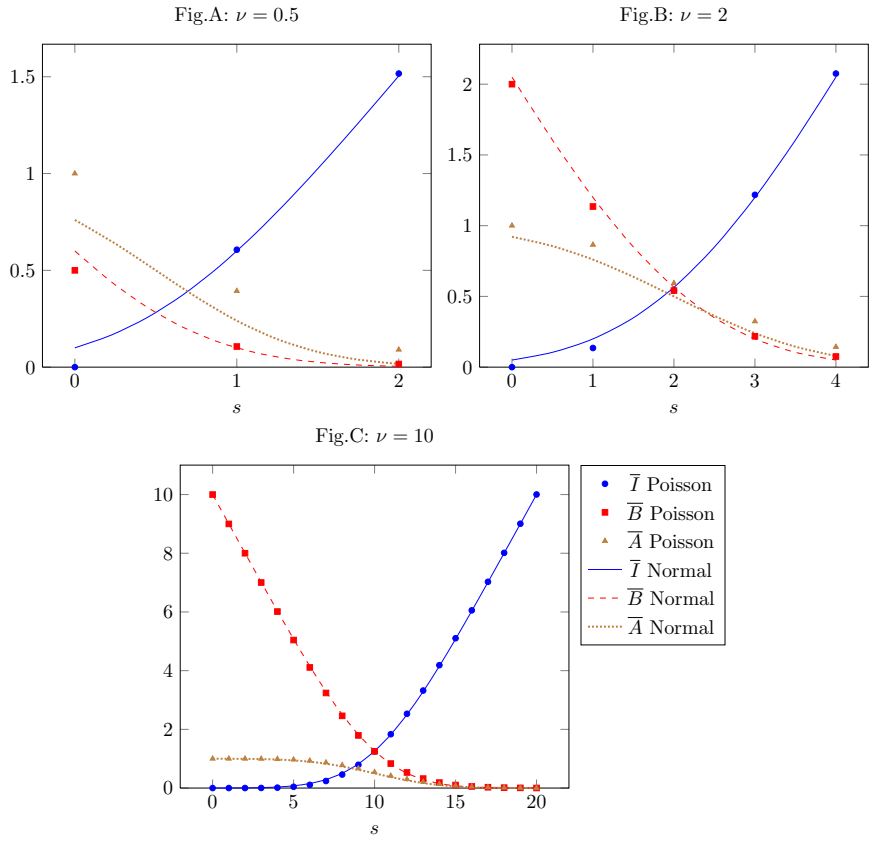

Figure 3.9: Base-stock $\overline{I}$, $\overline{B}$ & $\overline{A}$ for Poisson and normal demand

the normal approximation service levels in a base-stock policy. For larger ν, see $\nu = 10$ in Figure 3.9C, we see a very good match between the exact and the approximated KPI's.

As we have seen while comparing the distribution approximations, see section 2.1.4.1, the normal distribution can have negative demands, but we can now see here there is a limited impact on the KPI's. This impact increases with the ratio σ/ν, see (3.25). For a Poisson distribution this ratio is $1/\sqrt{\lambda L}$, for a compound Poisson this is even bigger: $\psi/(\lambda\sqrt{L})$.

$$Pr(D_L < 0) = \Phi^0(\frac{-\nu}{\sigma}) \tag{3.25}$$

3.4.2 Compound Poisson Base-stock normal approximation

Example 3.7 Base-stock normal demand, compound Poisson approximation

Example 3.7 in Table 3.8 is a normal approximation of the compound Poisson Example 3.2, see Table 3.2.

Input	Value	KPI	Value
λ	20	s^*	2.68
h	32	C_v^*	77.08
k	0	$\overline{I}$	1.36
b	50	$\overline{B}$	0.67
L	0.1	$\overline{A}$	0.39
ν	2	$\overline{OF}$	20
σ^2	6	$\overline{P}$	8.54
$a = p$	0	$S_1 = S_2 = S_3$	61%

Table 3.8: Example 3.7: Base-stock Normal demand, compound Poisson approximation

As expected the optimal target stock level is not an integer value $s^* = 2.68$, but again we can see in Figure 3.10 there is little cost impact when moving to the two neighboring integer values.

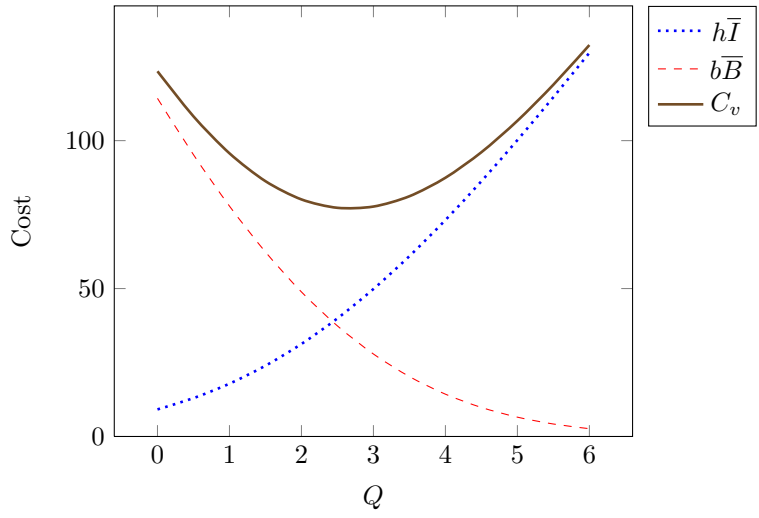

Figure 3.10: Example 3.7: Base-stock normal demand, compound Poisson approximation

In Table 3.9 we compare the compound Poisson Example 3.2 with the normal approximation Example 3.7, for the two integer values surrounding the optimal target stock level. As with the normal approximation of the

KPI	comp.Poisson Ex 3.2	normal Ex 3.7	normal Ex 3.7
s	2	2	3
C_v	80.69	80.13	77.72
$\overline{I}$	0.98	0.98	1.56
$\overline{B}$	0.98	0.98	0.56
$\overline{A}$	0.44	0.50	0.34
$\overline{P}$	12.90	10	6.83
$\overline{OF}$	20	20	20
S_3	56%	50%	66%
S_2	35%	50%	66%

Table 3.9: Example 3.7: Base-stocks KPI's compound Poisson and normal approximation

Poisson example we can notify also here a very close resemblance for $s = 2$ on variable cost C_v, average inventory $\overline{I}$ and average backorders $\overline{B}$. The stock out frequency $\overline{A}$ and the linked service levels also here show a larger deviation than the other KPI's. For the fill rate S_2 we can report a very high deviation of 15%. If we realize that in practice the fill rate is used to 'simulate' the order line service level S_{OL}, we must indicate that carefulness is necessary, especially if we are confronted with a small leadtime demand.

Now we will make a comparison between the compound Poisson demand and the normal approximation for the base-stock policy.

As we have a larger ratio σ/ν for the compound Poisson approximation than for the Poisson approximation, there is also a bigger deviation between the normal approximation and the compound Poisson distribution, see Figure 2.13, than between the Poisson distribution and the normal approximation, see Figure 2.11. Nonetheless the resemblance between the approximation of the KPI's is even for the compound Poisson high for $\nu = 2$ and certainly for $\nu = 10$, see Figure 3.11.

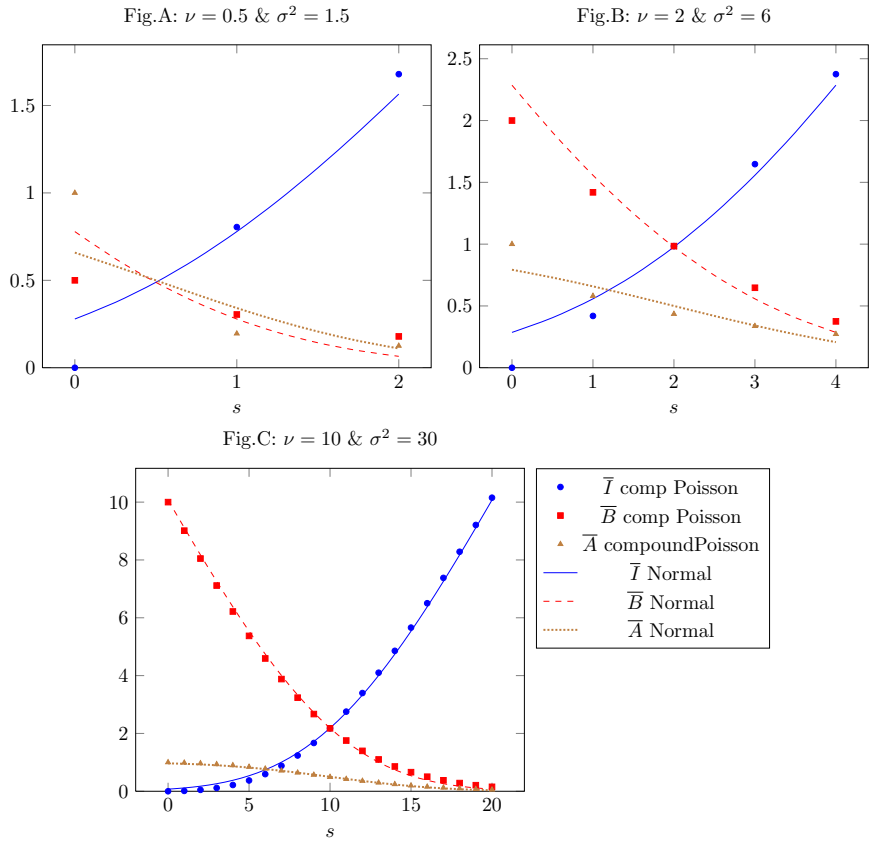

Figure 3.11: Base-stock $\overline{I}$, $\overline{B}$ & $\overline{A}$ for compound Poisson and normal demand

In Figure 3.12 we have plotted the average new shortages $\overline{P}$ and for this performance measure we do see a clear deviation between the compound Poisson KPI and the normal demand KPI, as it always underestimated in the normal approximation. So we can conclude that the normal approximation is good for $\overline{I}$ and $\overline{B}$. The accuracy is lower for $\overline{A}$ and is the lowest for $\overline{P}$. So we must be aware that fill rate constraints for a small leadtime demand does not lend itself for a normal approximation, accuracy is higher for large ν, Figure 3.12 D, then for small ν, Figure 3.12A.

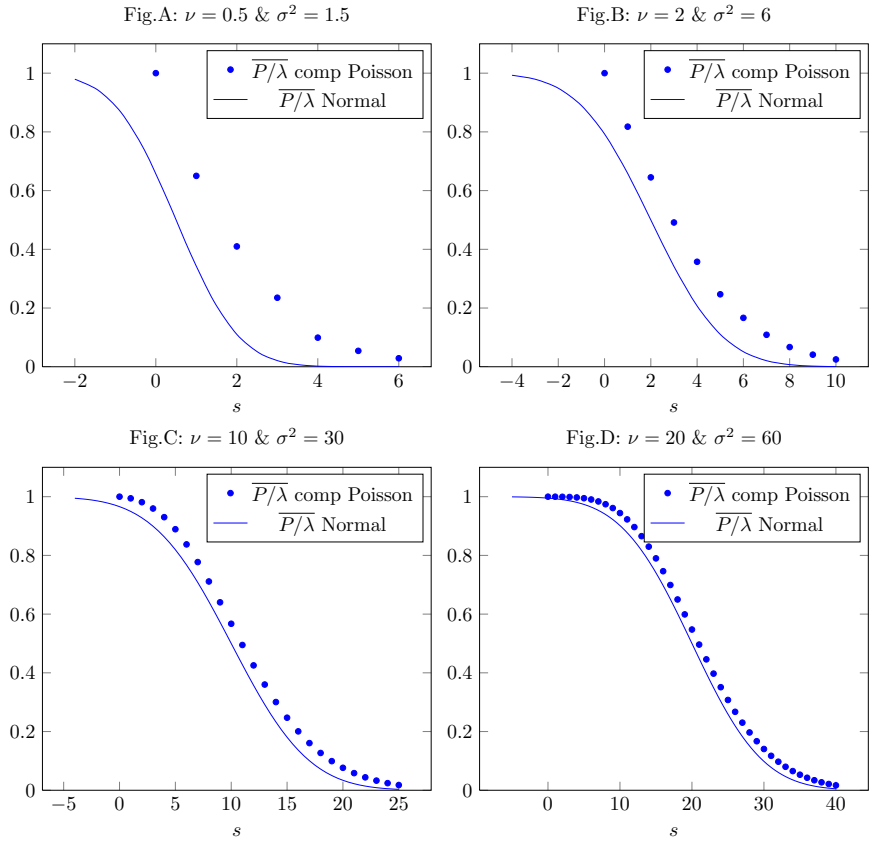

Figure 3.12: Base-stock $\overline{P}/\lambda$ for compound Poisson and normal demand

3.4.3 Poisson (r, Q) normal approximation

Example 3.8 (r, Q) **normal demand, Poisson approximation**

We will now use the normal demand to approximate the Poisson demand in case of an (r, Q) policy. Example 3.8 in Table 3.10 uses the same input as the Poisson Example 2.12, the optimal r^* and Q^* values are as expected no integer values.

Rounding the optimal values r^* and Q^* gives us $r = 1$ and $Q = 12$. Comparing the normal demand (r, Q) policy using these integer values for r and Q with the exact Poisson (Example 2.12) reveals there is a very close match between the found optimal and also all of the KPI's, see Table 3.11.

In Figure 3.13 we compare the normal demand approximation KPI's $\overline{I}$, $\overline{B}$ and $\overline{A}$ with the exact Poisson demand KPI's for three different order quantity Q values: $Q = 1$ (equals base-stock), $Q = 5$ and $Q = 10$.

Input	Value	KPI	Value
λ	20	r^*	0.59
h	32	Q^*	12.28
k	80	C_v^*	300.11
b	100	$\overline{I}$	3.74
L	0.2	$\overline{B}$	0.50
ν	4	$\overline{A}$	0.24
σ^2	4	$\overline{OF}$	1.63
a	0	$\overline{P}$	4.84%
p	0	$S_2 = S_3$	76%
		S_1	4%

Table 3.10: Example 3.8: (r, Q) normal demand, Poisson approximation

KPI	Poisson Ex 2.12	normal Ex 3.8
r	1	1
Q	12	12
C_v	300.13	300.78
$\overline{I}$	3.92	3.92
$\overline{B}$	0.42	0.42
$\overline{A}$	0.25	0.22
$\overline{P}$	5.03	4.34
$\overline{OF}$	1.67	1.67
$S_2 = S_3$	75%	78%
S_1	9%	7%

Table 3.11: Example 3.8: (r, Q) KPI's Poisson and normal approximation

As we made the $r + 0.5$ corrections for the integration, we see a very close match for $\overline{I}$, Figure 3.13A & B, and $\overline{B}$, 3.13C & D. While for the base-stock KPI's we still had a relevant deviation for small ν, this aspect is less prominent in the (r, Q) policy. The (r, Q) KPI's can be considered as an average of a set of base-stock KPI's. The $(Q = 1)$ lines and markers represent the base-stock KPI's in Figure 3.13. We have seen that the deviation was the greatest for small ν and for small r values. As the order quantity Q increases, the deviation will become smaller for $r = -Q$, see the examples for $Q = 5$ and $Q = 10$. This leads us to conclude that the normal approximation for Poisson demand is even better for an (r, Q) policy, while it was already well performing for a base-stock policy. Where the KPI with the biggest deviation for base-stock policy was the stockout frequency $\overline{A}$, we can see that this is still the weakest approximation, but we see that the resemblance increases with Q.

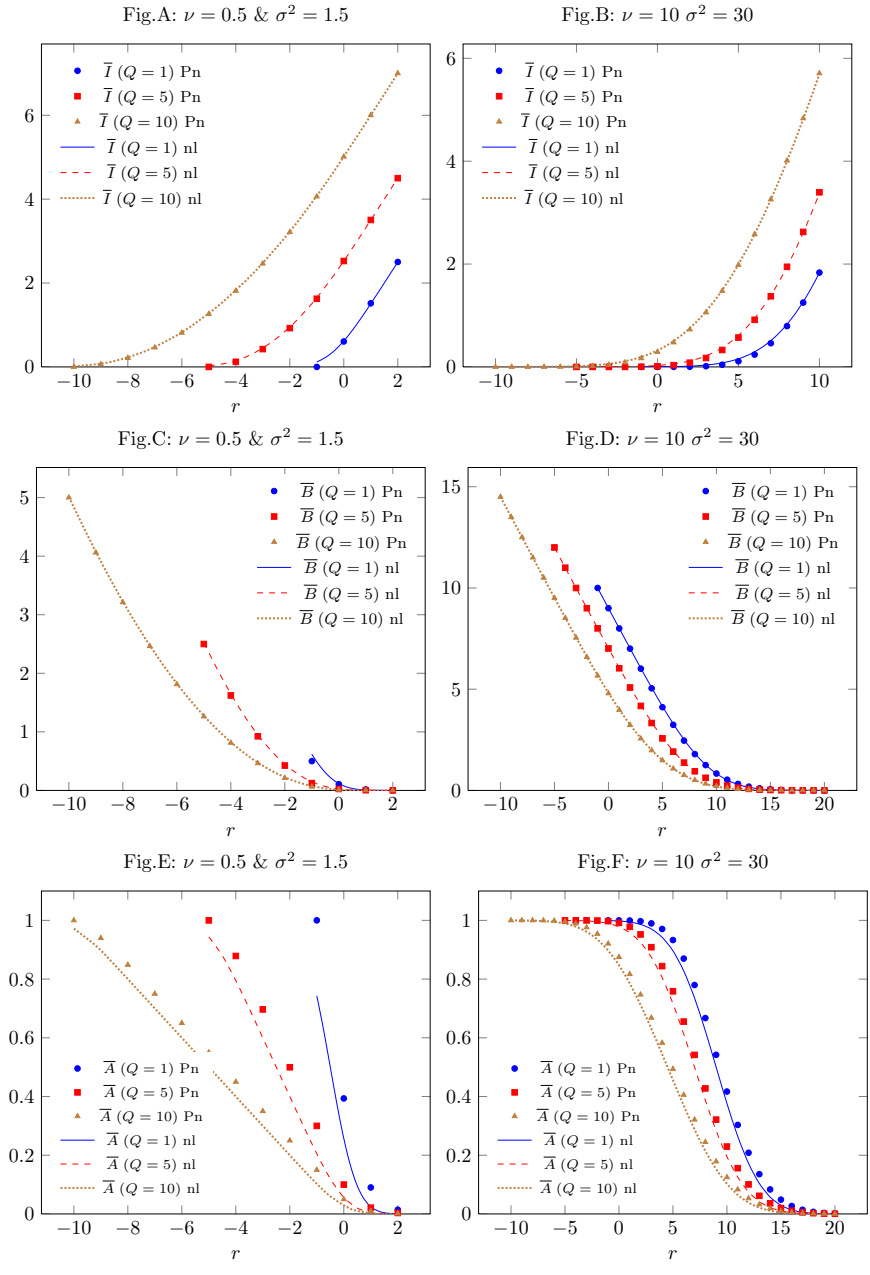

Figure 3.13: (r, Q) KPI's for Poisson & normal demand

3.4.4 Compound Poisson (r, Q) normal approximation

Example 3.9 Compound Poisson (r, Q) normal approximation

Converting the compound Poisson Example 3.3 into a normal approximation gives Example 3.9 in Table 3.12.

Input	Value	KPI	Value
λ	20	r^*	1.21
h	32	Q^*	13.25
k	80	C_v^*	335.17
b	100	$\overline{I}$	4.54
L	0.2	$\overline{B}$	0.69
ν	4	$\overline{A}$	0.24
σ^2	12	$\overline{OF}$	1.51
a	0	$\overline{P}$	4.82
p	0	$S_1 = S_2 = S_3$	76%
		S_1	17%

Table 3.12: Example 3.9: (r, Q) normal demand, compound Poisson approximation

If we compare the rounded result of Example 3.9 ($r = 1$ and $Q = 13$) we see in Table 3.13 there is little difference in the total variable cost C_v (1%). The found values for r and Q although are not the same, for the compound Poisson $(r^*, Q^*) = (0, 14)$ while for the normal approximation we find $(r, Q) = (1, 13)$. Which makes we can only compare the costs but no longer the KPI's. So we also added the normal approximation for $(r, Q) = (0, 14)$, which gives evidently a higher cost than $(r, Q) = (1, 13)$. Within the normal approximation for $(r, Q) = (0, 14)$ there is a very close match for $\overline{I}$, $\overline{B}$ and S_3. Only the fill rate S_2 and the linked $\overline{P}$ show a deviation worth mentioning.

As we did for the base-stock policy we will also make a comparison between the exact compound Poisson demand (r, Q) policy KPI's and the normal demand approximations, see Figure 3.15. For the client demand pattern we are using the definition form Figure 2.5 with $\chi = 2.3$. We have seen for compound Poisson base-stock policy KPI approximation, that the increase of the leadtime demand variation σ^2 deteriorated the quality of the approximated KPI's. We can still see this by comparing Figure 3.13B and 3.15B, the first has a smaller deviation, $\sigma^2/\nu = 1$, and this yields a better approximation. Just as with the (r, Q) Poisson demand approximation we see that the resemblance increases as the order quantity Q increases. The strongest point of attention goes to the fill rate S_2 and average new backlogs $\overline{P}$, Figure 3.15G & H. We approximated this by using $\overline{A}$. For a small leadtime demand

KPI	compound Poisson Ex 3.3	normal Ex 3.9	normal sol Ex 3.3
r	0	1	0
Q	14	13	14
C_v	339.10	335.44	336.43
$\overline{I}$	4.35	4.64	4.33
$\overline{B}$	0.85	0.64	0.83
$\overline{A}$	0.28	0.23	0.27
$\overline{P}$	7.08	4.58	5.40
$\overline{OF}$	1.43	1.54	1.43
S_2	65%	77%	73%
S_3	72%	77%	73%

Table 3.13: Example 3.9: (r, Q) KPI's compound Poisson and normal

we can see quite large deviations, where the estimated new backlogs is 10% smaller than the exact new backlogs. This means for small leadtime demand ν we could set the fill rate at e.g. 73% while the exact value can be much smaller, e.g. 65%, as we have seen in Example 3.9.

After making the comparison between a compound Poisson with $\sigma^2/\nu = 3$ and the normal approximation, we finally want to test the impact of a larger σ^2/ν on the approximation quality. We have already seen that it will deteriorate due to the fact that the normal approximation will have a large volume of negative values. As there is a larger variance in the customer demand pattern, there will be a larger impact due to discretization. We use a new customer demand pattern as defined in Figure 3.14 with $\chi = 4.9$ and $\sigma^2/\nu = 6.2$.

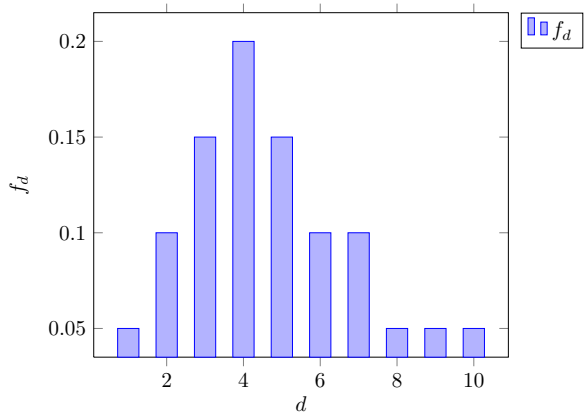

Figure 3.14: Compound Poisson customer demand size $\chi = 4.9$

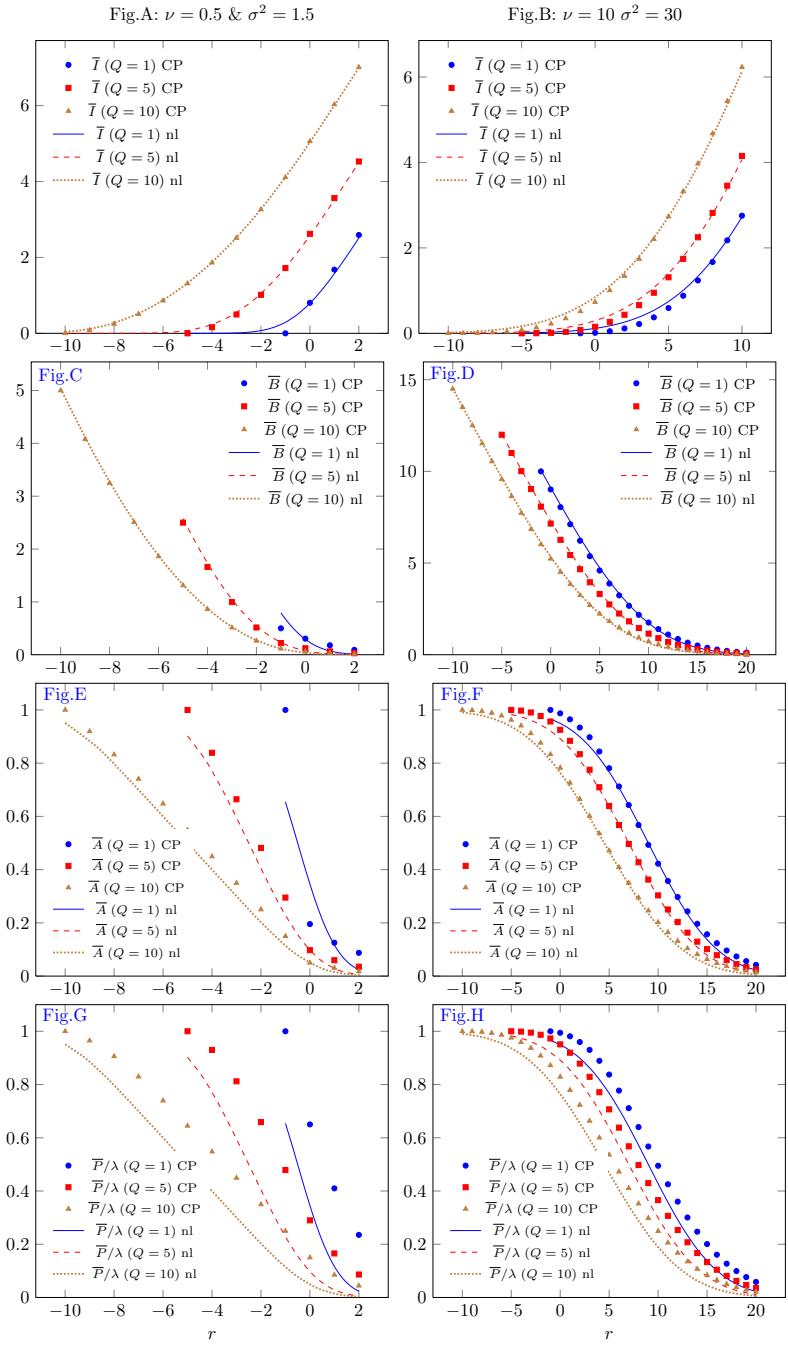

Figure 3.15: (r, Q) KPI's for compound Poisson & normal, $\sigma^2/\nu = 3$

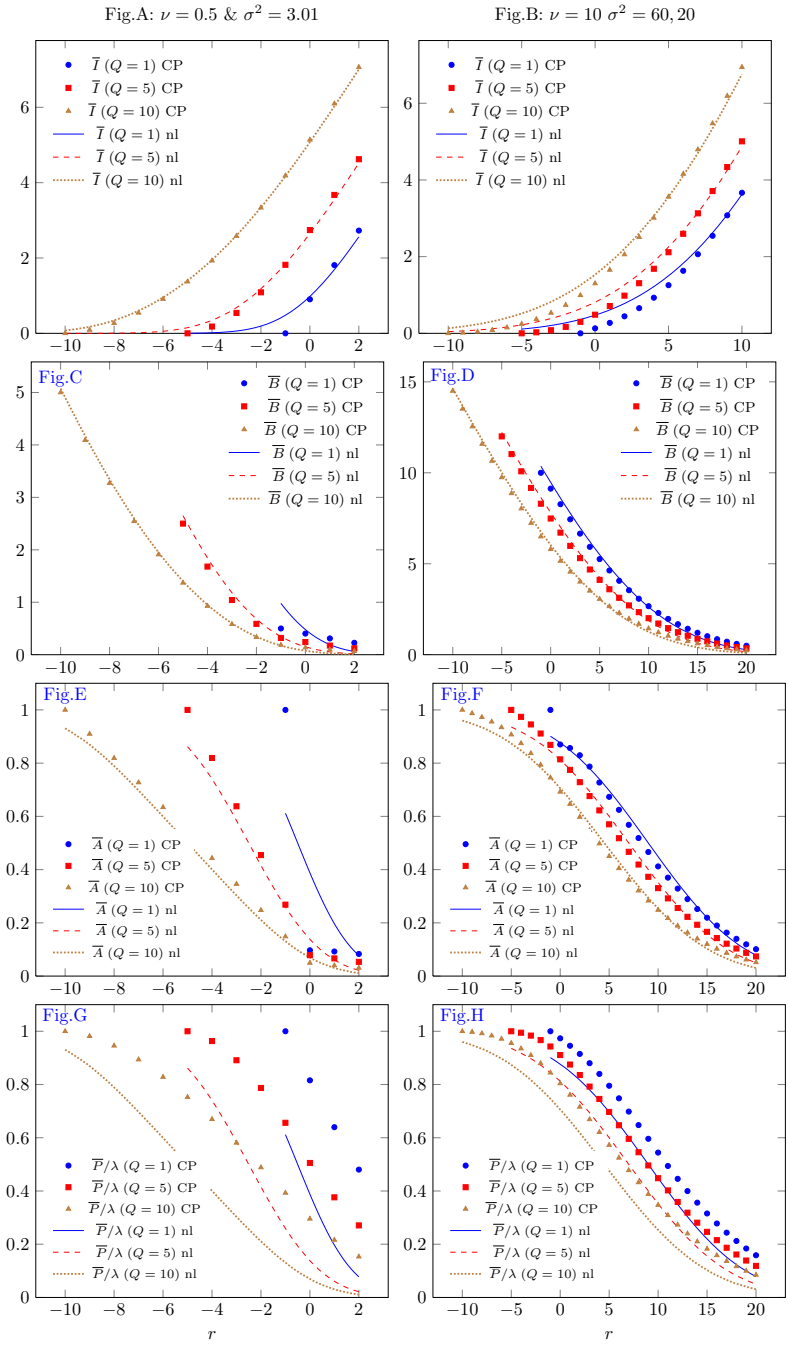

Figure 3.16: (r, Q) KPI's for compound Poisson & normal, $\sigma^2/\nu = 6$

3.4.5 Compound Poisson (s, S) normal approximation

Example 3.10 (s, S) **normal demand, compound Poisson approximation**

We would also like to have an idea how good a normal approximation works in case of an (s, S) policy. For the normal approximation we retake the (r, Q) normal approximation. So within this approximation we not only approximate the demand but we will also miss the non-uniform distribution of the (s, S) inventory position IP.

In Table 3.14 we compare the optimal (s, S) solution with the optimal normal (r, Q) solution in the second column. In the third column we also add the normal (r, Q) approximation for $(r, Q) = (0, 13)$ which is the optimal (s, S) solution. The cost C_v is really close to the exact value. Comparing the first and third column we see an underestimation of average inventory by 10% and an overestimation of the average backorders by 5%. The largest error is made on the average new backorders as it is nearly 20%.

KPI	compound Poisson Ex15	normal Ex15	normal sol Ex15
s	0	1	0
S	13	13	13
C_v	337.85	335.44	337.67
$\overline{I}$	4.31	4.64	3.90
$\overline{B}$	0.85	0.64	0.90
$\overline{A}$	0.28	0.23	0.29
$\overline{P}$	7.09	4.58	5.82
$\overline{OF}$	1.43	1.54	1.54
S_2	65%	77%	71%
S_3	72%	77%	71%

Table 3.14: Example 3.10: (s, S) *KPI's compound Poisson and normal*

We know that the (r, Q) and (s, S) are equal in case of continuous review and a demand that is continuous (e.g. normal) or Poisson. If the reorder quantity $Q = 1$ or $S - s = 1$ there is also no difference between an (r, Q) and (s, S) policy.

In the next figures we will give an indication of the KPI's if we would use a normal (r, Q) approximation for an (s, S) policy while having Q or $S-s$ for a small leadtime demand $\nu = 0.5$, charts on the left, and a larger leadtime demand $\nu = 10$, charts on the right. Figure 3.17 is based upon $\sigma^2/\nu = 3$ and Figure 3.18 is based upon a larger variation $\sigma^2/\nu = 6$. In section 3.5 we will analyze these differences in detail.

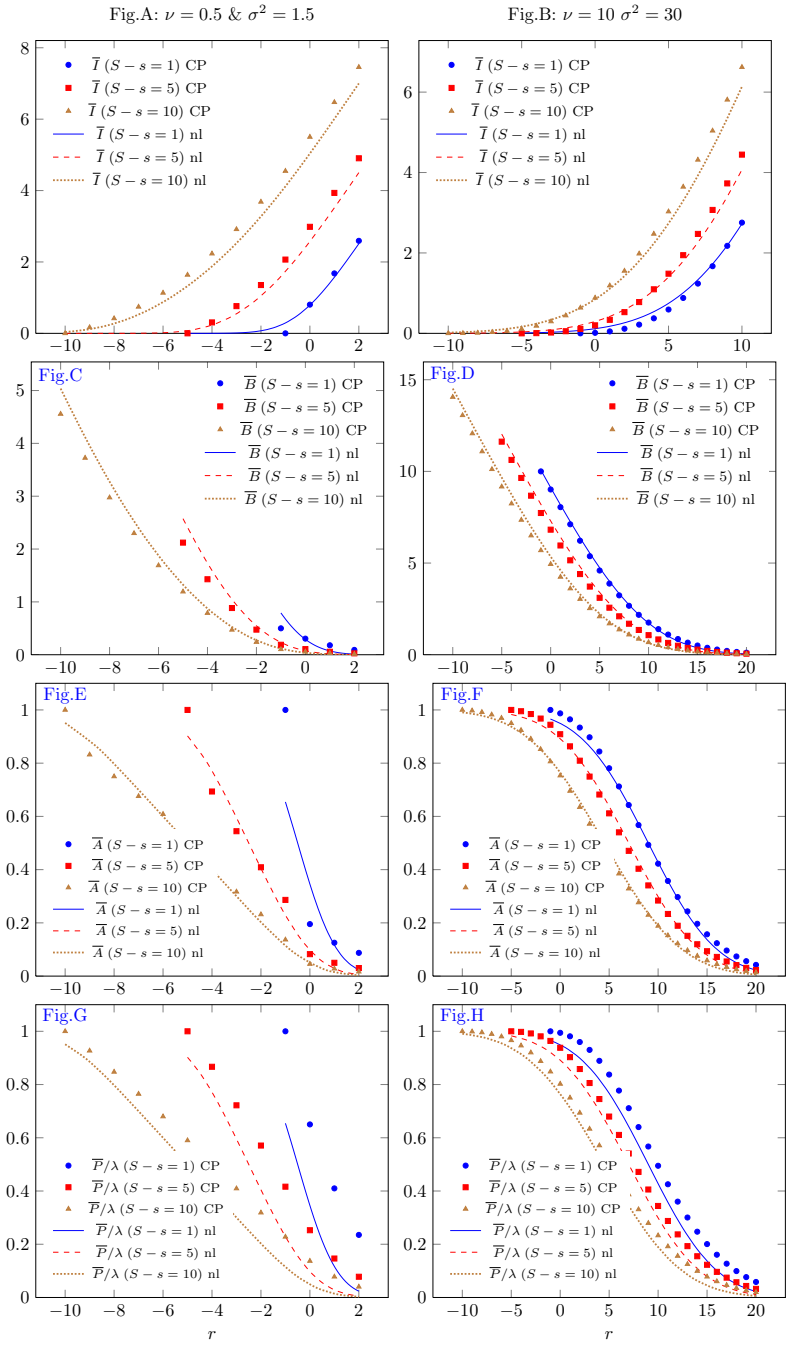

Figure 3.17: (s, S) KPI's for compound Poisson & normal, $\sigma^2/\nu = 3$

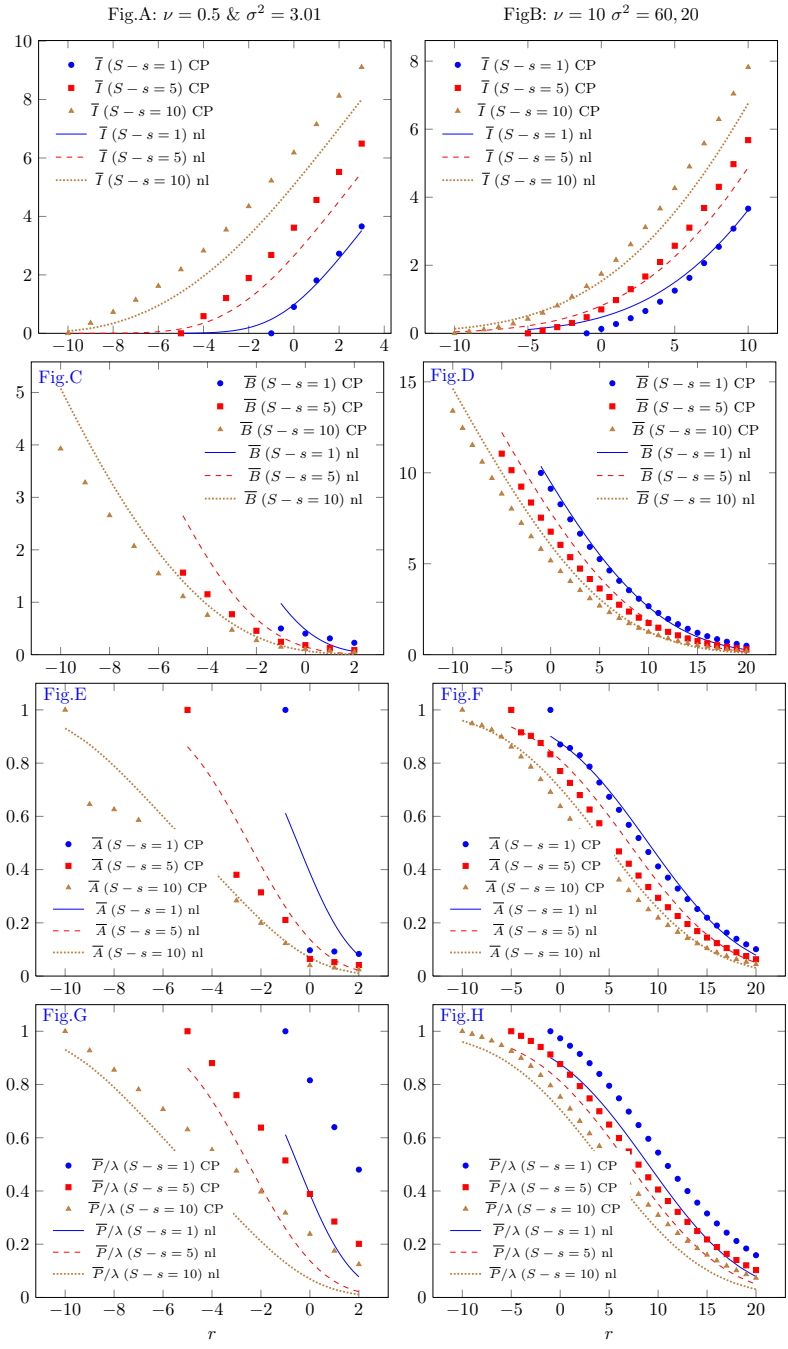

Figure 3.18: (s, S) KPI's for compound Poisson & normal, $\sigma^2/\nu = 6$

3.5 Analysis of KPI approximation errors

Contribution 7: Normal demand (r, Q) ***&*** (s, S) ***KPI approximation error analysis***

We give an overview of the errors made by approximating the Poisson (Pn) or compound Poisson (cP) demand by a normal leadtime demand, see also De Schrijver et al. (2012). We do this for a demand where χ ranges from 1 (Poisson, plotted on the top: figures A & B), over $\chi = 2.3$ (plotted in the middle, figures C & D) up to $\chi = 4.9$ (plotted on the bottom, figures E & F). Each error is calculated for a small leadtime demand ($\nu = 0.5$), plotted on the left, and also for a larger leadtime demand ($\nu = 10$), plotted on the right. For each of these six figures we let r or s range and plot three curves for three order quantities: $Q = 1$, $Q = 5$ and $Q = 10$. The curve for $Q = 1$ matches the base-stock policy. For the curves $Q = 5$ and $Q = 10$ we give the normal approximation error ϵ for an (r, Q) and an (s, S) policy. We prefer to use here S_2, S_3 and S_{OL} instead of $\overline{P}/\lambda$ and $\overline{A}$, as the service levels are easier to interpret. The error ϵ for $\overline{I}$ is given by 3.26, where nl stands for normal demand, Pn for Poisson demand and cP for compound Poisson demand. The errors for $\overline{B}$, S_2, S_3, S_{OL} and $\overline{OF}$ are calculated accordingly, see (3.27)-(3.31).

$$\epsilon_{\overline{I}} = \overline{I}_{nl} - \overline{I}_{Cp} \tag{3.26}$$

$$\epsilon_{\overline{B}} = \overline{B}_{nl} - \overline{B}_{Cp} \tag{3.27}$$

$$\epsilon_{S_3} = S_{3nl} - S_{3Cp} \tag{3.28}$$

$$\epsilon_{S_3} = S_{2nl} - S_{2Cp} \tag{3.29}$$

$$\epsilon_{S_{OL}} = S_{OLnl} - S_{OLCp} \tag{3.30}$$

$$\epsilon_{\overline{OF}} = \overline{OF}_{nl} - \overline{OF}_{Cp} \tag{3.31}$$

On the following pages we give a visual overview of the errors on the KPI's, here we comment and give conclusions on these figures:

- Error average inventory $\epsilon_{\overline{I}}$, see Figure 3.19
  - $Max(|\epsilon_{\overline{I}}|)$ decreases as Q increases for bS and (r, Q)
  - $Max(|\epsilon_{\overline{I}}|)$ increases as Q increases for (s, S)
  - Magnitude of $\epsilon_{\overline{I}}$ does not increase with ν (Fig A, C & E versus B, D & F, so the relative error on the total $\overline{I}$ becomes very small with increasing ν
  - $Max(|\epsilon_{\overline{I}}|)$ is larger for (s, S) than for (r, Q), Fig C-F
  - $\epsilon_{\overline{I}}$ can be positive and negative for all policies

  - $\epsilon_{\overline{I}}$ goes towards zero for large r in case of bS and (r, Q)
  - $\epsilon_{\overline{I}}$ goes towards a fixed negative value for large r in case of (s, S), see Fig C-D

- Error average backorders $\epsilon_{\overline{B}}$, see Figure 3.20
  - $Max(|\epsilon_{\overline{B}}|)$ decreases as Q increases for bS and (r, Q)
  - $|\epsilon_{\overline{B}}|$ for (s, S) is smaller than for (r, Q) for large r
  - Magnitude of $\epsilon_{\overline{B}}$ does not increase with ν (Fig A, C & E versus B, D & F, so the relative error on the total $\overline{B}$ becomes very small with increasing ν
  - $Max(|\epsilon_{\overline{B}}|)$ increases rapidly for small r for (s, S), see Fig C-F
  - $Max(|\epsilon_{\overline{B}}|)$ is larger for (s, S) than for (r, Q), as expected
  - $\epsilon_{\overline{B}}$ can be positive and negative for all policies
  - $Max(|\epsilon_{\overline{B}}|)$ is small for (r, Q)
  - $\epsilon_{\overline{B}}$ goes towards zero for large r for all policies
  - $\epsilon_{\overline{B}}$ is equal for (r, Q) and (s, S) if $r = -Q$ (Fig C-F)

- Error ready rate service level ϵ_{S_3}, see Figure 3.21
  - ϵ_{S_3} is always positive for Poisson demand (Fig A & B)
  - ϵ_{S_3} is positive for large r and high service levels and then goes towards zero
  - $Max(|\epsilon_{S_3}|)$ increases with increasing σ^2/ν (lower figures)
  - $Max(|\epsilon_{S_3}|)$ decreases with increasing ν (figures on the right)
  - ϵ_{S_3} is equal for (r, Q) and (s, S) if $r = -Q$ (Fig C-F)

- Error fill rate service level ϵ_{S_2}, see Figure 3.22
  - $\epsilon_{S_2} \geq 0$, so there is always an overestimation of the fill rate
  - $Max(|\epsilon_{S_2}|)$ decrease as ν increases (Fig A, C & E vs B, D & F)
  - $Max(|\epsilon_{S_2}|)$ is high for small ν (5% till 40%)! (Fig A, C & E)
  - ϵ_{S_2} is equal for (r, Q) and (s, S) if $r = -Q$ (Fig C-F)
  - ϵ_{S_2} goes towards zero for large r
  - $Max(|\epsilon_{S_2}|)$ increases with increasing σ^2/ν (lower figures)
  - $Max(|\epsilon_{S_2}|)$ decreases with increasing ν (figures on the right)

- Error order line service level $\epsilon_{S_{OL}}$, see Figure 3.23
  - $\epsilon_{S_{OL}} \geq \epsilon_{S_2}$, so there is always an overestimation of the order line service level
  - $\epsilon_{S_{OL}} \geq 0$
  - $\epsilon_{S_{OL}}$ goes towards zero for large r
  - $Max(|\epsilon_{S_{OL}}|)$ up to 50% and more for small ν (Fig A, C & E)
  - $Max(|\epsilon_{S_{OL}}|)$ up to 15% and more for larger ν (Fig B, D & F)

  - $\epsilon_{S_{OL}}$ is equal for (r, Q) and (s, S) if $r = -Q$ (Fig C-F)
  - $Max(|\epsilon_{S_{OL}}|)$ increases with increasing σ^2/ν (lower figures)
  - $Max(|\epsilon_{S_{OL}}|)$ decreases with increasing ν (figures on the right)
- Error order frequency (s, S) $\epsilon_{\overline{OF}}$, see Figure 3.24
  - $\epsilon_{\overline{OF}} \geq 0$, so we overestimate the (s, S) order frequency because the average order quantity $> S - s$
  - $\epsilon_{\overline{OF}}$ largest for $Q = 1$
  - $\epsilon_{\overline{OF}}$ decreases as Q increases, approaches zero for larger Q
  - $\epsilon_{\overline{OF}}$ is linear with λ, as is $\overline{OF}$, if χ is unchanged
  - $\epsilon_{\overline{OF}}$ increases with σ^2/ν and thus with χ

1. All $Max(|\epsilon|)$ increase with χ and σ^2/ν (increasing demand variability)
2. $\epsilon_{\overline{I}} = \epsilon_{\overline{B}}$ for bS and (r, Q)
3. All ϵ peak for bS at $r = -1$ for small ν
4. $\epsilon_{S_2} = \epsilon_{S_3} = \epsilon_{S_{OL}}$ for $\chi = 1$ (Poisson demand), as the KPI's are also equal in case of Poisson demand

3.5.1 Average inventory approximation error

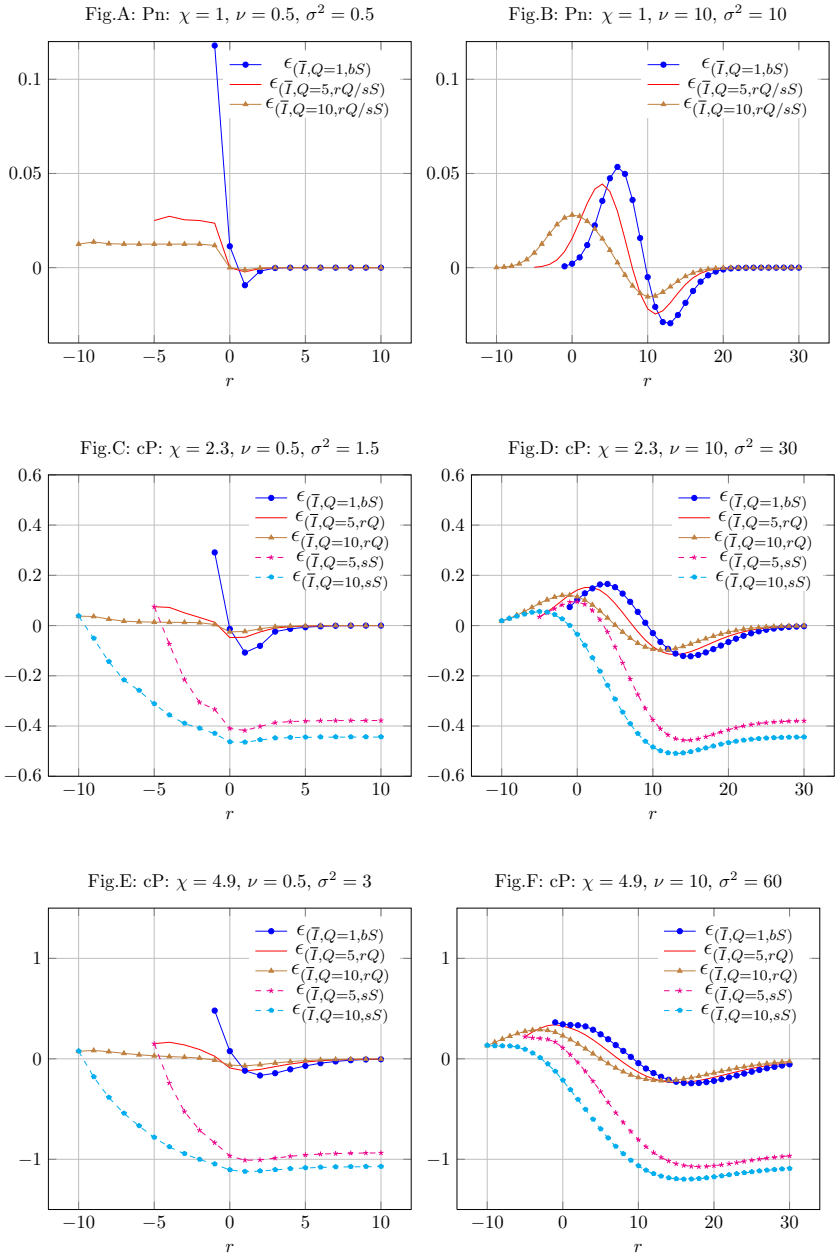

Figure 3.19: $\overline{I}$ approximation error

3.5.2 Average backorders approximation error

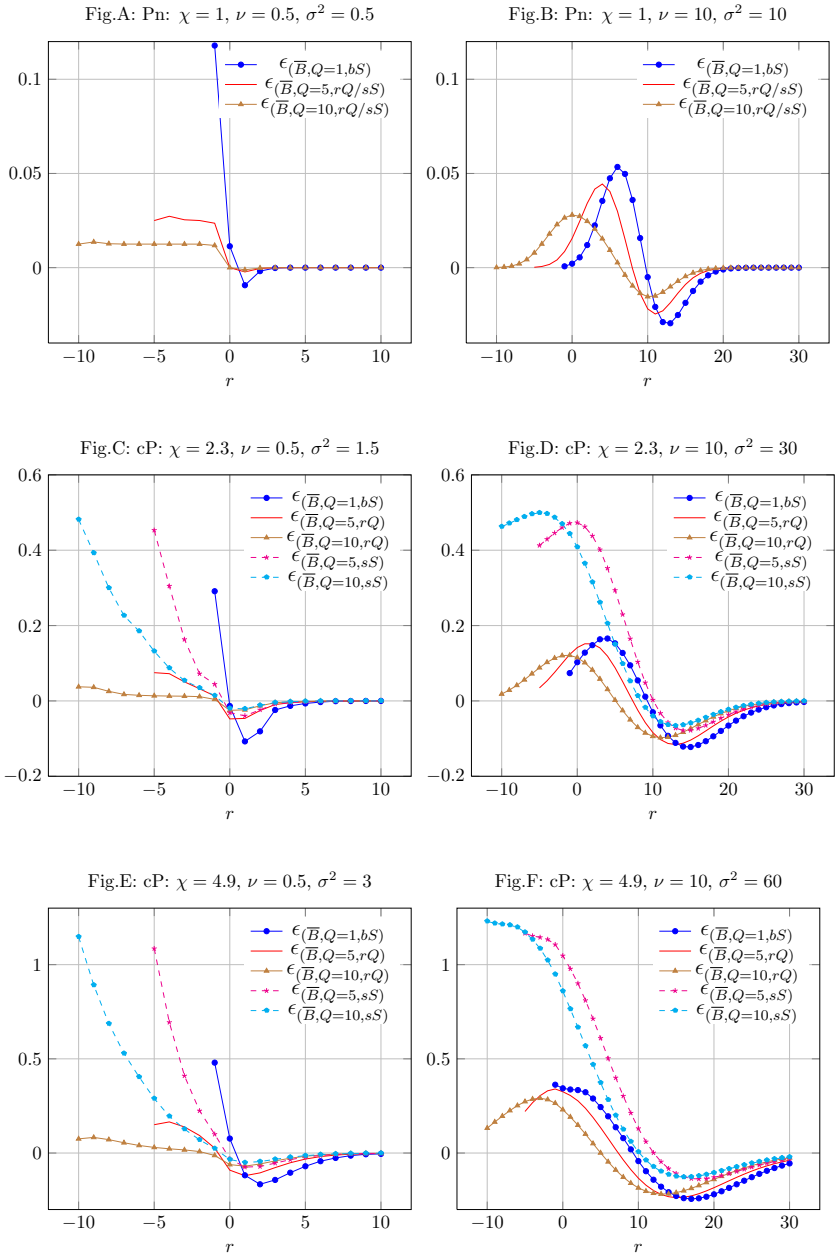

Figure 3.20: $\overline{B}$ approximation error

3.5.3 Ready rate approximation error

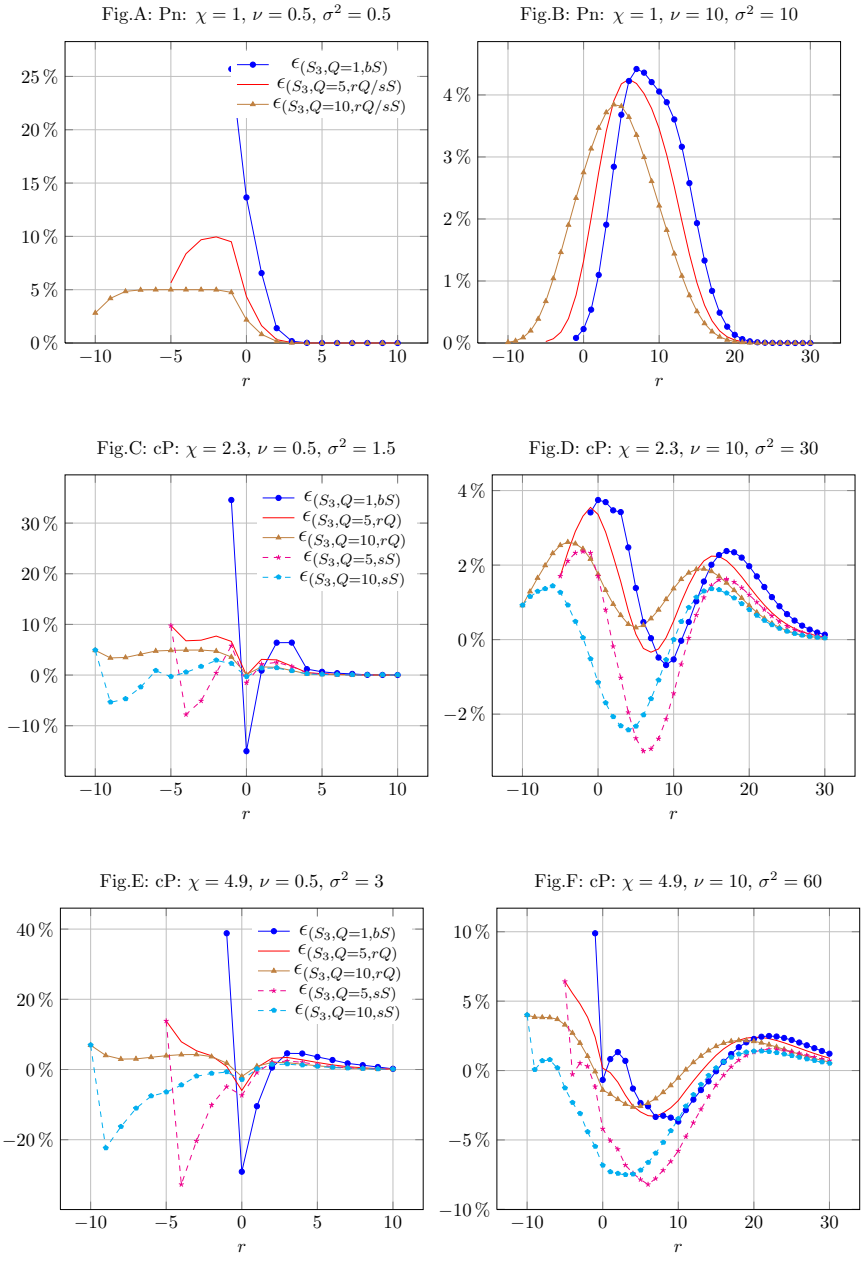

Figure 3.21: S_3 approximation error

3.5.4 Fill rate approximation error

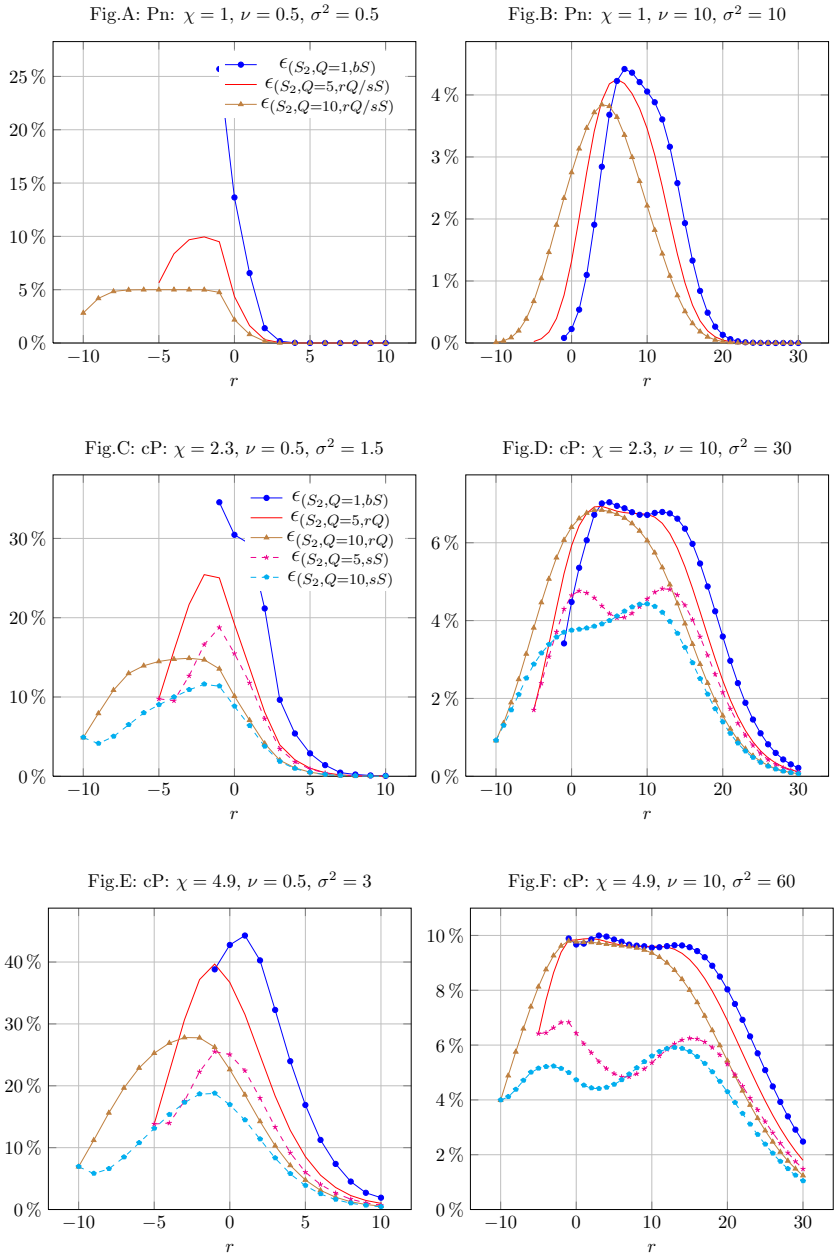

Figure 3.22: S_2 approximation error

3.5.5 Order line service level approximation error

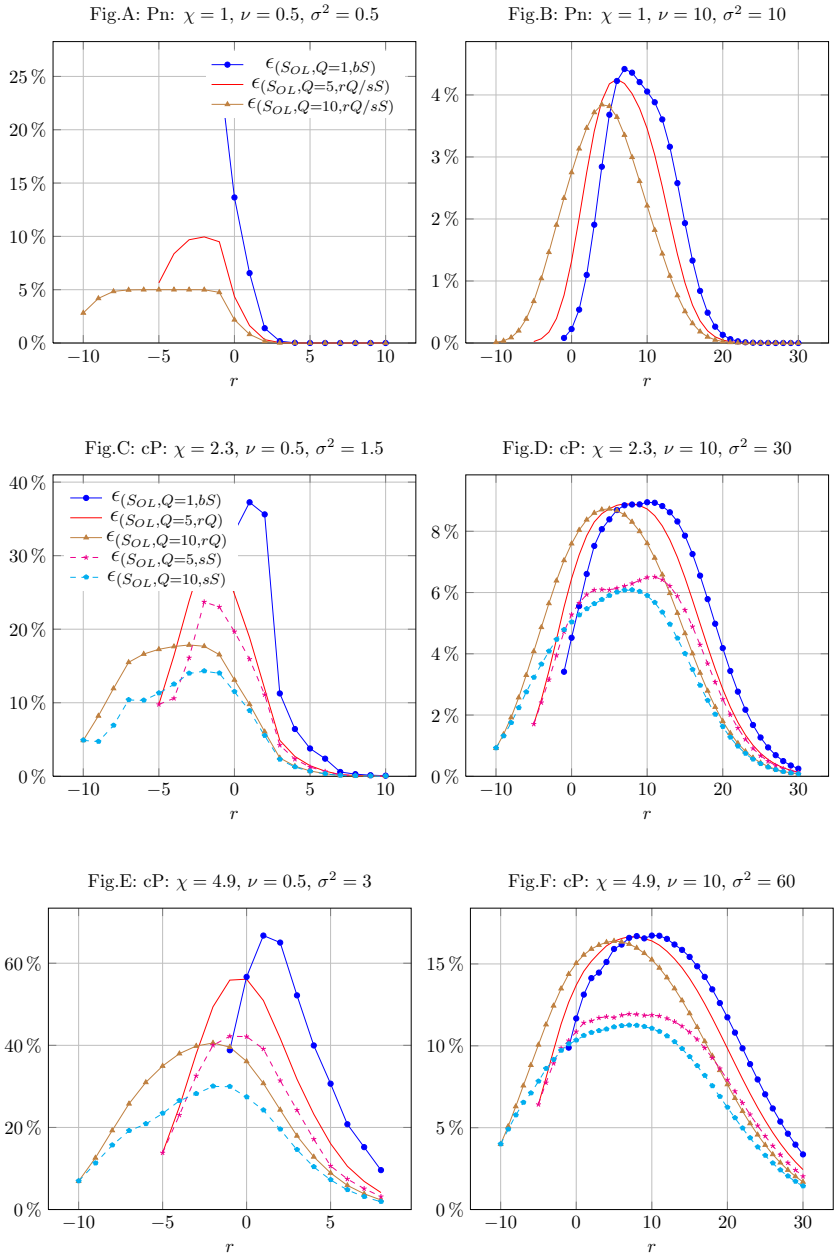

Figure 3.23: S_{OL} approximation error

3.5.6 Order frequency (s, S) approximation error

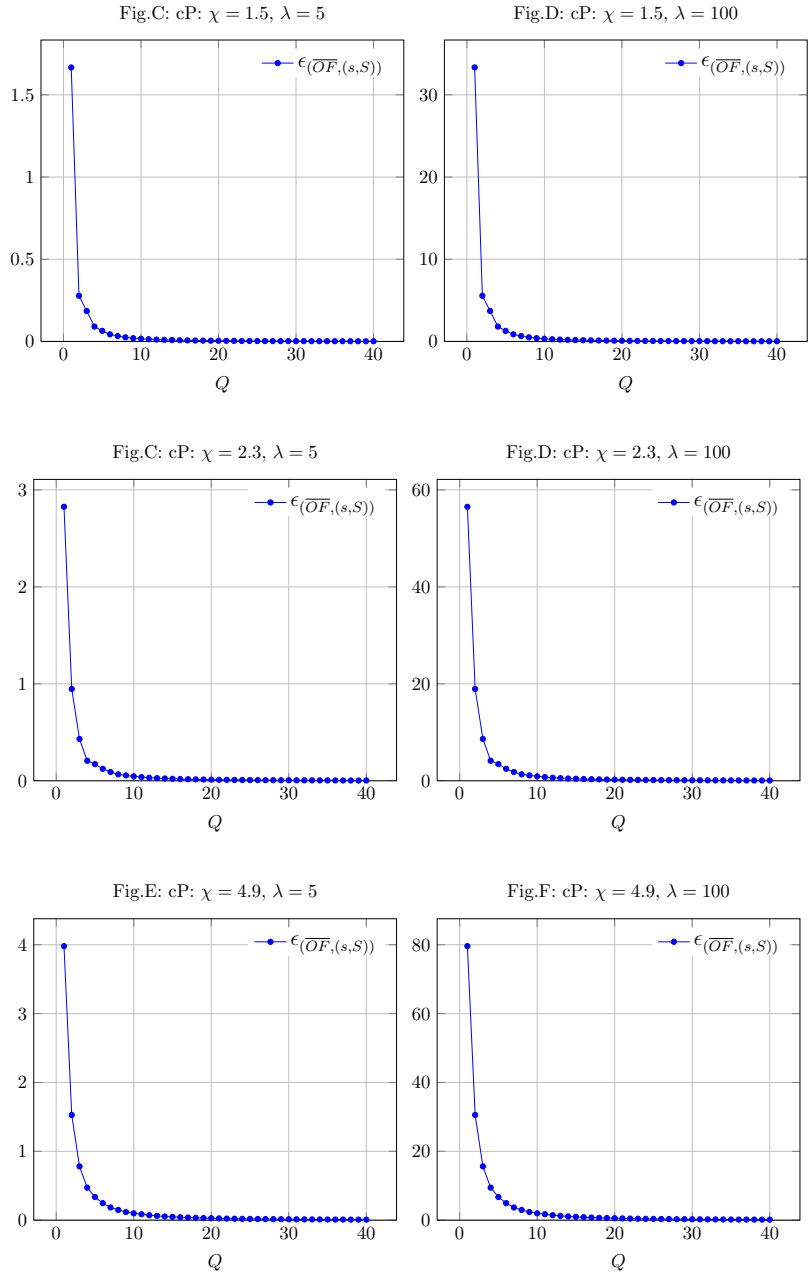

Figure 3.24: $\overline{OF}$ (s, S) approximation error

3.6 KPI error reduction functions

In the previous section 3.5 we have analyzed the errors made when applying a normal demand approximation for a Poisson or compound Poisson demand and for an (r, Q) or (s, S) policy. We now want to create functions that can reduce these errors, while still using the normal approximation as foundation.

3.6.1 Assumptions and data set

In practice we almost always encounter the following assumptions:

- $r \geq 0$
- $S_2 \geq 80\%$

Making use of these assumptions we create a data set of 1450 (J_{DS}) instances based upon the following variable ranges:

- $r \geq 0$, and service level $\geq 80\%$ and $\leq 99.9\%$
- $\nu \in [0.05, 0.5, 1, 2, 5, 10, 15, 20]$, for $\nu > 20$ normal approximation error is negligible
- $Q \in [1, 5, 9, 13, 17]$, so Q cover the range from 1 till ν
- $\sigma^2/\nu \in [1, 3, 6]$, so we have Poisson demand and compound Poisson demand based upon Figures 2.5 and 3.14

For the $\overline{OF}$ (s, S) approximation we have slightly adapted the data set, as this KPI does not depend on r. We have assumed $L = 1$, so $\lambda = \nu$ and $\psi = \sigma$. We here let Q vary between 1 and 40, ψ^2/λ was set at 1.18, 1.66, 3 and 6.

In the following sections we will create scenarios of performance indicator formulations, leading to an error reduction. The error functions are defined in (3.32) - (3.35). In (3.35) S can be replaced by S_2, S_3 and S_{OL}. For the average inventory and order frequency error we convert it in a relative error, so we can also express it in %. For the average backorders KPI $\overline{B}$ we do not express it in %, because in the range we are interested ($S_2 \geq 80\%$) there are really small $\overline{B}$-values, what would lead to extreme high % errors, but these errors have no practical meaning.

$$\epsilon_{|\overline{I}|} = |\frac{\overline{I}_{nl} - \overline{I}_{cP}}{\overline{I}_{cP}}| \tag{3.32}$$

$$\epsilon_{|\overline{B}|} = |\overline{B}_{nl} - \overline{B}_{cP}| \tag{3.33}$$

$$\epsilon_{|\overline{OF}|} = |\frac{\overline{OF}_{nl} - \overline{OF}_{cP}}{\overline{OF}_{cP}}| \tag{3.34}$$

$$\epsilon_{|S|} = |S_{nl} - S_{cP}| \tag{3.35}$$

The average and maximum error definitions are given by (3.36)-(3.43).

$$\epsilon_{max|\overline{I}|} = max(\epsilon_{|\overline{I}|j}), j = 1, ..., J_{DS} \tag{3.36}$$

$$\epsilon_{max|\overline{B}|} = max(\epsilon_{|\overline{B}|j}), j = 1, ..., J_{DS} \tag{3.37}$$

$$\epsilon_{max|\overline{OF}|} = max(\epsilon_{|\overline{OF}|j}), j = 1, ..., J_{DS} \tag{3.38}$$

$$\epsilon_{max|S|} = max(\epsilon_{|S|j}), j = 1, ..., J_{DS} \tag{3.39}$$

$$\epsilon_{avg|\overline{I}|} = \frac{\sum_{j=1}^{J_{DS}} \epsilon_{|\overline{I}|j}}{J_{DS}} \tag{3.40}$$

$$\epsilon_{avg|\overline{B}|} = \frac{\sum_{j=1}^{J_{DS}} \epsilon_{|\overline{B}|j}}{J_{DS}} \tag{3.41}$$

$$\epsilon_{avg|\overline{OF}|} = \frac{\sum_{j=1}^{J_{DS}} \epsilon_{|\overline{OF}|j}}{J_{DS}} \tag{3.42}$$

$$\epsilon_{avg|S|} = \frac{\sum_{j=1}^{J_{DS}} \epsilon_{|S|j}}{J_{DS}} \tag{3.43}$$

3.6.2 Error reductions: complete & corrected formulas

In practice most often, practically always, the simplified equations are used, this is here defined as scenario 1. In scenario 2 we use the complete equations and in scenario 3 we apply the 0.5 correction, see contribution 5. We analyze the average and maximum absolute error.

- **Scenario 1**: (r, Q) nl classic simplified equations, (2.162)-(2.163)
- **Scenario 2**: (r, Q) nl classic complete equations, (2.166)-(2.168)
- **Scenario 3**: (r, Q) nl with 0.5 correction, (3.5)-(3.7)

We highlighted the risk of using simplified equations in section 3.3.3. Figure 3.25 now illustrates this risk, the maximum absolute errors for ready rate, fill rate and order line service levels in both replenishment policies (r, Q) and (s, S) significantly drop in scenario 2, when no simplifications are made. The maximum errors on the service levels drop from respectively 100%, 58% and 44% to 11%, 12% and 15%. The error peaks of 100% (and more) happen for example on this case: $\nu = 20$, $\sigma^2/\nu = 6$, $\sigma = 10.97$, $Q = 1$, $r = 29$, the actual ready rate is 81%, while the simplification, ($S_3 = 1-(2.162)$), gives a ready rate of -27%, giving an absolute error of more than 100%! This case has been worked out in detail in Example 3.5, where we explain the conditions to use the simplified equations with an acceptable error. For the inventory KPI we see a major drop in error in scenario 3: going from 50% to

Figure 3.25: Maximum errors $\epsilon_{max|\overline{I}|}$, $\epsilon_{max|\overline{B}|}$ & $\epsilon_{max|S|}$: Scenarios 1-3

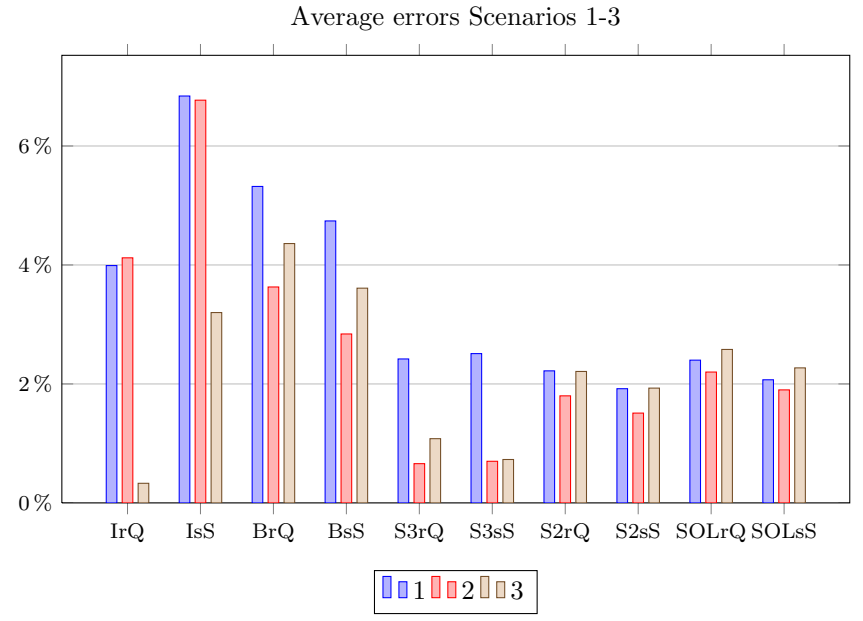

Figure 3.26: Average errors $\epsilon_{avg|\overline{I}|}$, $\epsilon_{avg|\overline{B}|}$ & $\epsilon_{avg|S|}$: Scenarios 1-3

2.5% for (r, Q) and to 19% for (s, S). The backorders KPI drops from 100% to 20% in scenario 2.

In Figure 3.26 we see that the average inventory error drops significantly in scenario 3 where we apply the 0.5 correction as described in section 3.3.1. For an (r, Q) replenishment the average error drops from 4% down to 0.2%. For (s, S) the average error is also reduced from 6% down to 3.5%. So the error reduction in scenario 3 is really significant for the average inventory in an (r, Q) policy, here we will no further look for additional error reduction functions. For an (s, S) policy we will later try to further reduce the error. Scenario 3 has no impact on the service levels, it even mildly increases the average compared to scenario 2. Further we see a significant decrease for the ready rate service level S_3 in scenario 2, going from 2% to 0.5%. For fill rate and order line service level the error remains around 2%.

The error on the order frequency in case of (s, S) replenishment has not been plotted on the previous figures, as these errors are much bigger, the maximum error is reached when $Q = 1$.

- $\epsilon_{max|\overline{OF}|} = 390\%$
- $\epsilon_{avg|\overline{OF}|} = 11.93\%$

3.6.3 Error reduction functions model

Contribution 8: KPI error reduction functions for (r, Q) ***and*** (s, S)

Based upon our generated data set and assumptions, see section 3.6.1, we now will create additional error reduction functions. The idea is to start from the KPI equations as used in scenario 3 and find error reduction functions, δ, that will reduce the error gap with the actual Poisson and compound Poisson KPI's. Using an LP (linear programming) model we want to improve the approximation quality of $\overline{I}_{cP}$, $\overline{B}_{cP}$, $\overline{OF}_{cP}$ and S_{cP} making use of respectively $\overline{I}_{nl}$, $\overline{B}_{nl}$, $\overline{OF}_{nl}$ and S_{nl} plus some of the other input parameters: ν, σ, r and Q, see (3.44)-(3.47). $\overline{I}_{nl}$, $\overline{B}_{nl}$, $\overline{OF}_{nl}$ and S_{nl} represent the KPI's as calculated by scenario 3 with normal demand and $\overline{I}_{cP}$, $\overline{B}_{cP}$, $\overline{OF}_{cP}$ and S_{cP}, are the exact compound Poisson KPI's.

$$Minimize\ f = \sum_{j=1}^{J} |1 - S_{cPj} - \delta_{Sj}|, j = 1, .., J_{DS} \tag{3.44a}$$

$$Subject\ to\ \delta_{Sj} \geq 0 \tag{3.44b}$$

$$\delta_{max} \geq |1 - S_{cPj} - \delta_{Sj}| \tag{3.44c}$$

$$\delta_{Sj} = p_{(Sa)}(1 - S_{nlj})^2 + p_{(Sb)}(1 - S_{nlj}) + p_{(Sc)} \tag{3.44d}$$

$$Minimize\ f = \sum_{j=1}^{J} |\frac{\overline{I}_{cPj} - \delta_{\overline{I}j}}{\overline{I}_{cPj}}|, j = 1, .., J_{DS} \tag{3.45a}$$

$$Subject\ to\ \delta_{\overline{I}j} \geq 0 \tag{3.45b}$$

$$\delta_{max} \geq |\frac{\overline{I}_{cPj} - \delta_{\overline{I}j}}{\overline{I}_{cPj}}| \tag{3.45c}$$

$$\delta_{\overline{I}j} = p_{(\overline{I}a)}(1 - \overline{I}_{nlj})^2 + p_{(\overline{I}b)}(1 - \overline{I}_{nlj}) + p_{(\overline{I}c)} \tag{3.45d}$$

$$Minimize\ f = \sum_{j=1}^{J} |\overline{B}_{cPj} - \delta_{\overline{B}j}|, j = 1, .., J_{DS} \tag{3.46a}$$

$$Subject\ to\ \delta_{\overline{B}j} \geq 0 \tag{3.46b}$$

$$\delta_{max} \geq |\overline{B}_{cPj} - \delta_{\overline{B}j}| \tag{3.46c}$$

$$\delta_{\overline{B}j} = p_{(\overline{B}a)}(1 - \overline{B}_{nlj})^2 + p_{(\overline{B}b)}(1 - \overline{B}_{nlj}) + p_{(\overline{B}c)} \tag{3.46d}$$

$$Minimize\ f = \sum_{j=1}^{J} |\frac{\overline{OF}_{cPj} - \delta_{\overline{OF}j}}{\overline{OF}_{cPj}}|, j = 1, .., J_{DS} \tag{3.47a}$$

$$Subject\ to\ \delta_{\overline{OF}j} \geq 0 \tag{3.47b}$$

$$\delta_{max} \geq |\frac{\overline{OF}_{cPj} - \delta_{\overline{I}j}}{\overline{OF}_{cPj}}| \tag{3.47c}$$

$$\delta_{\overline{OF}j} = p_{(\overline{OF}a)}/Q_j^2 + p_{(\overline{OF}b)}Q_j + p_{(\overline{OF}c)} \tag{3.47d}$$

So δ_S, $\delta_{\overline{A}}$, $\delta_{\overline{B}}$ and $\delta_{\overline{OF}}$ can be used to approximate respectively the appropriate service levels, the average inventory, the average backorders and the order frequency, see (3.48)-(3.51), the normal demand KPI's can be calculated using (3.5)-(3.7).

$$S_{cP} \approx 1 - \delta_S = 1 - \left[p_{(Sa)}(1 - S_{nl})^2 + p_{(Sb)}(1 - S_{nl}) + p_{(Sc)}\right] \tag{3.48}$$

$$\overline{I}_{cP} \approx \delta_{\overline{I}} = p_{(\overline{I}a)}(1 - \overline{I}_{nl})^2 + p_{(\overline{I}b)}(1 - \overline{I}_{nl}) + p_{(\overline{I}c)} \tag{3.49}$$

$$\overline{B}_{cP} \approx \delta_{\overline{B}} = p_{(\overline{B}a)}(1 - \overline{B}_{nl})^2 + p_{(\overline{B}b)}(1 - \overline{B}_{nl}) + p_{(\overline{B}c)} \tag{3.50}$$

$$\overline{OF}_{cP} \approx \delta_{\overline{OF}} = p_{(\overline{OF}a)}/Q^2 + p_{(\overline{OF}b)}/Q + p_{(\overline{OF}c)} \tag{3.51}$$

The decision variables of this model are included in p_{Xa}, p_{Xb} and p_{Xc}, where X is any of the KPI's and defined by (3.52)-(3.54) for δ_S, $\delta_{\overline{A}}$, $\delta_{\overline{B}}$. So the real decision variables are: $p_{a1} - p_{a9}$, $p_{b1} - p_{b13}$ and $p_{c1} - p_{c13}$, which gives in total 35 decision variables.

$$p_a = p_{a1}\nu + p_{a2}\sigma + p_{a3} + p_{a4}/\nu + p_{a5}/\nu^2 + p_{a2} + p_{a6}/\sigma + p_{a7}/\sigma^2 + p_{a8}Q + p_{a9}r \tag{3.52}$$

$$\begin{aligned} p_b = {} & p_{b1}\nu + p_{b2}\sigma + p_{b3} + p_{b4}\nu^2 + p_{b5}\sigma^2 \\ & + p_{b6}/\nu + p_{b7}/\nu^2 + p_{b8}/\sigma + p_{b9}/\sigma^2 + p_{b10}Q + p_{b11}r + p_{b12}/Q + p_{b13}/Q^2 \end{aligned} \tag{3.53}$$

$$\begin{aligned} p_c = {} & p_{c1}/\nu + p_{c2}\nu^2 + p_{c3}\sigma^2 + p_{c4}\nu^2 + p_{c5}\sqrt{\sigma} + p_{c6}\sqrt{\nu} \\ & + p_{c7}/\sigma + p_{c8}/\sigma^2 + p_{c9}\sigma/\nu + p_{c10}\sigma^2/\nu + p_{c11}\nu/\sigma + p_{c12}Q/\nu + p_{c13}Q/\nu^2 \end{aligned} \tag{3.54}$$

The decision variables for $\delta_{\overline{OF}}$ of this model are included in p_a, p_b and p_c as defined by (3.55)-(3.57).

$$\begin{aligned} p_a = {} & p_{a1}\lambda + p_{a2}\psi + p_{a3} \\ & + p_{a4}/\lambda + p_{a5}/\lambda^2 + p_{a2} + p_{a6}/\psi + p_{a7}/\psi^2 + p_{a8}Q + p_{a9}r \end{aligned} \tag{3.55}$$

$$\begin{aligned} p_b = {} & p_{b1}\lambda + p_{b2}\psi + p_{b3} + p_{b4}\lambda^2 + p_{b5}\psi^2 \\ & + p_{b6}/\lambda + p_{b7}/\lambda^2 + p_{b8}/\psi + p_{b9}/\psi^2 + p_{b10}Q + p_{b11}r + p_{b12}/Q + p_{b13}/Q^2 \end{aligned} \tag{3.56}$$

$$\begin{aligned} p_c = {} & p_{c1}/\lambda + p_{c2}\lambda^2 + p_{c3}\psi^2 + p_{c4}\lambda^2 + p_{c5}\sqrt{\psi} + p_{c6}\sqrt{\lambda} \\ & + p_{c7}/\psi + p_{c8}/\psi^2 + p_{c9}\psi/\lambda + p_{c10}\psi^2/\lambda + p_{c11}\lambda/\psi + p_{c12}Q/\lambda + p_{c13}Q/\nu^2 \end{aligned} \tag{3.57}$$

We can now create different δ functions by allowing different sets of the model decision variables to differ from zero. We gradually also decrease the δ_{max}, this is the maximum error allowed, from δ_1 to δ_3. We create three sets to be used in scenario 4, 5 and 6. The accuracy increases, but also does the complexity to compute these functions.

1. δ_1: $p_{Xa} = 0$, $p_{Xb} = f(\nu, \sigma)$ and $p_{Xc} = f(\nu, \sigma)$
   - This gives a linear relation between S_{cP} and S_{nl}
   - The parameters p_{Xb} and p_{Xc} only depend on values that are known up front, so $p_{b10} - p_{b13}$ and $p_{c12} - p_{c13}$ must be zero

- As such it can be easily integrated in optimization algorithms without increasing the algorithm complexity

2. δ_2: $p_{Xa} = f(\nu, \sigma)$, $p_{Xb} = f(\nu, \sigma)$ and $p_{Xc} = f(\nu, \sigma)$
   - This gives a quadratic relation between S_{cP} and S_{nl}
   - The parameters p_{Xa}, p_{Xb} and p_{Xc} only depend on values that are known up front, so $p_{a8} - p_{a9}$, $p_{b10} - p_{b13}$ and $p_{c12} - p_{c13}$ must be zero
   - It can be integrated in optimization algorithms, but it will require additional logic to solve
3. δ_3: $p_{Xa} = f(\nu, \sigma, r, Q)$, $p_{Xb} = f(\nu, \sigma, r, Q)$ and $p_{Xc} = f(\nu, \sigma, Q)$
   - This gives a quadratic relation between S_{cP} and S_{nl}
   - The parameters p_{Xa}, p_{Xb} and p_{Xc} also depend on r and Q, if these are unknown it increases complexity
   - It it not suggested to use this function for optimization algorithms
   - The main purpose of this function is to easily evaluate the KPI's if r and Q are known already

Next to the three previously defined scenarios in section 3.6.2, we have three additional scenarios:

- **Scenario 4**: Scenario 3 & an additional error reduction function δ_1
- **Scenario 5**: Scenario 3 & an additional error reduction function δ_2
- **Scenario 6**: Scenario 3 & an additional error reduction function δ_3

For the $\delta_{\overline{OF}}$ we only create two, as δ_3 makes no sense, since r has no impact and Q has already been implicitly included.

For δ_1, δ_2 and δ_3 we have created each time two sets of parameters (p_{Xa}, p_{Xb} and p_{Xc}). The first set of parameters, respectively called δ_{1a}, δ_{2a} and δ_{3a}, does not use all the allowed parameters, but only the most relevant. This has the advantage that fewer calculations are needed in (3.52)-(3.54), but it also has a small price on the accuracy. The second set of parameters does use all the allowed parameters, these are called: δ_{1b}, δ_{2b} and δ_{3b}. The list of all the parameters is given in appendix A.2.2. The links between the different scenarios and δ's are: Scenario 4a uses δ_{1a}, Scenario 4b uses δ_{1b}, Scenario 5a uses δ_{2a}, Scenario 5b uses δ_{2b}, Scenario 6a uses δ_{3a} and Scenario 6b uses δ_{3b}.

So we can conclude that in practice one has now the ability to continue to use the normal distribution (r, Q) equations and have a much better (r, Q) and (s, S) Poisson or compound Poisson KPI's approximation through a simple algebraic computation: (3.48)-(3.51). Per item and per KPI we once compute a set of three parameters (p_a, p_b and p_c, see (3.52-3.54) based upon the tables in appendix A.2.2.

3.6.4 Error reduction functions results

In Figure 3.27 we compare each of the maximum errors $\epsilon_{max|\overline{I}|}$, $\epsilon_{max|\overline{B}|}$ & $\epsilon_{max|S|}$ of scenario 3 with each of the new error reduction functions (scenario 4-6). The figure values are also available in appendix A.2.1. We observe:

- $\epsilon_{max|\overline{I}|}$: for (s, S) drops immediately from 19% to 7% in scenario 4. Scenario 6 decreases further to 4.5%.
- $\epsilon_{max|\overline{B}|}$: drops from over 20% to 10% in scenario 4, in scenario 6 maximum error further decreases to 4% for both (r, Q) and (s, S)
- $\epsilon_{max|S_3|}$: the maximum error gradually decreases from 9% (scenario 3) to 3% in scenario 6
- $\epsilon_{max|S_2|}$: drops immediately from over 14% (scenario 3) to 8% in scenario 4. For (r, Q) it further decreases to 5% in scenario 6, while for (s, S) it decreases to 7%
- $\epsilon_{max|S_{OL}|}$: immediately drops to half in scenario 4 (from 16% to 8%. It further decreases to 6% in scenario 6.

In Figure 3.28 the average errors $\epsilon_{avg|\overline{I}|}$, $\epsilon_{avg|\overline{B}|}$ & $\epsilon_{avg|S|}$ for scenario 3-6 are visualized.

- $\epsilon_{avg|\overline{I}|}$: for (s, S) drops from 3% (scenario 3) to 1.4% in scenario 4. It further decreases to 0.6% in scenario 6.
- $\epsilon_{avg|\overline{B}|}$: it drops from around 4% to +- 2.5% in scenario 4a. It gradually decreases towards 0.7% in scenario 6b.
- $\epsilon_{avg|S_3|}$: although already small (1.1% (r, Q) and 0.7% (s, S)), it steadily decreases towards 0.3% in scenario 6b
- $\epsilon_{avg|S_2|}$: drops immediately from 2% in scenario 3 to less than 1% in scenario 4a. From here it gradually further decreases towards 0.4% in scenario 6b
- $\epsilon_{avg|S_{OL}|}$: drops immediately from 2.4% in scenario 3 to 1.2% in scenario 4a. From here it gradually further decreases towards 0.6% in scenario 6b

As the $\epsilon_{avg|\overline{OF}|}$ and $\epsilon_{max|\overline{OF}|}$ is larger we will plot them in distinct Figures 3.29 and 3.30 for respectively the maximum and the average error.

- $\epsilon_{max|\overline{OF}|}$ goes up to 390%, which means that an uncorrected order frequency can be 5 times the correct one, this happens especially with $Q = 1$
- $\epsilon_{max|\overline{OF}|}$ can be reduced to 35% for scenario 5b
- $\epsilon_{avg|\overline{OF}|}$ is also larger than for the other KPI's, but we can reduce it from 12% in scenario 3 to less than 2% in scenario 5b

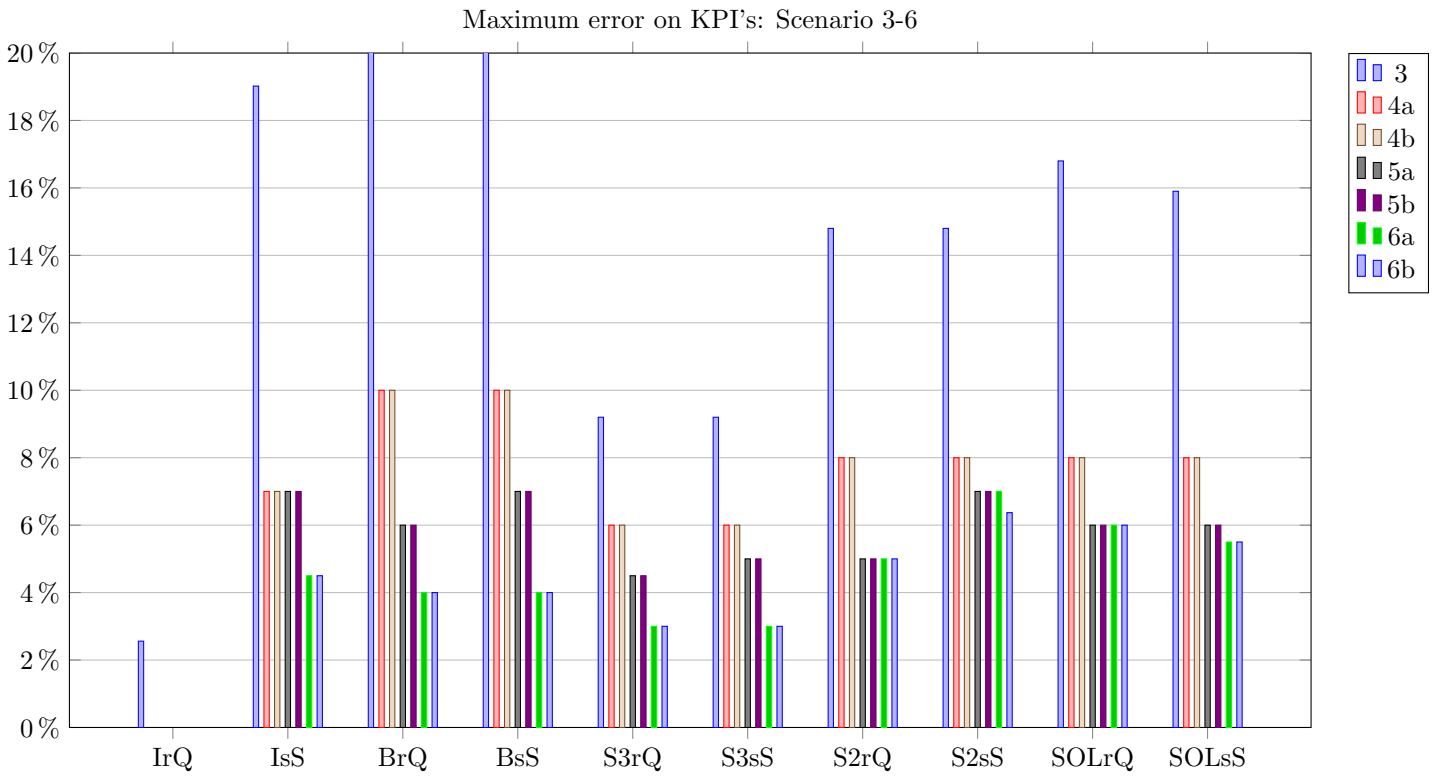

Figure 3.27: Maximum errors $\epsilon_{max|\overline{I}|}$, $\epsilon_{max|\overline{B}|}$ & $\epsilon_{max|S|}$: Scenarios 3-6

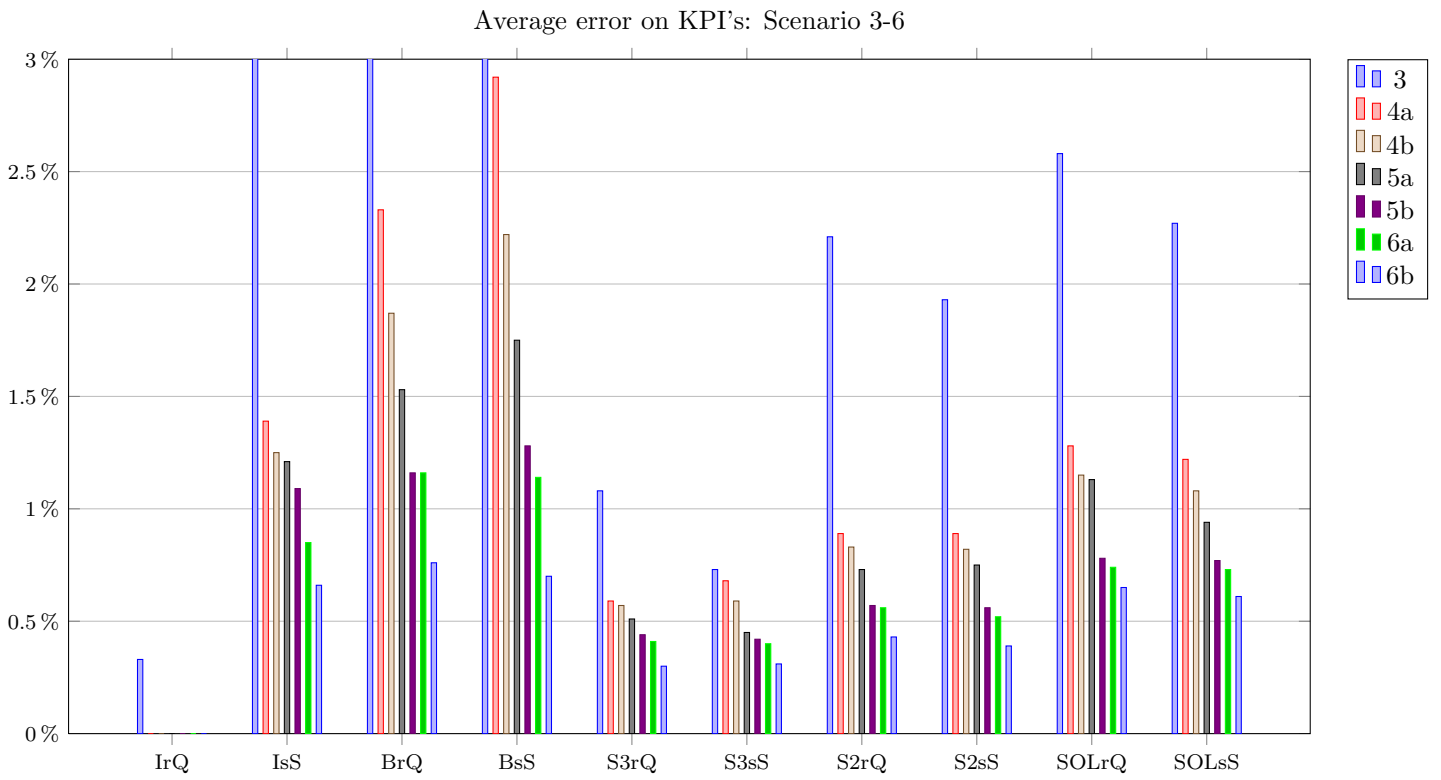

Figure 3.28: Average errors $\epsilon_{avg|\overline{I}|}$, $\epsilon_{avg|\overline{B}|}$ & $\epsilon_{avg|S|}$: Scenarios 3-6

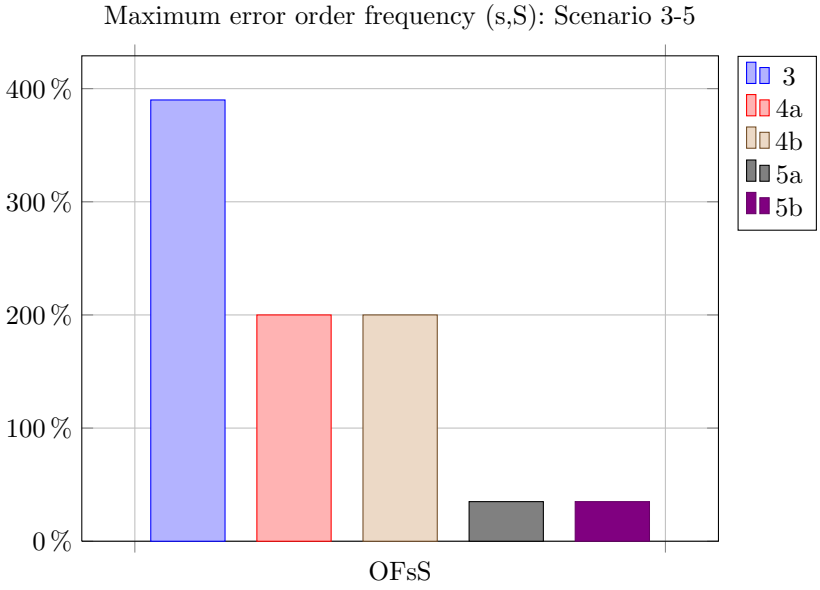

Figure 3.29: Maximum error $\epsilon_{max|\overline{OF}|}$: Scenarios 3-5

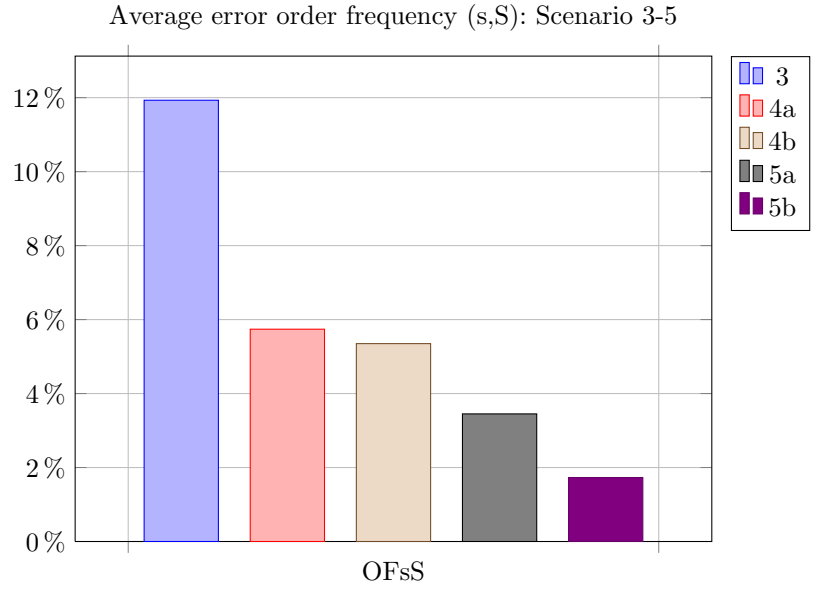

Figure 3.30: Average error $\epsilon_{avg|\overline{OF}|}$: Scenarios 3-5

Example 3.11 Example of error reduction functions

Let's now give an example how these functions can be applied. We take a problem that was not included in the data set for creation of the error reduction function. We do respect the assumption from section 3.6.1. We have the following parameters: $r = 12$, $Q = 10$, $\nu = 7$, $\sigma = 5.06$, $\chi = 3$. The client demand pattern is 20% chance to ask 1, 2, 3, 4 or 5 units.

	IrQ	IsS	BrQ	BsS	OFsS %
cP	10,7	11,2	0,16	0,14	61.8
1	10,1	10,1	0,10	0,10	
2	10,1	10,1	0,10	0,10	
3	10,6	10,6	0,08	0,08	70.0
4a		11,1	0,13	0,14	58.6
4b		11,1	0,12	0,12	59.3
5a		11,1	0,16	0,16	61.3
5b		11,1	0,16	0,14	62.1
6a		11,2	0,16	0,14	
6b		11,2	0,16	0,14	

	S3rQ %	S3sS %	S2rQ %	S2sS %	SOLrQ %	SolsS %
cP	94,5	95,0	91,7	92,6	90,2	91,1
1	95,7	95,7	95,7	95,7	95,7	95,7
2	95,7	95,7	95,7	95,7	95,7	95,7
3	96,4	96,4	96,4	96,4	96,4	96,4
4a	95,4	96,0	92,8	93,0	90,9	92,6
4b	95,5	96,0	93,0	93,5	91,7	92,6
5a	94,8	95,4	90,2	91,1	87,7	91,0
5b	94,9	95,4	91,0	92,3	89,3	90,1
6a	94,6	95,2	90,7	92,0	89,4	90,3
6b	94,6	95,1	91,1	92,3	89,3	90,5

Table 3.15: Example 3.11 KPI's and error reduction functions

Table 3.15 gives the KPI values for each of the scenarios. We can compare it with the exact compound Poisson (cP) values, listed on the first line of the table. In Figure 3.31 we can see the impact on the errors $\epsilon_{\overline{I}}$, $\epsilon_{\overline{B}}$ and ϵ_S, see (3.26)-(3.30), for Example 3.11. As we do not show the absolute error, we can now see that the approximation can be an under or overestimation. We can also clearly see that the error decreases going from scenario 1 up to 6. Per instance there may be some deviations, we notice here for example that scenario 5a gives a larger error than 4a and 4b for S2sS.

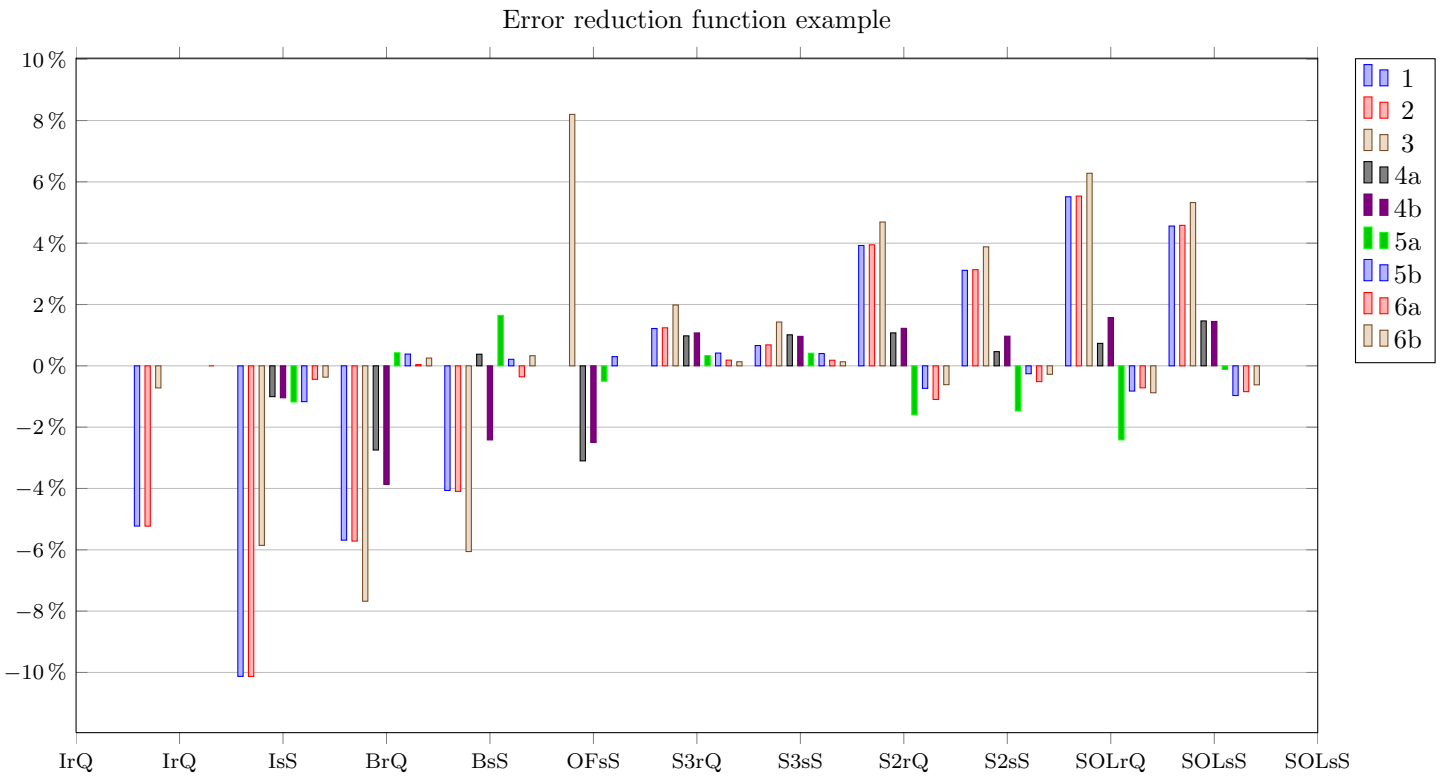

Figure 3.31: Ex. 3.11: $\epsilon_{\overline{I}}$, $\epsilon_{\overline{B}}$, $\epsilon_{\overline{OF}}$ and ϵ_S & impact error reduction functions

3.7 New contributions

In this chapter we have made new contributions on the following three topics:

1. Order line service level
2. Corrected and conditions for simplified (r, Q) KPI equations
3. (r, Q) and (s, S) KPI error analysis and error reduction functions

We developed an explicit equation for an order line service level in case of a base-stock, an (r, Q) and an (s, S) policy with compound Poisson demand. The order line service level is important as it is often used in practice, while no previous definition or equation exists for compound Poisson in other works or papers. See contributions 2, 3 and 4.

We worked out a corrected set of normal demand (r, Q) policy KPI's. For this we make a 0.5 correction in the equations, replacing r with $r + 0.5$. We showed the direct beneficial impact on the KPI's, especially on average inventory and average backorders. The direct benefits were greatest with small ν. See contribution 5.

We developed a set of conditions for a normal demand (r, Q) policy, to know when it is safe to use simpler equations, allowing a relative error of $1e-3$ compared to the complete formula. In practice the simpler equations are widespread used leading to large errors in some cases. See contribution 6.

We provided an analysis of the approximation errors while using a normal demand approximation for a Poisson or compound Poisson demand. We did this for three replenishment policies: base-stock, (r, Q) and (s, S). We saw very large errors for order line service level up to 60% for small leadtime demand and up to 15% for larger leadtime demand. See contribution 7. We created significantly improved approximation functions for the Poisson and compound Poisson exact KPI's based upon a normal distribution. This has the major benefit that we can still apply the simpler normal demand functions while having acceptable and significantly reduced errors. The average error in our realistic data set drops from more than 2% to 0.5% and less. The maximum error drops from 40%-50% to 5%. See contribution 8.

3.8 Conclusions

The replenishment rate S_1 has no valuable meaning and should not be used in practice.

It is now possible to calculate the order line service level in case of compound Poisson demand.

The normal demand (r, Q) inventory KPI is more accurate if a 0.5 correction is applied.

Using simplified normal demand (r, Q) equations can lead to tremendous errors, a simple set of conditions need to be checked first.

Blindly using the normal distribution demand to replace the Poisson or compound Poisson demand can lead to maximum errors of 50% for average inventory, 25% for average backorders, 10% on the ready rate, 15% on the fill rate and 17% on the order line service level for both the (r, Q) and (s, S) replenishment policies. The order frequency shows a maximum error of nearly 400% in case of (s, S) policy with small $S - s$.

Making use of simple error reduction functions, that are a small algebraic computations based on the normal distribution equations, we see an immense reduction of these maximum errors: a reduction from 50% down to 5% for average inventory, from 25% down to 4% for average backorders, from 10% down to 3% on the ready rate, from 15% down to 6% on the fill rate and from 17% down to 6% on the order line service level for both the (r, Q) and (s, S) replenishment policy. The (s, S) order frequency maximum error drops from 400% down to 35%.

Using the full set of error reduction functions the average error on inventory, backorders, ready rate, fill rate an order line service level can be reduced to less than 1%!

4

Best approximations normal loss functions

We present double precision algorithms based upon rational approximations for the standard normal first and second order loss functions and their inverse functions. These functions are used frequently in inventory management. No direct approximations or closed-form expressions exist for the first and second order normal loss functions and their inverse functions. Calculations are currently based on intermediate computations of the cumulative normal distribution or tabulations and results depend on the accuracy and valid range of these underlying functions. We deal with these issues and present direct, double precision accurate algorithms valid in the full range of double precision floating point numbers.

4.1 Introduction and motivation

Let φ be the standard normal probability density function, Φ the standard normal cumulative distribution and Φ^0 its complementary function, as respectively defined by (4.1), (4.2) and (4.3a). Then Φ^1 is the standard normal first order loss function, see (4.4a) and Φ^2 is the standard normal second order loss function, see (4.5a).

The inverse of the standard normal cumulative distribution, Φ^{0inv}, is defined by (4.3b). Wichura (1988) provides an approximation algorithm for Φ^{0inv}. The inverse standard normal first order loss function is Φ^{1inv}, (4.4b) and the inverse standard normal second order loss function is Φ^{2inv}, (4.5b). More details on these functions can be found in section 2.1.4.

$$\varphi(z) = \frac{exp\left(-z^2/2\right)}{\sqrt{2\pi}} \tag{4.1}$$

$$\Phi(z) = \int_{-\infty}^{z} \varphi(x)\, dx \tag{4.2}$$

$$\Phi^0(z_{p0}) = \int_{z_{p0}}^{\infty} \varphi(x)\, dx = p_0 \tag{4.3a}$$

$$\Phi^{0inv}(p_0) = z_{p0} \tag{4.3b}$$

$$\Phi^1(z_{p1}) = \int_{z_{p1}}^{\infty} \Phi^0(x)\, dx = p_1 \tag{4.4a}$$

$$\Phi^{1inv}(p_1) = z_{p1} \tag{4.4b}$$

$$\Phi^2(z_{p2}) = \int_{z_{p2}}^{\infty} \Phi^1(x)\, dx = p_2 \tag{4.5a}$$

$$\Phi^{2inv}(p_2) = z_{p2} \tag{4.5b}$$

The cumulative normal distribution and its approximation have been studied in several papers. Waissi and Rossin (1996) have presented a simple sigmoid function for the approximation of the cumulative standard normal probabilities for $-8 \leq z \leq 8$. Bryc (2002) presented two simple formulas for the approximation of the standard normal right tail probabilities. Kiani et al. (2008) worked out a closed-form expression and also a series approach for approximating the normal distribution. This formula has a maximum absolute error of 6.5e-9 and the series have a very high accuracy over the

whole range. Linhart (2008) compared three C functions to compute the logarithm of the cumulative standard normal distribution, based upon existing algorithms.

In Figure 4.1 Φ^0, Φ^1 and Φ^2 are shown for z in the range [0,3]. Figure 4.2 gives the functions Φ^{0inv}, Φ^{1inv} and Φ^{2inv} over the range [0.001, 0.999] and Figure 4.3 plots Φ^{1inv} and Φ^{2inv} over the range [0.001, 3].

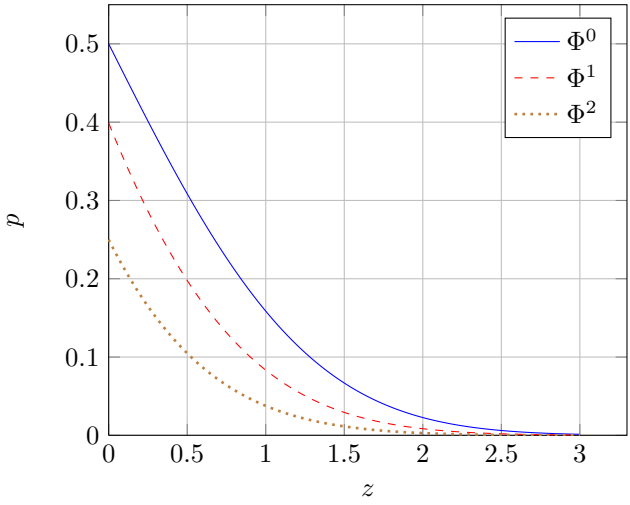

Figure 4.1: Φ^0, Φ^1 and Φ^2 over range [0, 3]

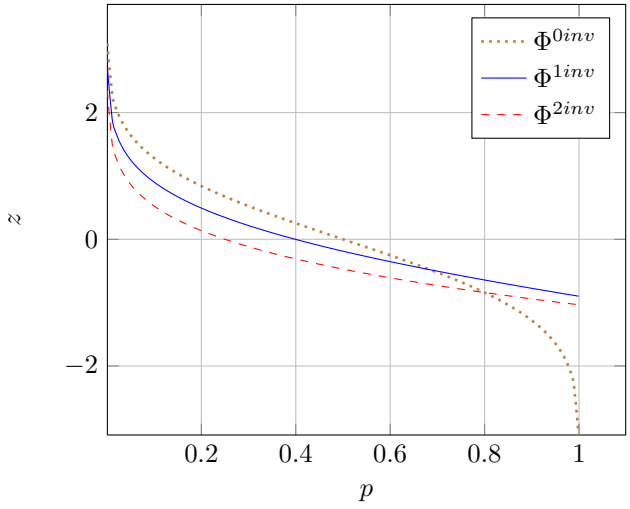

Figure 4.2: Φ^{0inv}, Φ^{1inv} and Φ^{2inv} over range [0.001, 0.999]

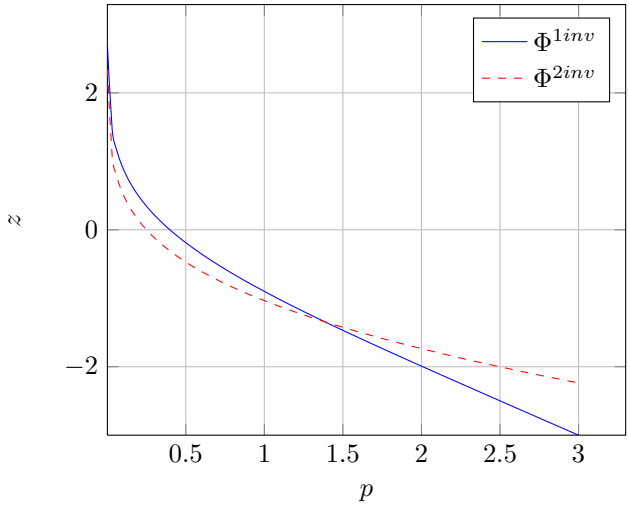

Figure 4.3: Φ^{1inv} and Φ^{2inv} over range [0, 3]

Closed-form expressions for computing Φ^1 and Φ^2 do not exist. Calculating the values of the standard normal first and second order loss function can be done using the cumulative normal distribution, see (4.6) and (4.7), which on its turn could be based on the cumulative error function. So currently an intermediate step is needed to calculate Φ^1 and Φ^2. The accuracy of the result depends on the accuracy and valid range of the underlying cumulative normal function (2.39).

For the calculation of Φ^{1inv} and Φ^{2inv} a root-finding method is necessary on top of the functions (4.6) and (4.7) to calculate Φ^{1inv} or Φ^{2inv}. With the approximations developed here we want to deal with these issues.

$$\Phi^1(z) = -z\Phi^0(z) + z\varphi(z) \tag{4.6}$$

$$\Phi^1(z) = \frac{1}{2}\left[(z^2+1)\Phi^0(z) - z\varphi(z)\right] \tag{4.7}$$

Within inventory management Φ^1, Φ^2, Φ^{1inv} and Φ^{2inv} are needed to calculate the KPI's in a base-stock policy and a (r, Q) replenishment policy with normal distribution demand, see section 2.3.4.3 and 2.3.6.3. As these computations are needed on a large scale in inventory problems with a huge amount of items, we see the direct benefit of having a highly efficient and effective approximation within the range and the accuracy of double precision floating point numbers. Other statistical applications where these and other repeated integrals can be used are listed in Withers and Nadarajah (2010):

- Calculation of moments of truncated normal distribution
- Expression of the non-central t density
- Calculation of shape distributions

As we want to cover the range of double precision floating point values of the IEEE 754 standard, see Goldberg (1991), the smallest positive value that can be represented is $2.2250738585072014e-308$. Lower values are subnormal and thus cannot be represented to full precision. The maximum double value is $1.7976931348623157e+308$. The considered range for our approximations is $[2.2250738585072014e-308, 1.7976931348623157e+308]$.

4.2 Repeated integrals of the univariate normal distribution

We first describe a Taylor series function for Φ^1 and Φ^2 with managed precision used as the foundation for the approximation with double precision of Φ^1, Φ^2, Φ^{1inv} and Φ^{2inv}. Then a rational approximation is generated, making use of the Remez minimax C++ implementation, see Boost (2011). Withers and Nadarajah (2010) extended the divergent expansion for Mill's ratio for repeated integrals of the univariate normal distribution and Taylor series are also presented. It is shown that the error approximated by the first k terms of its Laplace-type approximation is bounded in magnitude by the k^{th} term. Φ^1 and Φ^2 can be translated in the following Taylor series:

$$\Phi^1(z) = \varphi(z) \sum_{j=0}^{\infty} \frac{(-z)^j t_{j+1}}{j!} \tag{4.8}$$

$$\Phi^2(z) = \frac{\varphi(z)}{2} \sum_{j=0}^{\infty} \frac{(-z)^j t_{j+2}}{j!} \tag{4.9}$$

where

$$t_0 = \sqrt{\frac{\pi}{2}} \tag{4.10a}$$

$$t_1 = 1 \tag{4.10b}$$

$$t_k = 1 * 3 * ...(k-3)(k-1)t_0, \textit{ for k even} \tag{4.10c}$$

$$t_k = 2 * 4 * ...(k-3)(k-1), \textit{ for k odd} \tag{4.10d}$$

Equations (4.8) and (4.9) are used to calculate very high precision Φ^1 and Φ^2 values, with a relative error magnitude of less than 1e-32. Validation sets are created for 1 000, 10 000, 100 000 and 1 000 000 random z values

with this accuracy, to be used to test our approximation algorithms. The necessary number of terms in (4.8) and (4.9) to achieve a maximum ϵ_{rel} is limited to k, defined for Φ^1 and Φ^2 respectively by condition (4.11) and (4.12):

$$\epsilon_{rel} = |\left(\frac{(-z)^{k+1} t_{k+2}}{(k+1)!} \right) / \left(\sum_{j=0}^{k} \frac{(-z)^j t_{j+1}}{j!} \right)| \leq 1e-32 \tag{4.11}$$

$$\epsilon_{rel} = |\left(\frac{(-z)^{k+1} t_{k+3}}{(k+1)!} \right) / \left(\sum_{j=0}^{k} \frac{(-z)^j t_{j+2}}{j!} \right)| \leq 1e-32 \tag{4.12}$$

4.3 Developing rational approximations

The development of a rational approximation with double precision, making use of double precision numbers, is a cumbersome and time demanding task. The creation of the four approximations described in this chapter has taken over a year before the current accuracy and efficiency was reached. In order to end up with this approximation we have to walk through a series of steps, but it also involves a lot of iterations. An introduction to best approximation functions is given in 2.2 and the Remez algorithm is described in A.1. We describe the steps needed to develop a rational minimax approximation making use of Remez algorithm.

1. Decide which function you want to approximate.
2. Decide on the range over which you want to approximate it: we want to cover the full range of double precision numbers.
3. Decide on the accuracy of the approximation: we want to end up with double precision accuracy.
4. Decide on the type of error: absolute or relative: we choose to minimize the relative error.
5. Have an exact description of the function we want to approximate over the range we want to approximate it. If no exact functions exists, then there should be a function with at least twice the precision of our target approximation precision, so in our case we need an 'exact' function with a precision of 1e-32 (see section 4.2). For this we use an arbitrary precision floating point library, Shoup (2011). Creating this 'exact' function was an approximation on its own, but then of a much higher accuracy. Here we did not have the burden of limiting number accuracy or we did not need to limit the number of regions or number of

parameters. On the other hand we needed to use the iterative formulation (4.8) and (4.9), which had a very high computation effort and was thus slow. After creating sets of rational approximations using the Remez algorithm with Boost (2011) for Φ^1 and Φ^2 with a very high accuracy, we also created very high precision approximations for the inverse functions Φ^{1inv} and Φ^{2inv} making use of a bisection root-finding method.

6. Choose the number of regions and the range of each region. We want to minimize the number of regions. The regions have preferably a small overlap, as we know that errors tend to enlarge near the borders
7. Find an appropriate 'dominant' function that 'looks like' the function we want to approximate in the considered region. Here it is no longer an exact science. It involves a lot of trial and error, rescaling the X and Y axis, shifting the axes, using a logarithmic scale, skewing the initial control points, ... Quite often we find here we need to split up certain regions (again) or redefine them, so go back to step 6. So here lays the 'craftsmanship' to 'persuade' the Remez algorithm to converge with the target accuracy. Some functions show divergent behavior and need to be adjusted first. Also we need to be very careful to always use numbers that can be represented exactly in double precision format. Sometimes the Remez algorithm solution is mathematically sound, but is useless in practice due to too much rounding off on the errors or because of roots in the denominator. When we believe we have covered the whole range, using several regions, we need to test the approximation and go to step 8.
8. Testing. Here we are looking for errors due to roots in the denominator or due to the double precision calculation, while within the Remez algorithm we applied a far more accurate precision (1e-32). When encountering these issues, we need to go back to step 6. We created test sets with 1 000, 10 000, 100 000 and one million instances that have random input values over the whole range that can be represented exactly using double number precision, e.g. the number 1 can be represented exactly, the number 0.1 not. If no errors are found we can go to step 9.
9. Stop

4.4 SN1OLF: Φ^1 Approximation

Contribution 9: SN1OLF algorithm

In this section we will create an algorithm that allows a one pass double precision accurate calculation of $\Phi^1(z)$ over the full range of double precision floating point numbers, see De Schrijver et al. (2011c). Figure 4.4 shows $\Phi^1(z)$ in the range [-1,3].

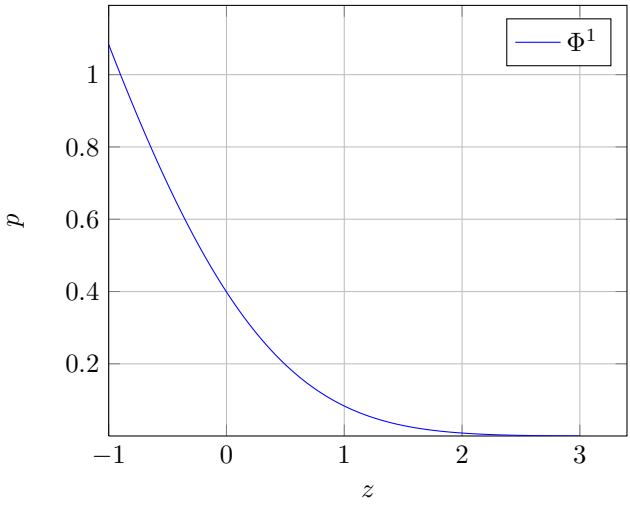

Figure 4.4: Φ^1 over range [-1, 3]

4.4.1 SN1OLF Algorithm

As there is a direct relation between $\Phi^1(z)$ and $\Phi^1(-z)$, we will only consider the positive range for the approximation algorithm. This relation is given by (4.13) for calculating Φ^1 in case of negative z values. We also know that $\Phi^1\left(7.96582630953042053\right) = 1e-16$, so for $z > 7.96582630953042053$ we can further simplify (4.13) into (4.14).

$$\Phi^1\left(-z\right) = \Phi^1\left(z\right) + z \tag{4.13}$$

$$\Phi^1\left(-z\right) = z, if\ z > 7.96582630953042053 \tag{4.14}$$

For the approximation of the standard normal first order loss function Φ^1 we have defined three regions. The maximum value of the third region, 37.4227413752986, yields the smallest value that can be represented in double precision: $2.2250738585072014e-308$, see equation (4.15).

$$\phi^1(37.4227413752986) = 2.2250738585072014e - 308 \tag{4.15}$$

SN1OLF range	from	to
1	0	2
2	2	5
3	5	37.4227413752986

Table 4.1: SN1OLF algorithm ranges

For each of the three regions we use a dominant function $exp(-0.5r^2)$ as a basis, where $r = -z$ if z is negative and else $r = z$. We also perform a correction in the calculation of $exp(-0.5r^2)$, based upon a cut-off to allow exact multiplication, see (4.16) - (4.18). A likewise correction was also used in the approximation of the cumulative normal distribution, the pnorm function of the R project, see R (2011). This pnorm function was based upon Cody (1993). In each of the three regions we use a different formulation for the abscissa. For z values larger than the 37.4227413752986, the algorithm returns zero as Φ^1 value. For negative z values (4.13) is used in the algorithm.

$$q = \frac{\lfloor 64r \rfloor}{64} \tag{4.16}$$

$$d = (r - q)(r + q) \tag{4.17}$$

$$e = exp(-0.5q^2)exp(-0.5d) \tag{4.18}$$

SN1OLF Algorithm

The SN1OLF acronym stands for standard normal first order loss function. The SN1OLF rational functions and its parameters are given in Appendix A.3.

1. If $z < -7.96582630953042053$ then $p = 0$ and go to step 7 else go to step 2
2. If $z < 0$ then $r = -z$ else $r = z$, go to step 3
3. If $r \leq 2$: with $x = r$, set $p =$ (A.4), go to step 7
4. If $r \leq 5$: with $x = r - 2$, set $p =$ (A.5), go to step 7
5. If $r \leq 37.4227413752986$: with $x = 1/r^2$, set $p =$ (A.6), go to step 7
6. If $r > 37.4227413752986$: $p = 0$, go to step 7
7. If $z < 0$ then $\Phi^1(z) = p - z$ else $\Phi^1(z) = p$

4.4.2 SN1OLF Algorithm validation

We use four validation sets with respectively 1 000, 10 000, 100 000 and 1 million random z points uniformly distributed in the range $[0, 37.4227413752986]$. The z values in these data sets were chosen such that it are exact double precision values. These $\Phi^1(z)$ test values have a relative error (4.19) of less than 1e-32. For each of the sets we give the maximum magnitude of the relative error (4.20) of algorithm SN1OLF and the RMS, root mean square, (4.21) of the relative error, see Table 4.2.

$$\epsilon_{rel} = \frac{\Phi^1(z) - SN1OLF(z)}{|\Phi^1(z)|} \tag{4.19}$$

$$Max(|\epsilon_{rel}|) = Max(|\epsilon_{rel1}|, |\epsilon_{rel2}|, ..., |\epsilon_{reln}|) \tag{4.20}$$

$$RMS(\epsilon_{rel}) = \sqrt{\frac{1}{n}(\epsilon_{rel1}^2 + \epsilon_{rel2}^2 + ... + \epsilon_{reln}^2)} \tag{4.21}$$

Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
1 000	5.33e-16	1.60e-16
10 000	6.10e-16	1.62e-16
100 000	6.70e-16	1.62e-16
1 000 000	7.04e-16	1.62e-16

Table 4.2: Magnitude relative error SN1OLF

In Figure 4.5 we plot the relative error ϵ_{rel} of the SN1OLF algorithm for the validation set with 1 000 points.

4.4.3 SN1OLF test data for algorithm

Table 4.3 gives values that may be used to check whether the SN1OLF algorithm has been correctly implemented.

z	$p =$ SN1OLF
1	$8.331547058768629e-002$
10	$7.474560254589329e-025$
37	$1.545199190512203e-301$

Table 4.3: Test data for algorithm SN1OLF

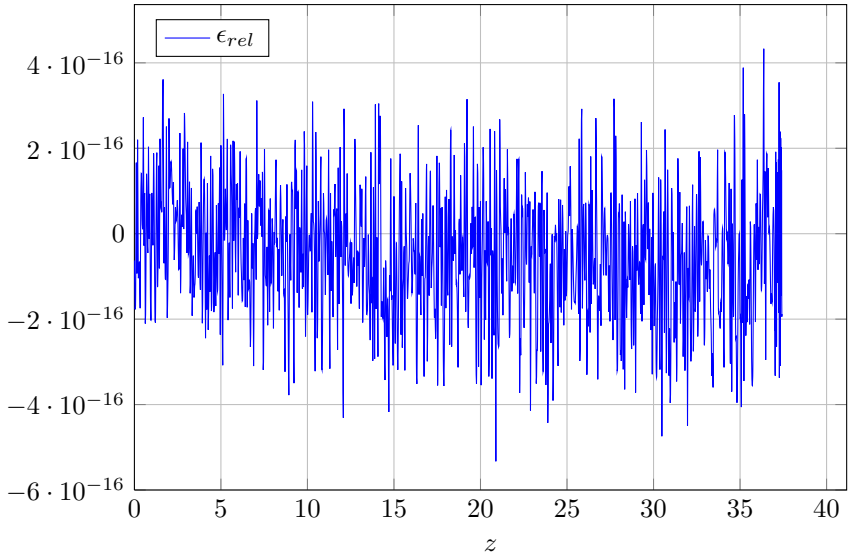

Figure 4.5: SN1OLF relative error (1 000 test points)

4.5 SN2OLF: Φ^2 Approximation

Contribution 10: SN2OLF algorithm

In this section we will create an algorithm that allows a one pass double precision accuracy calculation of $\Phi^2(z)$ over the full double range of of double precision floating point numbers, see De Schrijver et al. (2011c). Figure 4.6 shows $\Phi^2(z)$ in the range [0,3].

4.5.1 SN2OLF Algorithm

As there is also a direct relation between $\Phi^2(z)$ and $\Phi^2(-z)$, we will only consider the positive range for the approximation algorithm. This relation is given by (4.22) for calculating Φ^2 in case of negative z values. We also know that $\Phi^2\left(7.707353552279367\right) = 1e-16$, so for $z > 7.707353552279367$ we can further simplify (4.22) into (4.23).

$$\Phi^2\left(-z\right) = -\Phi^2\left(z\right) + \left(\frac{z^2+1}{2}\right) \tag{4.22}$$

$$\Phi^2\left(-z\right) = \frac{z^2+1}{2}, when\ z > 7.707353552279367 \tag{4.23}$$

For the approximation of the standard normal second order loss function Φ^2 we have defined three regions. The maximum value of the third range,

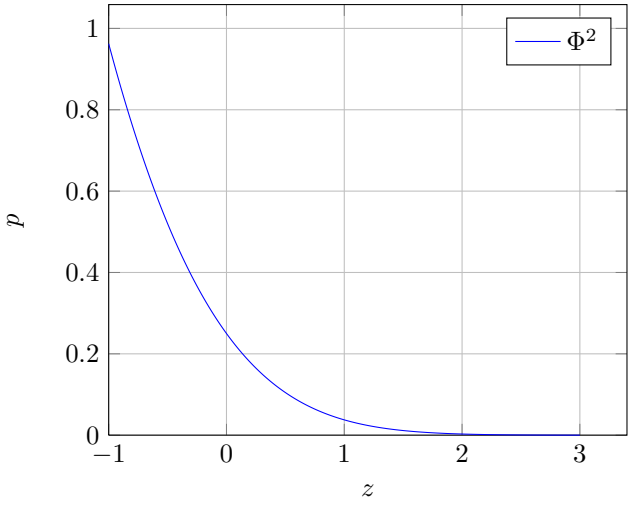

Figure 4.6: Φ^2 over range [-1, 3]

37.3259727019187, yields the smallest value that can be represented in double precision: $2.2250738585072014e-308$, (4.24).

$$\Phi^2(37.3259727019187) = 2.2250738585072014e-308 \tag{4.24}$$

SN2OLF range	from	to
1	0	2
2	2	5
3	5	37.3259727019187

Table 4.4: SN2OLF algorithm ranges

For the first and the second range we use a dominant function $exp(-0.5r^2)$ as a basis, for the third range we use $exp(-0.5r^2)/r$. In each of the three regions we use a different formulation for the abscissa. For z values larger than the 37.3259727019187, the algorithm returns zero as Φ^2 value. For negative z values (4.22) is used in the algorithm.

SN2OLF Algorithm

The SN2OLF acronym stands for standard normal second order loss function. The SN2OLF rational functions and its parameters are given in Appendix A.4.

1. If $z < -7.707353552279367$ then $p = 0$ & go to step 7 else go to step 2
2. If $z < 0$ then $r = -z$ else $r = z$, go to step 3
3. If $r \leq 2$: with $x = r$, set $p =$ (A.7), go to step 7
4. If $r \leq 5$: with $x = r - 2$, set $p =$ (A.8), go to step 7
5. If $r \leq 37.3259727019187$: with $x = 1/r^2$, set $p =$ (A.9), go to step 7
6. If $r > 37.3259727019187$: $p = 0$, go to step 7
7. If $z < 0$ then $\Phi^2(z) = -p + 0.5(z^2 + 1)$ else $\Phi^2(z) = p$

4.5.2 SN2OLF Algorithm validation

We used four validation sets with respectively 1 000, 10 000, 100 000 and 1 million random z points uniformly distributed over the broad range defined by $[0, 37.4227413752986]$. These Φ^2 validation values have a relative error (4.25) of less than 1e-32. For each of the sets we give the maximum magnitude of the relative error (4.26) of algorithm SN2OLF and the RMS (4.27) of the relative error, see Table 4.5. The definition of ϵ_{rel} for SN2OLF is given by (4.25). $Max(\epsilon_{rel})$ is given by 4.26) and $RMS(\epsilon_{rel})$ by (4.27).

$$\epsilon_{rel} = \frac{\Phi^2(z) - SN2OLF(z)}{|\Phi^2(z)|} \tag{4.25}$$

$$Max(|\epsilon_{rel}|) = Max(|\epsilon_{rel1}|, |\epsilon_{rel2}|, ..., |\epsilon_{reln}|) \tag{4.26}$$

$$RMS(\epsilon_{rel}) = \sqrt{\frac{1}{n}(\epsilon_{rel1}^2 + \epsilon_{rel2}^2 + ... + \epsilon_{reln}^2)} \tag{4.27}$$

Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
1 000	5.63e-16	1.64e-16
10 000	6.15e-16	1.61e-16
100 000	6.74e-16	1.62e-16
1 000 000	7.39e-16	1.61e-16

Table 4.5: Magnitude relative error SN2OLF

In Figure 4.7 we plot the relative error ϵ_{rel} of the SN2OLF algorithm, defined by (4.25).

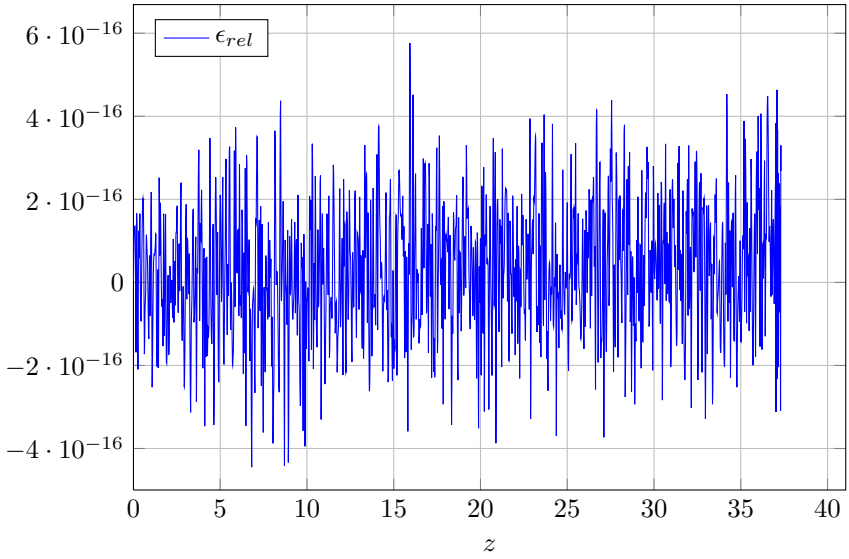

Figure 4.7: SN2OLF relative error (1 000 test points)

4.5.3 SN2OLF test data for algorithm

Table 4.6 gives values that may be used to check whether the SN2OLF algorithm has been correctly implemented.

z	$p = \text{SN2OLF}$
1	$3.766989167188537e-002$
10	$7.264638478559902e-026$
37	$4.167108814713861e-303$

Table 4.6: Test data for algorithm SN2OLF

4.6 ISN1OLF: Φ^{1inv} Approximation

Contribution 11: ISN1OLF algorithm

In this section we will create an algorithm that allows a one pass double precision accuracy calculation of $\Phi^{1inv}(p)$ over the full range of double precision floating point numbers, see De Schrijver et al. (2011b). Figure 4.8 shows $\Phi^{1inv}(p)$ in the range [0,2].

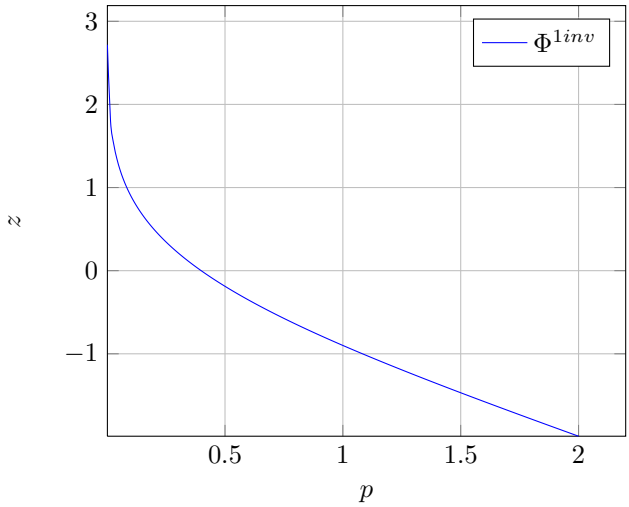

Figure 4.8: Φ^{1inv} over range [0, 2]

4.6.1 Φ^{0inv} and Φ^{1inv} approximation differences

In comparison with the approximation of Φ^{0inv}, see Wichura (1988), Φ^{1inv} has two additional difficulties.

First, the Φ^{1inv} root (4.28a) is not an exact double value, where for Φ^{0inv} the root is an exact double: 0.5. The Φ^{1inv} root value is expressed with 32 digits in (4.28b). The most accurate double presentation is (4.28c), which is greater than the actual root value. The two values with an exact double representation closest to the root are (4.28c) and (4.28d), respectively to the right and to the left of the root. These values will be used in ranges close to the root, as we do not want the algorithm to depend on the representation of π, and its accuracy, within the used programming language.

$$\Phi^{1}(0) = \frac{1}{\sqrt{2\pi}} \Rightarrow \Phi^{1inv}\left(\frac{1}{\sqrt{2\pi}}\right) = 0 \tag{4.28a}$$

$$= 0.39894228040143267793994605993 \tag{4.28b}$$

$$< 0.398942280401432702 \tag{4.28c}$$

$$> 0.398942280401432647 \tag{4.28d}$$

The second difference is that Φ^{0inv} has a closed-form expression for the range left and right of its root, (4.29). Although there is a closed-form expression for positive and negative values of Φ^1, (4.30), this cannot be converted into a closed-form expressions for Φ^{1inv} values left and right of its root. So the range covered by the Φ^{1inv} approximation should not only cover from $2.2250738585072014e-308$ to the root, but from $2.2250738585072014e-308$ to $1.7976931348623157e+308$.

$$\Phi^{0inv}(1-p) = -\Phi^{0inv}(p), p \in [0,1] \tag{4.29}$$

$$\Phi^{1}(-z_p) = \Phi^{1}(z_p) + z_p \tag{4.30}$$

Although (4.30) does not give a closed-form expression for values to the right of the root for Φ^{1inv}, it enables an approximation for p values larger than 7.96582630953042053. As $\Phi^1(7.96582630953042053) = 1e-16$, (4.30) can be approximated by (4.31) in case of double precision calculations. As such Φ^{1inv} can be approximated by (4.32) in double precision for p values larger than 7.96582630953042053.

$$\Phi^{1}(-z_p) = z_p, \ \textit{when } z_p > 7.96582630953042053 \tag{4.31}$$

$$\Phi^{1inv}(p) = -p, \ \textit{when } p > 7.96582630953042053 \tag{4.32}$$

4.6.2 ISN1OLF Algorithm

The list below gives the parameters used in the approximation and its algorithm.

$r_0 = 2.2250738585072014e-308$ $r_1 = 7.96582630953042053$
$r_2 = 1.1875$ $r_3 = 0.5234375$
$r_4 = 0.6640625$ $r_5 = 0.398942280401432702$
$r_6 = 0.398942280401432647$ $r_7 = 0.367692280401432647$
$r_8 = 0.102067280401432647$ $r_9 = 0.296875$
$r_{10} = 1.7976931348623157e+308$

Here we describe algorithm ISN1OLF, given a value p, it will compute $z = \Phi^{1inv}(p)$ with double precision in the range $[r_0, r_{10}]$. The acronym ISN1OLF stands for 'inverse standard normal first order loss function'. The

valid range is divided in 8 sub ranges, 4 left of the root and 4 to the right of the root. For p values close to zero we first perform a conversion based upon the square root and the natural logarithm of p.

ISN1OLF Algorithm

The rational functions and its parameters are given in Appendix A.5.

1. If $p > r_1$ set $z = -p$, go to step 10
2. If $p \geq r_2$: with $x = (p - r_2)$, set $z =$ (A.10), go to step 10
3. If $p \geq r_3$: with $x = (p - r_3)\,/r_4$, set $z =$ (A.11), go to step 10
4. If $p \geq r_5$: with $x = (p - r_5) * 8$, set $z =$ (A.12), go to step 10
5. If $p \geq r_7$: with $x = (r_6 - p) * 32$, set $z =$ (A.13), go to step 10
6. If $p \geq r_8$: with $y = (r_6 - p)$, $x = (r_9 - y)$, set $z =$ (A.14), go to step 10
7. Set $y = \sqrt{\ln(-p)}$
8. If $y \leq 6$: with $x = (y - 1.5)$, set $z =$ (A.15), go to step 10
9. Else: with $x = (y - 6)$, set $z =$ (A.16), go to step 10
10. $\Phi^{1inv}(p) = z$

4.6.3 ISN1OLF Algorithm validation

We used four validation sets with respectively 1 000, 10 000, 100 000 and 1 million random p points uniformly distributed in the range $[1e-100, 8]$. The Φ^{1inv} values have a relative error (4.33) of less than 1e-32. For each of the sets we give the maximum magnitude (4.20) of the relative error of algorithm ISN1OLF and the RMS (4.35) of the relative error, see Table 4.7.

$$\epsilon_{rel} = \frac{\Phi^{1inv}(p) - ISN1OLF(p)}{|\Phi^{1inv}(p)|} \tag{4.33}$$

$$Max(|\epsilon_{rel}|) = Max(|\epsilon_{rel1}|, |\epsilon_{rel2}|, ..., |\epsilon_{reln}|) \tag{4.34}$$

$$RMS(\epsilon_{rel}) = \sqrt{\frac{1}{n}(\epsilon_{rel1}^2 + \epsilon_{rel2}^2 + ... + \epsilon_{reln}^2)} \tag{4.35}$$

Range	Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
$[1e-100, 8]$	1 000	3.95e-16	6.08e-17
$[1e-100, 8]$	10 000	4.62e-16	6.05e-17
$[1e-100, 8]$	100 000	5.77e-16	6.02e-17
$[1e-100, 8]$	1 000 000	6.33e-16	6.05e-17

Table 4.7: Magnitude relative error

In Figure 4.9 we plot the relative error ϵ_{rel} of the ISN1OLF algorithm.

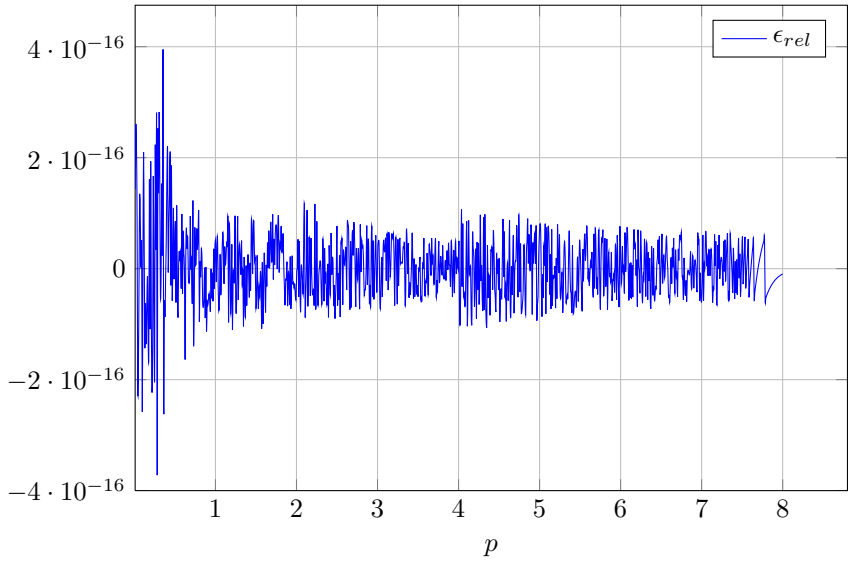

Figure 4.9: ISN1OLF relative error (1 000 test points)

In Table 4.8 we test the values close to $p = 0$. Again we used four validation sets with respectively 1 000, 10 000, 100 000 and 1 million random p points and Φ^{1inv} values that have an error of less than 1e-32, but now the random p values are uniformly distributed on a logarithmic scale within the considered range. We again make sure that the z values can be exactly represented by a double precision number.

Range	Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
$[r_0, r_8]$	1 000	6.30e-16	1.86e-16
$[r_0, r_8]$	10 000	7.16e-16	1.85e-16
$[r_0, r_8]$	100 000	7.99e-16	1.80e-16
$[r_0, r_8]$	1 000 000	9.47e-16	1.86e-16

Table 4.8: Magnitude relative error ISN1OLF: p close to 0 on a logarithmic scale

In Figure 4.10 we plot the relative error ϵ_{rel} of the ISN1OLF algorithm on a logarithmic scale. This allows us to check the precision for p values close to zero where Φ^{1inv} increases rapidly.

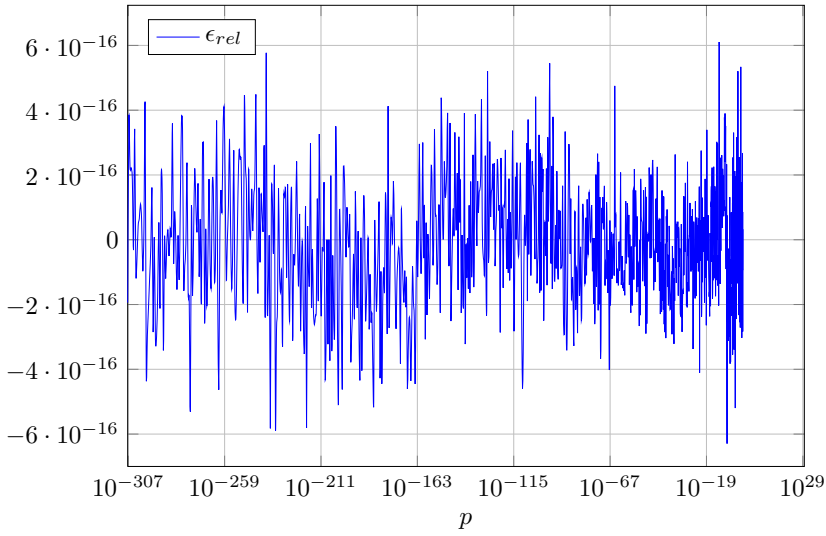

Figure 4.10: ISN1OLF relative error (1 000 test points on logarithmic scale)

4.6.4 ISN1OLF test data for algorithm

Table 4.9 gives values that may be used to check whether the algorithm has been correctly implemented.

p	$z = \text{ISN1OLF}$
$1e-100$	21.12967328021652
0.25	0.3448674639990244
1	−0.8994715612537435

Table 4.9: Test data for algorithm ISN1OLF

4.7 ISN2OLF: Φ^{2inv} Approximation

Contribution 12: ISN2OLF algorithm

In this section we will create an algorithm that allows a one pass double precision accuracy calculation of $\Phi^{2inv}(p)$ over the full range of double precision floating point numbers. Figure 4.11 shows $\Phi^{2inv}(p)$ in the range [0,2].

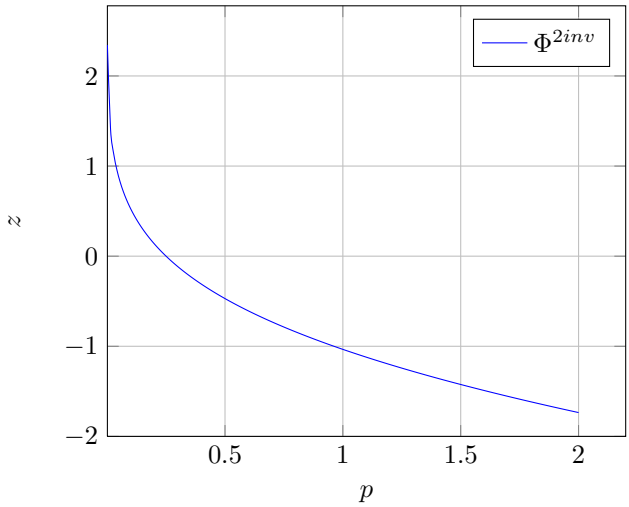

Figure 4.11: Φ^{2inv} over range [0, 2]

(4.36) gives the Φ^2 relation for positive and negative z values. We know that $\Phi^2(7.707353552279367) = 1e-16$, so for larger z values we can further simplify (4.36) into (4.37). We now can find a very simple $\Phi^2(p)$ expression for $p > 30.2016493899167$, see (4.38).

$$\Phi^2(-z) = -\Phi^2(z) + (\frac{z^2+1}{2}) \tag{4.36}$$

$$\Phi^2(-z) = \frac{z^2+1}{2} = p, when\ z > 7.707353552279367 \tag{4.37}$$

$$\Phi^{2inv}(p) = -\sqrt{2p-1} = z, when\ p > 30.2016493899167 \tag{4.38}$$

4.7.1 ISN2OLF Algorithm

We describe the ISN2OLF algorithm, given a value p, it will compute $z = \Phi^{2inv}(p)$ with double precision in the range [2.2250738585072014e-308,

1.7976931348623157e+308]. The acronym ISN2OLF stands for 'inverse standard normal second order loss function'. The valid range is divided in 7 sub ranges, 3 left of the root and 4 to the right of the root. For p values close to zero we first perform a conversion based upon the square root and the natural logarithm of p.

ISN2OLF Algorithm

The rational functions and its parameters are given in Appendix A.6.

1. If $p > 30.2016493899167$ set $z = -\sqrt{2p-1}$, go to step 9
2. If $p \geq 10$: with $x = (30.25 - p)$, set $z =$ (A.17), go to step 9
3. If $p \geq 2.25$: with $x = (10 - p)$, set $z =$ (A.18), go to step 9
4. If $p \geq 0.25$: with $x = (8p - 2)$, set $z =$ (A.19), go to step 9
5. If $p \geq 0.125$: with $x = (8p - 1)$, set $z =$ (A.20), go to step 9
6. Set $y = \sqrt{\ln(-p)}$
7. If $y \leq 4.4375$: with $x = (y - 1.4375)$, set $z =$ (A.21), go to step 9
8. Else: with $x = (y - 4.4375)$, set $z =$ (A.22), go to step 9
9. $\Phi^{1inv}(p) = z$

4.7.2 ISN2OLF Algorithm validation

We used four validation sets with respectively 1 000, 10 000, 100 000 and 1 million random p points uniformly distributed in the range $[1e-100, 30.25]$. These Φ^{2inv} validation values have a relative error (4.39) of less than 1e-32 in the used test sets. For each of the sets we give the maximum magnitude (4.40) of the relative error of algorithm ISN2OLF and the RMS (4.41) of the relative error, see Table 4.10. The definition of ϵ_{rel} for ISN2OLF is given by (4.39).

$$\epsilon_{rel} = \frac{\Phi^{2inv}(p) - ISN2OLF(p)}{|\Phi^{2inv}(p)|} \tag{4.39}$$

$$Max(|\epsilon_{rel}|) = Max(|\epsilon_{rel1}|, |\epsilon_{rel2}|, ..., |\epsilon_{reln}|) \tag{4.40}$$

$$RMS(\epsilon_{rel}) = \sqrt{\frac{1}{n}(\epsilon_{rel1}^2 + \epsilon_{rel2}^2 + ... + \epsilon_{reln}^2)} \tag{4.41}$$

Range	Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
$[1e-100, 30.25]$	1 000	5.82e-16	8.69e-17
$[1e-100, 30.25]$	10 000	6.06e-16	8.83e-17
$[1e-100, 30.25]$	100 000	7.22e-16	8.74e-17
$[1e-100, 30.25]$	1 000 000	9.02e-16	8.74e-17

Table 4.10: Magnitude relative error ISN2OLF

In Figure 4.12 we plot the relative error ϵ_{rel} of the ISN2OLF algorithm, defined by (4.25).

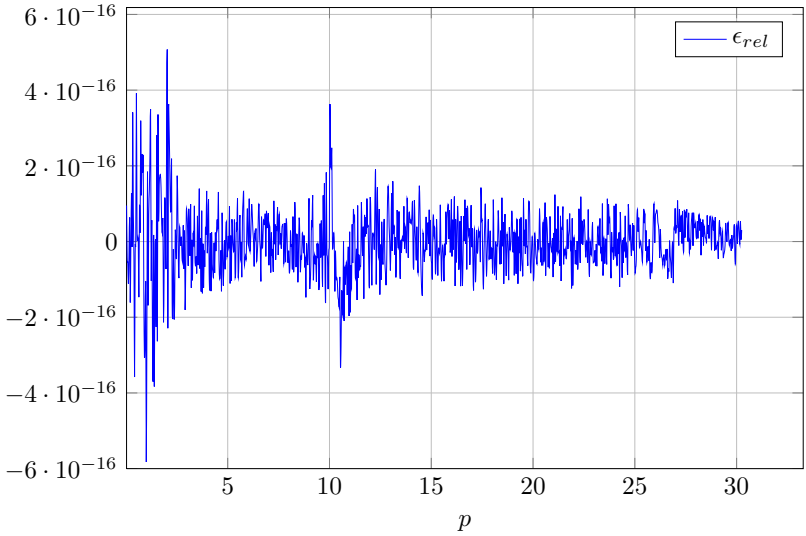

Figure 4.12: ISN2OLF relative error (1 000 test points)

In Table 4.11 we test the values close to $p = 0$. Again we used three validation sets with respectively 10 000, 100 000 and 1 million random p points and Φ^{2inv} values that have an error of less than 1e-32, but now the random p values are uniformly distributed on a logarithmic scale within the considered range.

Range	Test points	$Max(\epsilon_{rel})$	$RMS(\epsilon_{rel})$
$[2e-308, 1]$	1 000	7.20e-16	2.03e-16
$[2e-308, 1]$	10 000	7.80e-16	2.06e-16
$[2e-308, 1]$	100 000	8.83e-16	2.03e-16
$[2e-308, 1]$	1 000 000	8.98e-16	1.95e-16

Table 4.11: Magnitude relative error ISN2OLF: p close to 0 on a logarithmic scale

In Figure 4.13 we plot the relative error ϵ_{rel} of the ISN2OLF algorithm on a logarithmic scale, defined by (4.25). This allows us to check the precision for p values close to zero where Φ^{2inv} increases rapidly.

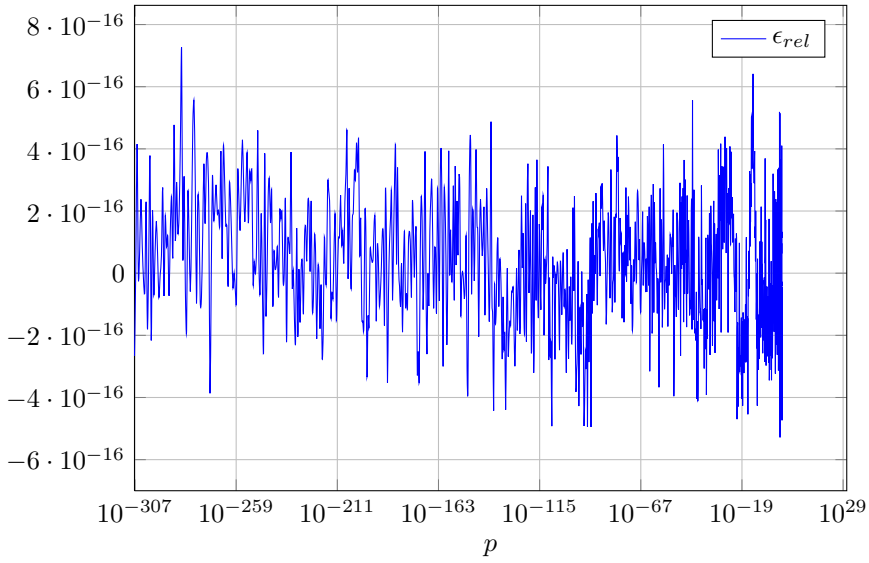

Figure 4.13: ISN2OLF relative error (1 000 test points on logarithmic scale)

4.7.3 ISN2OLF test data for algorithm

Table 4.12 gives values that may be used to check whether the algorithm has been correctly implemented.

p	$z = \text{ISN2OLF}$
$1e-100$	20.985451858273187
0.125	0.40523380736278680
1	−1.03431406136896097

Table 4.12: Test data for algorithm ISN2OLF

4.8 New contributions

In this chapter we have made new contributions on the following topic:

- Closed form approximations for standard normal loss functions

We created the SN1OLF, SN2OLF, ISN1OLF and ISN2OLF algorithms. These are double precision rational approximations for respectively

- SN1OLF: Φ^1 approximation, see contribution 9
- SN2OLF: Φ^2 approximation, see contribution 10
- ISN1OLF: Φ^{1inv} approximation, see contribution 11
- ISN2OLF: Φ^{2inv} approximation, see contribution 12

4.9 Conclusion

No closed-form expressions existed for the standard normal loss functions and their inverse to allow rapid and accurate computations. With these new algorithms it is now possible to have a double precision accurate computation of these functions and their inverse in the full range of double precision floating point numbers. These algorithms will enable a direct reduction of the necessary calculation time when solving inventory problems based on the normal distribution.

5

Multi-item inventory model and solution algorithms

Finally we come to the core goal of this dissertation and formulate the multi-item inventory model with one or more aggregate constraints. We use the previous sections to make a concise formulation. We work out three methods to solve a multi-item inventory model:

1. MIIAC algorithm: a general method to solve a multi-item problem with aggregate and individual constraint(s)
2. MIISSC heuristic: to solve a specific multi-item problem with one aggregate and for each item one individual service constraint
3. MIINLP approach: the use of a non-linear programming engine to solve real life multi-item problems

5.1 Multi-item inventory problem

Now we have defined the necessary demand distributions, the exact model formulations for the different inventory policies and the error reduction functions, we can easily make the step to a multi-item inventory model with one or more multiple aggregate and individual constraints. We will focus on a normal distribution, where we can use additional error reduction functions to close the gap with the exact Poisson or compound Poisson formulations. We choose this approach as it allows us to use the easier normal distribution, but we can avoid significant errors using the error reduction functions, as desired in practice. We retake the general definition of our problem, (5.1), see also the literature review in section 2.4. We now focus on an (r, Q) model. This can be reduced to a base-stock policy, setting $Q = 1$. Making use of the error reduction functions we can have better KPI values for (r, Q) but also for (s, S), see section 3.6.

$$Minimize\ f(r,Q) = \sum_{j=1}^{J} f_j(r_j, Q_j),\ j = 1, .., J \tag{5.1a}$$

$$Subject\ to\ g_{an}(r,Q) = \sum_{j=1}^{J} g_{anj}(r_j, Q_j) \leq e_{an},\ n = 1, .., N \tag{5.1b}$$

$$g_{ioj}(r_j, Q_j) \leq e_{io},\ o = 1, .., O \tag{5.1c}$$

There are J different inventory items. Each item has 2 variables r and Q. The decision variable values are real as we consider normal distribution. There is one goal function $f(r, Q)$, one or multiple (N) aggregate constraint functions $g_{an}(r, Q)$ and one or multiple (O) individual item constraint functions $g_{ioj}(r_j, Q_j)$. The functions f_j, g_{a1j}, ..., g_{aNj} and g_{i1j}, ..., g_{iOj} are thus defined on $\mathbb{R}^2$. Each aggregate constraint function g_{an} has an upper limit e_{an} and each individual constraint function g_{io} has an upper limit e_{io}. Upper and lower bounds on r and Q can be included in g_{io}. As we assume a normal demand, a continuous function, we have $r \in \mathbb{R}$ and $Q \in \mathbb{R}$. For these goal and constraint functions we will always need one or multiple KPI's as defined for the single item inventory models. S_1 is not included in the list of KPI's here as S_1 has no practical value for the customer, as it only represents the % of order cycles without stockouts.

- $\overline{I}$: average inventory
- $\overline{B}$: average backorders
- $\overline{A}$: stockout frequency
- $\overline{P}$: average new backorders

- $\overline{OF}$: average order frequency
- C_v: variable cost
- S_2: fill rate service level
- S_3: ready rate service level
- S_{OL}: order line service level

We will base us here on the normal demand (r, Q) error reduction functions. For the multi-item model optimization we will make use of δ_1, see section 3.6.3, so $p_a = 0$ and (3.48)-(3.51) can be reduced to (5.2)-(5.5).

$$S_{cP} \approx \widetilde{S}_{cP} = p_{(Sb)} S_{nl} + p_{(Sf)} \tag{5.2}$$

$$\overline{I}_{cP} \approx \widetilde{I}_{cP} = p_{(\overline{I}d)} \overline{I}_{nl} + p_{(\overline{I}e)} \tag{5.3}$$

$$\overline{B}_{cP} \approx \widetilde{B}_{cP} = p_{(\overline{B}d)} \overline{B}_{nl} + p_{(\overline{B}e)} \tag{5.4}$$

$$\overline{OF}_{cP} \approx \widetilde{OF}_{cP} = p_{(\overline{OF}b)} / Q_j + p_{(\overline{OF}c)} \tag{5.5}$$

In order to have a clear readable linear formulation we have inserted p_d, p_e and p_f, see (5.6)-(5.8). The p_b and p_c values can be found in Appendix A.2.2 for each of the KPI's and for (r, Q) and (s, S) replenishment policies.

$$p_d = -p_b \tag{5.6}$$

$$p_e = p_b + p_c \tag{5.7}$$

$$p_f = 1 - p_b - p_c \tag{5.8}$$

We now can reformulate approximations (5.2)-(5.5) for the compound Poisson KPI's: (5.9)-(5.16). Notice that we only need to change the parameters p_a-p_f, if we need it for an (r, Q) or an (s, S) policy. The tables with these parameters are available in Appendix A.2.2.

$$\widetilde{B}_{cP} = p_{(\overline{B}d)} \frac{\sigma^2}{Q} \left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) \right] + p_{(\overline{B}e)} \tag{5.9}$$

$$\widetilde{I}_{cP} = p_{(\overline{I}d)} \left[\frac{Q}{2} + r - \nu + 0.5 + \frac{\sigma^2}{Q} \left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) \right] \right] + p_{(\overline{I}e)} \tag{5.10}$$

$$\widetilde{S}_{2cP} = p_{(S2b)} \left[1 - \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right] \right) + p_{(S2f)} \tag{5.11}$$

$$\widetilde{S}_{3cP} = p_{(S3b)} \left[1 - \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right) \right] + p_{(S3f)} \tag{5.12}$$

$$\widetilde{S}_{OLcP} = p_{(SOLb)} \left[1 - \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right] \right] + p_{(SOLf)} \tag{5.13}$$

$$\widetilde{OF}_{cP} = p_{(\overline{OF}b)}/Q + p_{(\overline{OF}c)} \tag{5.14}$$

$$\widetilde{A}_{cP} = 1 - \widetilde{S}_{3cP} \tag{5.15}$$

$$\widetilde{P}_{cP} = \lambda \left(1 - \widetilde{S}_{2cP}\right) \tag{5.16}$$

We only consider items with independent demand subject to at least one aggregate constraint ($N \geq 1$). The inventory cost of these items cannot be optimized independently due to the active aggregate constraint(s) (5.1b).

Using these KPI's we can model a wide variety of multi-item inventory models with one or multiple constraints.

Example 5.1 Classic multi-item inventory problem

We present now a classic example to show how easily this can now be modeled.

A warehouse manager has to minimize his inventory costs, while respecting the overall demand weighted fill rate service level of 95%. He is allowed to vary the fill rate service level per item, but each item also has a minimum fill rate service level of 85%. Additionally it is said that no item can have a negative reorder point, out of practical arrangements. The manager applies an (r, Q) replenishment strategy. There are no shortage costs ($a = p = b = 0$), but within the company a fixed replenishment cost is applied $k = 80$ and the holding cost component is defined as 20% of the unit cost, so $h = 0.2c$. The model becomes:

$$Minimize\ f(x) = \sum_{j=1}^{J} \widetilde{C}_{vcPj}(r_j, Q_j) \tag{5.17a}$$

$$= \sum_{j=1}^{J} [0.2c_j \widetilde{I}_{cPj} + 80\widetilde{OF}_{cPj}] \tag{5.17b}$$

$$Subject\ to\ g_0(x) = \sum_{j=1}^{J} [\lambda_j \widetilde{S}_{2cPj}] \geq 0.95 \tag{5.17c}$$

$$g_j(x) = \widetilde{S}_{2cPj} \geq 0.85,\ j = 1, .., J \tag{5.17d}$$

$$r_j \geq 0,\ j = 1, .., J \tag{5.17e}$$

It is this kind of problems we will solve.

5.2 Lagrange multipliers and convexity

The method of Lagrange multipliers is based on the fact that the gradient vector of the objective function is perpendicular to the constraints surface at an optimal point. This method is suitable for some optimization problems with equality constraints. In case inequality constraints are involved, one needs first to determine which of these inequality constraints are binding and then add them to the equality constraints. These constraints are called the active set of constraints and this set changes during the iterative solution search. Let ζ be the Lagrange variables associated to the aggregate constraints and ξ the Lagrange variables of the individual constraints. The sum of the goal function and the product of the Lagrange multipliers with the active constraints form the 'Lagrange function' $Z(x, \zeta, \xi)$ (5.18a). Setting the partial derivatives of the Lagrangian function equal to zero (5.18b) provides a necessary condition for a solution to the constrained problem (5.1), an extensive and detailed discussion of this approach can be found in Bertsekas (1996) and Bazaraa et al. (2006).

$$\begin{aligned} Z(r, Q, \zeta, \xi) = f(r, Q) &+ \sum_{n=1}^{N} \zeta_n \left[g_{an}(r, Q) - e_{an} \right] \\ &+ \sum_{o=1}^{O} \sum_{j=1}^{J} \xi_{oj} \left[g_{ioj}(r, Q) - e_{io} \right] \end{aligned} \tag{5.18a}$$

$$\nabla Z = 0 \tag{5.18b}$$

Everett (1963) points out the usefulness of Lagrange multipliers for optimization in the presence of constraints. He underlines that it is not limited only to differentiable functions. This method is specifically useful to solve allocation problems with limited resources when faced with independent activities. Patriksson (2008) gives a survey on the continuous nonlinear resource allocation problem. In his paper, a rich list of applications is given, amongst which a few inventory cases. Most of available techniques are based on iteratively finding the Lagrange multiplier(s).

Within each iteration the r and Q values are calculated or approximated, which allows a check on the constraint validation. The challenge in the solution of (5.18b) lies in limiting the number of iterations and reducing the complexity to calculate r and Q in each iteration in order to find the appropriate Lagrange multipliers ζ and ξ.

Note that the problem (5.1) can also be approached using other techniques than Lagrange multipliers, see section 2.4. Some authors apply linear programming and heuristics. In case of integer demand specific enumeration techniques or sometimes mixed integer programming is used, working

fine for smaller models. Continuous approximations can also provide lower bounds.

Rosling (2002a) proved that shortage cost may include nonlinear backorder costs ($\overline{A}$, $\overline{P}$, S_2 and S_3) of dimensions (moneys/quantity unit) and (moneys/time unit). He states that the cost rate that summarizes the expected holding and shortage costs with nonlinear shortage costs remains quasi-convex, although it is not convex, if and only if for all non-negative cost coefficients the cumulative distribution of lead time demand is log-concave. It is proven that in the case of a continuous review model where the cumulative demand has a continuous sample path, e.g. normal distribution, that the necessary conditions are fulfilled and thus the cost rate is quasi-convex. Rosling (2002b) explains that if the cost rate function is quasi-convex, then the cost per period, C_v, is pseudo-convex. The necessary and sufficient condition for a minimum of a pseudo-convex function is to set the partial derivatives equal to zero.

5.3 Multi-item inventory model

5.3.1 Lagrangian function

Our multi-item problem is defined in (5.1). As we know each of the KPI's, (5.9)-(5.16), we can construct a general aggregate goal, aggregate constraint and single constraint function. Next to the 8 KPI's we also included the decision variables r and Q themselves.

Each of the problems we encountered can be constructed using the functions (5.19)-(5.21). In practice only a limited number of the parameters α, β and γ are different from zero.

$$\begin{aligned} f_j(r,Q) = {} & \alpha_{\overline{I}j}\widetilde{I}_{cPj} + \alpha_{\overline{OF}j}\widetilde{OF}_{cPj} + \alpha_{\overline{B}j}\widetilde{B}_{cPj} + \alpha_{\overline{A}j}\widetilde{A}_{cPj} + \alpha_{\overline{P}j}\widetilde{P}_{cPj} \\ & + \alpha_{S2j}\widetilde{S}_{2cPj} + \alpha_{S3j}\widetilde{S}_{3cPj} + \alpha_{SOLj}\widetilde{S}_{OLcPj} + \alpha_{rj}r + \alpha_{Qj}Q \end{aligned} \tag{5.19}$$

$$\begin{aligned} g_{anj}(r,Q) = {} & \beta_{\overline{I}nj}\widetilde{I}_{cPnj} + \beta_{\overline{OF}nj}\widetilde{OF}_{cPnj} + \beta_{\overline{B}nj}\widetilde{B}_{cPnj} \\ & + \beta_{\overline{A}nj}\widetilde{A}_{cPnj} + \beta_{\overline{P}nj}\widetilde{P}_{cPnj} + \beta_{S2nj}\widetilde{S}_{2cPnj} \\ & + \beta_{S3nj}\widetilde{S}_{3cPnj} + \beta_{SOLnj}\widetilde{S}_{OLcPnj} + \beta_{rnj}r + \beta_{Qnj}Q \end{aligned} \tag{5.20}$$

$$\begin{aligned} g_{ioj}(r,Q) = {} & \gamma_{\overline{I}oj}\widetilde{I}_{cPoj} + \gamma_{\overline{OF}oj}\widetilde{OF}_{cPoj} + \gamma_{\overline{B}oj}\widetilde{B}_{cPoj} + \gamma_{\overline{A}oj}\widetilde{A}_{cPoj} + \gamma_{\overline{P}oj}\widetilde{P}_{cPoj} \\ & + \gamma_{S2oj}\widetilde{S}_{2cPoj} + \gamma_{S3oj}\widetilde{S}_{3cPoj} + \gamma_{SOLoj}\widetilde{S}_{OLcPoj} + \gamma_{roj}r + \gamma_{Qoj}Q \end{aligned} \tag{5.21}$$

Combining the problem definition (5.1) and (5.18a) results in the Lagrangian function: (5.22).

$$Z(r,Q,\zeta,\xi) = \sum_{j=1}^{J} f_j(r_j,Q_j) + \sum_{n=1}^{N} \zeta_n \left[\sum_{j=1}^{J} g_{anj}(r_j,Q_j) - e_{an} \right] + \sum_{o=1}^{O} \sum_{j=1}^{J} \xi_{oj} \left[g_{ioj}(r,Q) - e_{io} \right] \tag{5.22a}$$

$$\nabla Z = 0 \tag{5.22b}$$

5.3.2 Conditions for optimum

Finding a solution to our problem (5.1) can now be transformed to solving (5.22), where we only need to consider the binding constraints. Setting the partial derivatives equal to zero yields a set of equations forming the necessary conditions for our solution. We are confronted with the following set of $(2J + N + OJ)$ equations: (5.23)-(5.26).

$$\frac{\partial Z}{\partial r_j} = 0, \quad j = 1,..,J \tag{5.23}$$

$$\frac{\partial Z}{\partial Q_j} = 0, \quad j = 1,..,J \tag{5.24}$$

$$\frac{\partial Z}{\partial \zeta_n} = 0, \quad n = 1,..,N \tag{5.25}$$

$$\frac{\partial Z}{\partial \xi_{oj}} = 0, \quad o = 1,..,O, \quad j = 1,..,J \tag{5.26}$$

To ease the formulations we introduce θ, defined by (5.27), where X can be any of the KPI's. So the Lagrangian function (5.22a) becomes (5.28), see Appendix B.3.3 for intermediate steps.

$$\theta_{Xj} = \alpha_{Xj} + \sum_{n=1}^{N} \beta_{Xnj}\zeta_n + \sum_{o=1}^{O} \gamma_{Xnj}\xi_{oj} \tag{5.27}$$

$$\begin{aligned} Z(r,Q,\zeta,\xi) = \sum_{j=1}^{J} \Big[& \theta_{\overline{I}j}\widetilde{I}_{cPj} + \theta_{\overline{OF}j}\widetilde{OF}_{cPj} + \theta_{\overline{B}j}\widetilde{B}_{cPj} + \theta_{\overline{A}j}\widetilde{A}_{cPj} + \theta_{\overline{P}j}\widetilde{P}_{cPj} \\ & + \theta_{S2j}\widetilde{S}_{2cPj} + \theta_{S3j}\widetilde{S}_{3cPj} + \theta_{SOLj}\widetilde{S}_{OLcPj} + \theta_{rj}r + \theta_{Qj}Q \Big] \\ & - \sum_{n=1}^{N} \zeta_n e_{an} - \sum_{o=1}^{O} \xi_{oj} e_{io} \end{aligned} \tag{5.28}$$

We also introduce the abbreviations: (5.29)-(5.34)

$$\Phi^0_{rj} = \Phi^0(z_{(r+0.5)j}) \tag{5.29}$$

$$\Phi^0_{rQj} = \Phi^0(z_{(r+0.5+Q)j}) \tag{5.30}$$

$$\Phi^1_{rj} = \Phi^1(z_{(r+0.5)j}) \tag{5.31}$$

$$\Phi^1_{rQj} = \Phi^1(z_{(r+0.5+Q)j}) \tag{5.32}$$

$$\Phi^2_{rj} = \Phi^2(z_{(r+0.5)j}) \tag{5.33}$$

$$\Phi^2_{rQj} = \Phi^2(z_{(r+0.5+Q)j}) \tag{5.34}$$

The partial derivatives (5.23) and (5.24) can be converted into respectively (5.35) and (5.36), where we use six new parameters θ^1_j- θ^6_j, see (5.37)-(5.42). The intermediate steps are explained in Appendix B.3.4.

$$\theta^1_j \left[\Phi^1_{rj} - \Phi^1_{rQj}\right] + \theta^2_j \left[\Phi^0_{rj} - \Phi^0_{rQj}\right] + \theta^3_j = 0 \tag{5.35}$$

$$Q_j = \sqrt{\theta^4_j \left[\Phi^2_{rj} - \Phi^2_{rQj} - \frac{Q_j \Phi^1_{rQj}}{\sigma_j}\right] + \theta^5_j \left[\Phi^1_{rj} - \Phi^1_{rQj} - \frac{Q_j \Phi^0_{rQj}}{\sigma_j}\right] + \theta^6_j} \tag{5.36}$$

$$\theta^1_j = -\left(\theta_{\overline{I}j} p_{(\overline{I}d)j} + \theta_{\overline{B}j} p_{(\overline{B}d)j}\right) \sigma_j \tag{5.37}$$

$$\theta^2_j = (\theta_{S2j} - \theta_{\overline{P}j}\lambda_j) p_{(S2b)j} + (\theta_{S3j} - \theta_{\overline{A}j}) p_{(S3b)j} + \theta_{SOLj} p_{(SOLb)j} \tag{5.38}$$

$$\theta^3_j = Q_j \left(\theta_{rj} + \theta_{\overline{I}j} p_{(\overline{I}d)j}\right) \tag{5.39}$$

$$\theta^4_j = \frac{2\sigma^2_j \left(\theta_{\overline{I}j} p_{(\overline{I}d)j} + \theta_{\overline{B}j} p_{(\overline{B}d)j}\right)}{\theta_{\overline{I}j} p_{(\overline{I}d)j} + 2\theta_{Qj}} \tag{5.40}$$

$$\theta^5_j = \frac{-2\sigma_j \left[(\theta_{S2j} - \theta_{\overline{P}j}\lambda_j) p_{(S2c)j} + (\theta_{S3j} - \theta_{\overline{A}j}) p_{(S3c)j} + \theta_{SOLj} p_{(SOLc)j}\right]}{\theta_{\overline{I}j} p_{(\overline{I}d)j} + 2\theta_{Qj}} \tag{5.41}$$

$$\theta^6_j = \frac{2\theta_{\overline{OF}j} p_{(\overline{OF}b)j}}{\theta_{\overline{I}j} p_{(\overline{I}d)j} + 2\theta_{Qj}} \tag{5.42}$$

The partial derivatives (5.25) and (5.26) give each of the N aggregate and OJ individual constraints and these can be translated in respectively (5.43) and (5.44). We must stress that this mathematical representation assumes equality constraints. In reality we will have inequality constraints ($\leq$ or $\geq$). In case of an $\geq$ constraint, the functions (5.43) and (5.44) need to be greater than zero. If this is the case the respective Lagrange variables can be omitted in the Lagrange function, if the functions are negative, a Lagrange variable needs to be integrated in the Lagrange function. In an iterative search this needs to be checked each iteration for each Lagrange variable.

$$\begin{aligned}\sum_{j=1}^{J}\Big[&\beta_{\overline{I}nj}\widetilde{I}_{cPnj}+\beta_{\overline{OF}nj}\widetilde{OF}_{cPnj}+\beta_{\overline{B}nj}\widetilde{B}_{cPnj}+\beta_{\overline{A}nj}\widetilde{A}_{cPnj}\\&+\beta_{\overline{P}nj}\widetilde{P}_{cPnj}+\beta_{S2nj}\widetilde{S}_{2cPnj}+\beta_{S3nj}\widetilde{S}_{3cPnj}\\&+\beta_{SOLnj}\widetilde{S}_{OLcPnj}+\beta_{rnj}r+\beta_{Qnj}Q\Big]-e_{an}=0\end{aligned}\tag{5.43}$$

$$\begin{aligned}&\gamma_{\overline{I}oj}\widetilde{I}_{cPoj}+\gamma_{\overline{OF}oj}\widetilde{OF}_{cPoj}+\gamma_{\overline{B}oj}\widetilde{B}_{cPoj}+\gamma_{\overline{A}oj}\widetilde{A}_{cPoj}\\&+\gamma_{\overline{P}oj}\widetilde{P}_{cPoj}+\gamma_{S2oj}\widetilde{S}_{2cPoj}+\gamma_{S3oj}\widetilde{S}_{3cPoj}\\&+\gamma_{SOLoj}\widetilde{S}_{OLcPoj}+\gamma_{roj}r+\gamma_{Qoj}Q-e_{io}=0\end{aligned}\tag{5.44}$$

For the aggregate constraints (5.43) it is desirable to create an expression for the Lagrange variable. We assume there is only one KPI per aggregate constraint, which is always the case in practice. In (5.45) we create an expression for ζ_n, see Appendix B.3.4.3 for deduction of this equation. We assume first that the aggregate constraint is derived from $\overline{A}$: $\widetilde{A}$, $\widetilde{P}$, $\widetilde{S}_2$, $\widetilde{S}_3$ or $\widetilde{S}_{OL}$. We use in these equations $\widetilde{A}$, but this can be replaced by the previously mentioned KPI's. We find $\theta_j^{2-\zeta}$ and $\theta_j^{2+\zeta}$ in the equation, that are parts of θ_j^2. $\theta_j^{2-\zeta}$ has all the factors except the ζ_n factor, $\theta_j^{2+\zeta}$ has only the ζ_n factor, without ζ_n itself.

$$\zeta_n=\sum_{j=1}^{J}\left[\frac{-(\theta_j^1\left[\Phi_{rj}^1-\Phi_{rQj}^1\right]+\theta_j^{2-\zeta}\left[\Phi_{rj}^0-\Phi_{rQj}^0\right]+\theta_j^3)}{\theta_j^{2+\zeta}\left[\Phi_{rj}^0-\Phi_{rQj}^0\right]}\beta_{\overline{A}nj}\widetilde{A}_{cPnj}\right]/e_{an}\tag{5.45}$$

For an aggregate constraint based on a $\overline{B}$ derivative ($\widetilde{B}$ and $\widetilde{I}$) this becomes (5.46). In this equation we also see $\theta_j^{1-\zeta}$ and $\theta_j^{1+\zeta}$, that are parts of θ_j^1. $\theta_j^{1-\zeta}$ has all the factors except the ζ_n factor, $\theta_j^{1+\zeta}$ has only the ζ_n factor,

without ζ_n itself.

$$\zeta_n = \sum_{j=1}^{J} \left[\frac{-(\theta_j^{1-\zeta} \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right] + \theta_j^2 \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right] + \theta_j^3)}{\theta_j^{1+\zeta} \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right]} \beta_{\overline{B}nj} \widetilde{B}_{cPnj} \right] / e_{an} \tag{5.46}$$

For the individual $\overline{A}$ derived constraints the Lagrange multipliers can be determined by (5.47). See Appendix B.3.4.4 for the deduction of this equation. We find $\theta_j^{2-\xi}$ and $\theta_j^{2+\xi}$ in the equation, that are parts of θ_j^2. $\theta_j^{2-\xi}$ has all the factors except the ξ_{oj} factor, $\theta_j^{2+\xi}$ has only the ξ_{oj} factor, without ξ_{oj} itself.

$$\xi_{oj} = \left[\frac{-(\theta_j^1 \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right] + \theta_j^{2-\xi} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right] + \theta_j^3)}{\theta_j^{2+\xi} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right]} \gamma_{\overline{A}oj} \widetilde{A}_{cPoj} \right] / e_{io} \tag{5.47}$$

For the individual $\overline{B}$ derived constraints the Lagrange multipliers can be determined by (5.48). In this equation we also see $\theta_j^{1-\xi}$ and $\theta_j^{1+\xi}$, that are parts of θ_j^1. $\theta_j^{1-\xi}$ has all the factors except the ξ_{oj} factor, $\theta_j^{1+\xi}$ has only the ξ_{oj} factor, without ξ_{oj} itself.

$$\xi_{oj} = \left[\frac{-(\theta_j^{1-\xi} \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right] + \theta_j^2 \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right] + \theta_j^3)}{\theta_j^{1+\xi} \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right]} \gamma_{\overline{B}oj} \widetilde{B}_{cPoj} \right] / e_{io} \tag{5.48}$$

5.4 Multi-item inventory algorithm

Contribution 13: MIIAC algorithm

MIIAC stands for multi-item inventory problems with aggregate constraint(s). We now will describe a general optimization algorithm for a multi-item inventory model (5.1) with the goal function given by (5.19), the aggregate constraints by (5.20) and with a possibility for individual constraints (5.21). The lower and upper bounds on the decision variables r and Q can be included in the individual constraints. We assume that at least one of the aggregate constraints are binding, if not they can be discarded from the problem.

Our approach is based upon an iterative search method. Within this algorithm we make multiple use of a procedure to calculate r and Q for each item given a set of Lagrange variables $(\zeta_1..\zeta_n, \xi_{11}..\xi_{oj})$, this we call the rQ-Lagrange procedure. When we refer to the complete set of Lagrange multipliers we will notify this as (ζ, ξ) without indexes, when we write an index, we refer to one specific Lagrange multiplier. We refer to this method by $rQ_j^*(\zeta, \xi)$. The accuracy of this model is set by the allowed error ϵ. r^{min}, r^{max}, Q^{min} and Q^{max} are the lower and upper bounds on the decision variables r and Q.

rQ-Lagrange procedure: $rQ_j^*(\zeta, \xi)$

1. Set $Q^0 = EOQ$ with (2.62) and $r^0 = 0$
2. Set $i = 1$
3. Compute r^i with (5.35) using Q^{i-1}
4. If $r^i < r^{min}$, set $r^i = r^{min}$
5. If $r^i > r^{max}$, set $r^i = r^{max}$
6. Compute Q^i with (5.36) using r^i
7. If $Q^i < Q^{min}$, set $Q^i = Q^{min}$
8. If $Q^i > Q^{max}$, set $Q^i = Q^{max}$
9. If $|Q^i - Q^{i-1}| \leq \epsilon$ and $|r^i - r^{i-1}| \leq \epsilon$ go to step (11) else go to step (10)
10. Set $i = i + 1$ and got to step (3)
11. Stop

The MIIAC algorithm is a search procedure for the correct set of Lagrange multipliers (ζ, ξ). Once these are found, we can determine the solution (r, Q) values with the rQ-Lagrange procedure. The MIIAC algorithm first finds an upper and lower bound for each Lagrange multiplier. Using a bi-section method we than successively find the multipliers for each ag-

gregate and individual constraints. Repeating these steps converges to the Lagrange multipliers solving our problem.

MIIAC algorithm

1. Initialize
   (a) Set initial Q^0 (2.62) and r^0 (2.170) values for each item
   (b) For $n = 1..N$ calculate an upper ζ_n^{0+} and lower ζ_n^{0-} value, based on two successive calculations of (5.45) or (5.46)
   (c) For $o = 1..O$ and $j = 1..J$ calculate an upper ξ_{oj}^{0+} and lower ξ_{oj}^{0-} value, based on two successive calculations of (5.47) or (5.48). Exclude upper and lower bounds, these are directly included in the procedure for $rQ_j^*(\zeta, \xi)$.
   (d) Check for feasibility: set all variables to lower bounds and check aggregate constraints, do the same for upper bounds, if infeasible go to step 5
2. Find aggregate constraint Lagrange multiplier(s)
   (a) Set $n = 1$
   (b) If $g_{an}(rQ_j^*(\zeta, \xi)) \leq e_{an}$ then set $\zeta_n = 0$ and go to step (2.h) else go to step (2.c)
   (c) Set $\zeta_n = \zeta_n^{0+}$, if $g_{an}(rQ_j^*(\zeta, \xi)) \leq e_{an}$ then set $\zeta_n^{0+} = \zeta_n^{0+} * 2$ and repeat this step, else set $\zeta_n^{+} = \zeta_n^{0+}$
   (d) Set $\zeta_n = \zeta_n^{0-}$, if $g_{an}(rQ_j^*(\zeta, \xi)) \geq e_{an}$ then set $\zeta_n^{0-} = \zeta_n^{0-}/2$ and repeat this step, else set $\zeta_n^{-} = \zeta_n^{0+}$
   (e) Set $\zeta_n^* = (\zeta_n^{+} + \zeta_n^{-})/2$ and set $\zeta_n = \zeta_n^*$
   (f) If $g_{an}(rQ_j^*(\zeta, \xi)) \leq e_{an}$ then set $\zeta_n^{-} = \zeta_n^*$ else set set $\zeta_n^{+} = \zeta_n^*$
   (g) If $|g_{an}(rQ_j^*(\zeta, \xi)) - e_{an}| \geq \epsilon$ then go back to step (2.e) else go to step (2.h)
   (h) If $n < N$ then set $n = n + 1$ and go back to step (2.b) else go to step (3)
3. Find individual constraints Lagrange multipliers
   (a) Set $o = 1$ and set $j = 1$
   (b) If $g_{ioj}(rQ_j^*(\zeta, \xi)) \leq e_{io}$ then set $\xi_{oj} = 0$ and go to step (3.h) else go to step (3.c)
   (c) Set $\xi_{oj} = \xi_{oj}^{0+}$, if $g_{io}(rQ_j^*(\zeta, \xi)) \leq e_{io}$ then set $\xi_n^{0+} = \xi_n^{0+} * 2$ and repeat this step, else set $\xi_{io}^{+} = \xi_{io}^{0+}$
   (d) Set $\xi_{oj} = \xi_{oj}^{0-}$, if $g_{io}(rQ_j^*(\zeta, \xi)) \geq e_{io}$ then set $\xi_n^{0-} = \xi_n^{0-}/2$ and repeat this step, else set $\xi_{io}^{-} = \xi_{io}^{0-}$
   (e) Set $\xi_{io}^* = (\xi_{io}^{+} + \xi_{io}^{-})/2$ and set $\xi_{io} = \xi_{io}^*$

(f) If $g_{io}(rQ_j^*(\zeta,\xi)) \leq e_{io}$ then set $\xi_n^- = \xi_n^*$ else set set $\xi_n^+ = \xi_n^*$
(g) If $|g_{io}(rQ_j^*(\zeta,\xi)) - e_{io}| \geq \epsilon$ then go back to step (3.e) else go to step (3.h)
(h) If $j < J$ then set $j = j + 1$ and go to step (3.b) else go to step (3.h)
(i) If $o < O$ then set $o = o + 1$ and go to step (3.b) else go to step (4)

4. Check solution quality
   (a) Set $n = 1$
   (b) If $|g_{an}(rQ_j^*(\zeta,\xi)) - e_{an}| \geq \epsilon$ then go to step (4.d) else go to step (4.c)
   (c) If $n < N$ then set $n = n + 1$ and go back to step (4.b) else go to step (5)
   (d) Go back to step (2)
5. Stop

In the MIIAC algorithm we assume that the KPI increases as the Lagrange multiplier increases. If the KPI decreases as the Lagrange multiplier increases, then bi-section method in the steps (2.c), (2.d) and (2.f) for the aggregate constraint(s) and (3.c), (3.d) and (3.f) for the individual constraints need to be reversed.

5.5 Multi-item inventory heuristic

We now want to create an approximating heuristic for a specific case encountered very much in practice:

- Cost: only ordering costs k and holding costs h
- Aggregate constraint: Service level
- Individual constraint: minimum service level per item

5.5.1 Multi-item aggregate fill rate constraint analysis

The MIIAC algorithm can deal with a much wider range of scenarios then the one we focus on here, but we focus on this specific set of goal function, aggregate and individual constraint(s) due to its dominant presence in practice. We start from the data set and assumptions used in section 3.6. In our formulations $C_v(S_{2nl} = 99.9\%)$ is replaced with $C_v(99.9\%)$. In the next figures we plot the following cost functions $(C_v, C_v^a, C_v^b, C_v^c, C_v^d)$ in respect to the fill rate in case of normal demand S_{2nl}. We used a large data set, but plot here only ten items to keep a good oversight of the dynamics. We start from the marginal cost per unit demanded given a certain fill rate S_2: $C_v^a(S_{2nl})$. The expressions C_v^b-C_v^d serve to find a general usable approximation that can easily and generally be approximated with a simple function. In step 1 till 4 we rescale and reformulate the costs versus the fill rate. This brings us to step 4 or Figure 5.4, where each of the lines are almost on top of each other. In step 5, Figure 5.5, we create a very simple algebraic expression that is an approximation of the curves from step 4. So basically we approximate the curves in Figure 5.4 with one simple expression, seen in Figure 5.5. Here we list the different steps:

1. $C_v(S_{2nl})$ variable cost (2.142), Figure 5.1
2. $C_v^a(S_{2nl})$ marginal variable cost (5.49) per unit of demand, Figure 5.2
3. $C_v^b(S_{2nl})$ reset marginal variable cost (5.50) per unit of demand, Figure 5.3
4. $C_v^c(S_{2nl})$ reset and normalized marginal variable cost ratio (5.51) per unit of demand, Figure 5.4. So here we normalize each of the curves so they all can be expressed in a % going from 0% till 100%
5. $C_v^d(S_{2nl})$ reset and normalized marginal variable cost ratio approximation (5.52) per unit of demand, Figure 5.5

The Figures 5.1 - 5.5 visually show how the C_v^d function helps to approximate the variable cost C_v for each S_{2nl}.

$$\begin{aligned} C_v^a(x_2) &\approx \frac{\partial C_v(x_2)}{\partial S_{2nl}} \frac{1}{\lambda} \\ &= \lim_{x_1 \to 0} \left[\frac{C_v(x_2) - C_v(x_2 - x_1)}{x_1} \right] \frac{1}{\lambda} \end{aligned} \tag{5.49}$$

$$C_v^b(S_{2nl}) = C_v^a(S_{2nl}) - C_v^a(S_{2nl} = 80\%) \tag{5.50}$$

$$C_v^c(S_{2nl}) = \frac{C_v^b(S_{2nl})}{C_v^b(S_{2nl} = 99.9\%)} \tag{5.51}$$

$$C_v^c(S_{2nl}) \approx C_v^d = 1/[100\,(1 - (S_{2nl} - 0.8)/0.199)] \tag{5.52}$$

Figure 5.4 clearly shows a quite uniform relation over the different items between S_{2nl} and C_v^c. Combining (5.50)-(5.52), and using some algebra gives (5.53). In the same way we can use it to express S_{2nl} in function of the cost, see (5.54).

$$C_v^a(S_{2nl}) \approx C_v^e(S_{2nl}) = C_v^a(60\%) + \frac{C_v^a(99.9\%) - C_v^a(80\%)}{100\,(1 - (S_{2nl} - 0.8)/0.199)} \tag{5.53}$$

$$S_{2nl} \approx S_{2nl}^e = 0.8 + 0.199 \left[1 - \frac{C_v^a(99.9\%) - C_v^a(80\%)}{100\,[C_v^e(S_{2nl}) - C_v^a(80\%)]} \right] \tag{5.54}$$

We can now approximate the marginal costs C_v^a in (5.54) by using the variable costs C_v, (5.55).

$$\begin{aligned} S_{2nl} &\approx S_{2nl}^f \\ &= 0.8 + 0.199 \left[1 - \frac{[C_v(99.9\%) - C_v(99.5\%)]/0.4\% - C_v(80\%)/80\%}{100\,[C_v^e(S_{2nl}) - C_v(80\%)/80\%]} \right] \end{aligned} \tag{5.55}$$

Rescaling the x-axis (S_{2nl}) allows the use of the error reductions functions. $S_{cP}(S_{2nl} = 80\%)$ is noted as $S_{cP}(80\%)$, it represents the approximate service level for a compound Poisson distribution when $S_{2nl} = 80\%$, making use of the error corrections functions, see (3.48). We can now reformulate (5.55) into (5.56).

$$\begin{aligned} S_{cP} \approx S_{cP}(0.8) + [S_{cP}(0.999) - S_{cP}(0.8)] \Bigg[1 \\ - \frac{[C_v(99.9\%) - C_v(99.5\%)]/[S_{cP}(0.999) - S_{cP}(0.8)] - C_v(80\%)/S_{cP}(0.8)}{100\,[C_v^e(S_{2nl}) - C_v(80\%)/S_{cP}(0.8)]} \Bigg] \end{aligned} \tag{5.56}$$

The approximation function C_v^d is at the heart of the heuristic that we will work out in the next section. It simplifies the relationship between the

variable cost and the fill rate: (5.56). We only need to compute three points at the fill rates: 80%, 99.5% and 99.9%. All intermediate points can be computed with (5.53).

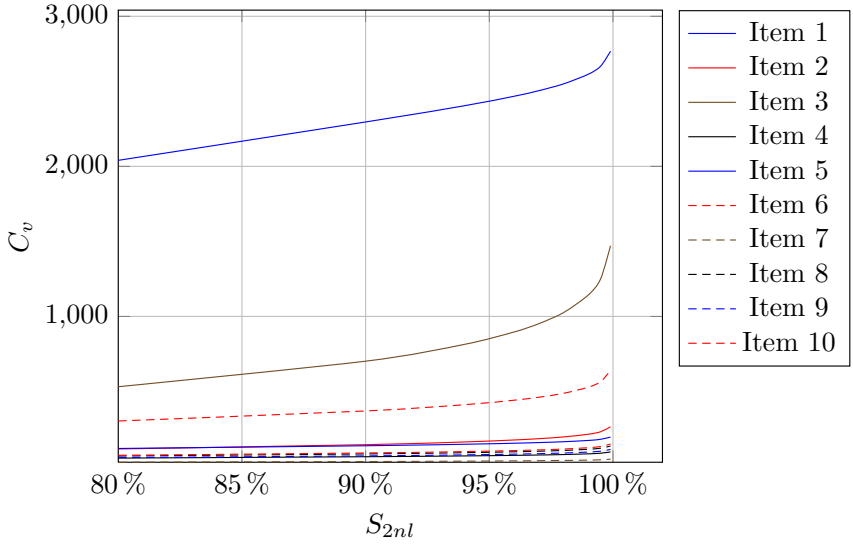

Figure 5.1: Variable cost versus fill rate: $C_v(S_{2nl})$

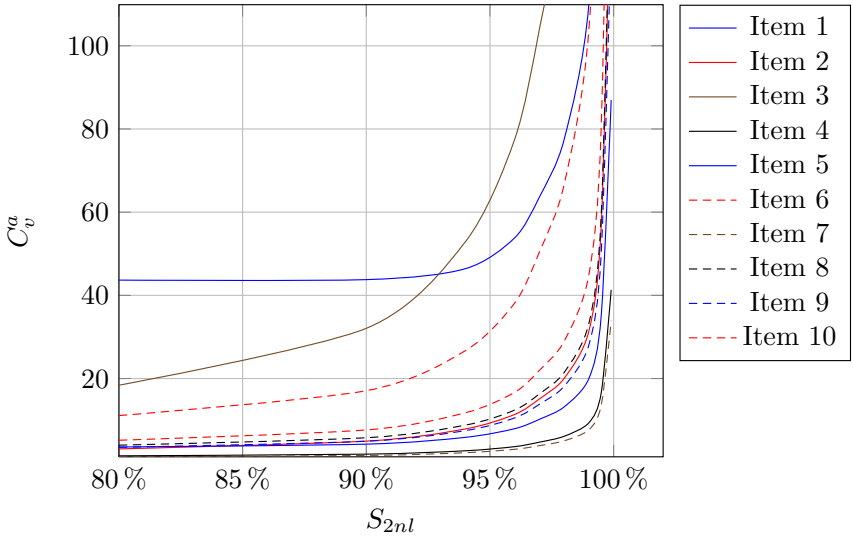

Figure 5.2: Marginal variable cost versus fill rate: $C_v^a(S_{2nl})$

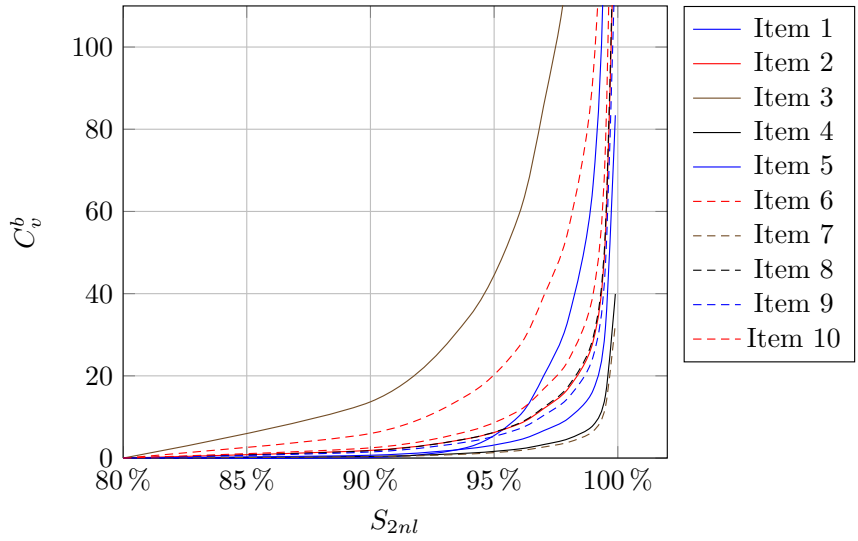

Figure 5.3: Reset marginal variable cost versus fill rate: $C_v^b(S_{2nl})$

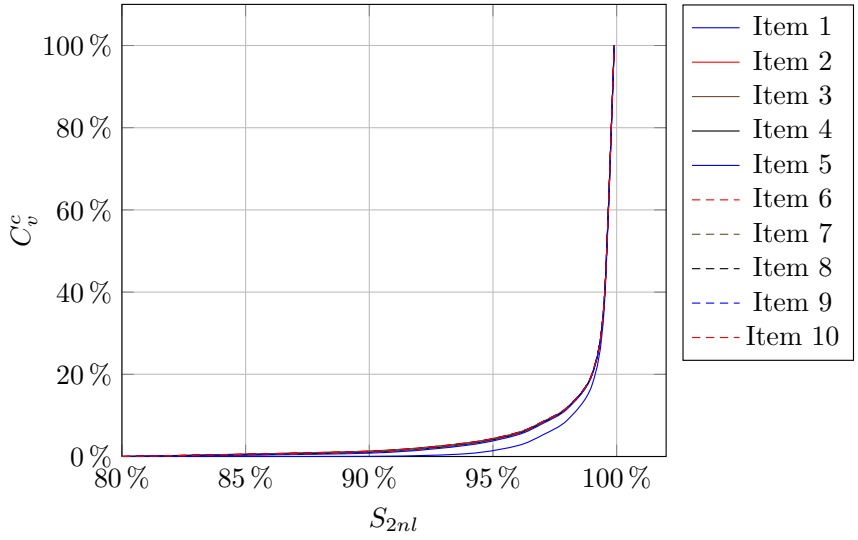

Figure 5.4: Reset and normalized marginal variable cost versus fill rate: $C_v^c(S_{2nl})$

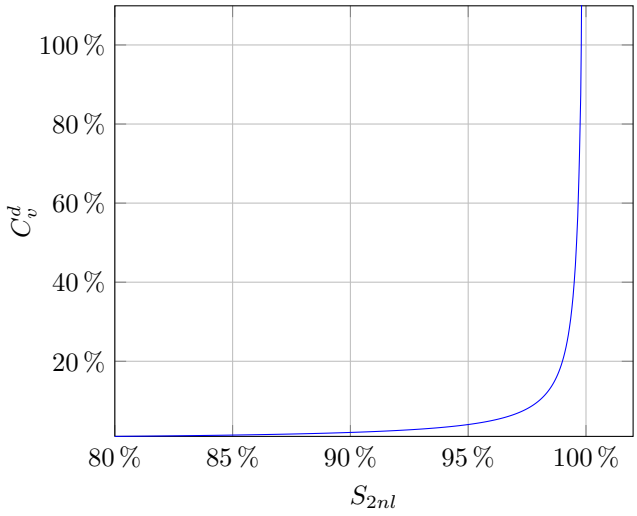

Figure 5.5: Reset and normalized marginal variable cost approximation versus fill rate: $C_v^d(S_{2nl})$

5.5.2 Multi-item inventory heuristic procedure

Contribution 14: MIISSC: Multi-item heuristic

Once we have a mathematical representation of the marginal cost per unit at a given fill rate $C_v^a(S_{2nl})$ (5.53), it opens the doors to apply a marginal analysis, as it was used in a base-stock model, see section 2.4.4. Marginal analysis considers the decrease in backorders by adding one unit to the target stock level, while comparing with the cost of adding one unit for each item. Here we apply the same principle.

You could imagine that we start with each item on the minimum service level, e.g. 80%. Then we could ask how to invest one extra monetary unit. The answer to this question is the item that will have the largest contribution in increasing the system wide fill rate service level that is here defined as the demand weighted service level. Figure 5.2 helps in understanding this principle. We have plotted there 10 items and their marginal costs for one extra item at each service level. If we set $C_v^a = 20$ we can see that for some items we can have a service level of nearly 100%, while two items then have a fill rate of respectively 82% and 92%. For one item we need to allow a marginal cost of 44 in order to reach the individual minimum service level of 80%. Setting each item's service level equal to the service level encountered at the marginal cost line (e.g. 20 or another value), will reach the aim we have in case of an aggregate service level constraint. Cheaper items will have a higher service level than more expensive items. This can now be translated in the MIISSC (Multi-item inventory problem with a system service constraint) heuristic procedure. The goal is to iteratively find the marginal cost C_v^a that will yield the desired aggregate service level, for some items the cost may be higher as we also need to respect a minimum service level per item. The interesting part on this approach is that we can use the error reduction functions created for each of the types of service levels in section 3.6.3. We can easily include the simple (δ_1) or more complex (δ_2) error reduction functions applying equation (3.48) on the x-axis in our examples. We write the heuristic for a general service level definition based on the error reduction functions. $S_{cP}(S_2 = 80\%)$ refers to the service level for compound Poisson demand based on error reduction and the normal demand fill rate service level here equal to 80%. Using these error reduction functions we can also determine the normal demand fill rate that corresponds with a certain compound Poisson service level, we express this as: $S_2(S_{cP} = 80\%)$.

So the goal of MIISSC is: achieve a system wide service level S_{cP}^{goal} and also respect the minimum individual service level S_{cP}^{min}. The service level definition takes into account the error reduction function.

MIISSC heuristic

1. Initialize
   (a) Set initial Q^0 (2.62) and r^0 (2.170) based on target service level for each item
   (b) Calculate $C_{vj}(r^0, Q^0)$ (2.142) for each item
   (c) Set $C_v^{tot} = \sum_{j=1}^{J} C_{vj}$ and $\Lambda = \sum_{j=1}^{J} \lambda$
   (d) Set initial system marginal cost $C_v^e(1) = 0.5C_v^{tot}/\Lambda$
2. Find system marginal cost linked to aggregate fill rate
   (a) Set iteration $i = 1$
   (b) Set $j = 1$
   (c) Set $C_{vj}^{amin} = C_{vj}^a(S_{2nl}(S_{cP}^{min}))$ using (5.53)
   (d) If $C_{vj}^{amin} \geq C_v^e(i)$ then set $S_{cPj} = S_{cP}^{min}$ and go to step (2f) else go to step (2e)
   (e) Set $S_{cPj} = S_{cPj}^e$ using (5.56) and go to step (2f)
   (f) If $j < J$ then set $j = j + 1$ and go to step (2c) else go to step (2g)
   (g) Set $S_{cP}^{sys} = (\sum_{j=1}^{J} \lambda_j S_{cPj})/\Lambda$ and go to step (2h)
   (h) If $|S_{cP}^{sys} - S_{cP}^{goal}| \leq \epsilon$ then go to step (3) else go to step (2i)
   (i) If $S_{cP}^{sys} > S_{cP}^{goal}$ then set $i^+ = i$ and go to step (2l)
   (j) If $S_{cP}^{sys} < S_{cP}^{goal}$ then set $i^- = i$ and go to step (2k)
   (k) If $i^+ = 0$ then set $C_v^e(i+1) = C_v^{ei} * 2$ and go to step (2n)
   (l) If $i^- = 0$ then set $C_v^e(i+1) = C_v^{ei}/2$ and go to step (2n)
   (m) Set $C_v^e(i+1) = [C_v^e(i^+) + C_v^e(i^-)]/2$ and go to step (2n)
   (n) Set $i = i + 1$ and go to step (2b)
3. Compute r_j and Q_j given S_{2j} (See Rosling (2002b)), for $j = 1..J$ and go to step 4
4. Stop

The main advantage of MIISSC compared with MIIAC is that in MIISSC only three service level points need to be calculated for each item for the equations (5.53) and (5.56). All the other points are deducted from the approximation (5.52), only requiring a simple algebra computation and no extra statistical iterative steps. This reduces the number of necessary intermediate computations and reduces calculation time. In the numerical example in section 5.7 we will see that the MIISSC results closely match the MIIAC results.

5.6 General non-linear programming

Contribution 15: MIINLP: non-linear programming

In section 2.4 we have seen multiple examples where general optimization engines are used to solve an approximated model representation. Some use linear programming models (LP): see Niederhoff (2007), Zhou et al. (2008) and Ghalebsaz-Jeddi et al. (2004). Others apply mixed integer programming (MIP): Haksever and Moussourakis (2005). Also a quadratic programming approach was used, see Abdel-Malek and Areeratchakul (2007). We once encountered the use of non-linear solver used: Özler et al. (2009), but then only for a problem with a very limited size (50 items).

We now use a non-linear mathematical programming approach for our problem (5.1). We use the IPOPT (Interior point optimization) engine, a software package for large-scale non-linear optimization. Wachter and Biegler (2006) present IPOPT as a primal-dual interior point algorithm with a filter line-search method for non-linear programming. They analyzed local and global convergence properties and provide a comprehensive description of the IPOPT algorithm. Heuristics are also considered in IPOPT to allow faster performance.

We use as distributed optimization environment 'Optimization services' (OS), see Fourer et al. (2010), in which solvers, modeling languages and analyzers are implemented as services under a unified network. Standards are defined for all necessary activities: representation of optimization instances, results and solver options. It offers the big advantage that it can address different solvers from one point and that it has a uniform modeling language. For our inventory model it allows to use specific non-linear expressions, we are more precisely interested in the normal statistical distributions. The normal cumulative functions were not yet available, but the error function was. We can then use this error function to represent the normal cumulative distribution function, see (2.39), and from there we can use this function to represent the standard normal first and second order loss function, see (2.41) and (2.42). OS integrates the Vedder (1987) implementation of the error function.

We were able to model and solve our problem for large scale instances. So it is possible to use these general non-linear optimization engines for real-life cases. MIINLP approaches the MIIAC quality, but it is slower, on the other hand MIINLP is more versatile.

5.7 Numerical example

Example 5.2 Multi item problem, single item solution

As an example we will start with a very simple three item example to show the benefits of the several contributions made in this dissertation.

Input	Item1	Item2	Item3	Input	Item1	Item2	Item3
λ	40	20	20	f_1	5%	40%	100%
h	20	250	70	f_2	10%	20%	0%
k	80	80	80	f_3	15%	10%	0%
b	0	0	0	f_4	20%	30%	0%
L	0.1	0.1	0.1	f_5	15%	0%	0%
ν	4	2	4	f_6	10%	0%	0%
σ^2	24.08	6	4	f_7	10%	0%	0%
τ	8.2	8.7	20	f_8	5%	0%	0%
χ	4.9	2.3	1	f_9	5%	0%	0%
a	0	0	0	f_{10}	5%	0%	0%
p	0	0	0				

Table 5.1: Example 5.2: Multi-item example

Consider a situation where we want an aggregate demand weighted order line service level of 94%, where the individual order line service level can be no lower than 80%. We now use five methods to solve this:

1. Individual fill rate (94%) as approximation, normal demand
2. Individual order line (94%), compound Poisson demand
3. MIIAC: Aggregate order line (94%), normal demand
4. MIISSC: Aggregate order line (94%), normal demand
5. MIINLPC: Aggregate order line (94%), normal demand

In practice method 1 is used most often, order line is simply replaced with the fill rate service level. Method 1 and 2 apply an individual approach, setting an identical individual target to each item. Only method 2 uses a discrete distribution. All the other methods are based on a normal distribution, so for each of these found (r, Q) solutions we need to perform a rounding step. This step will induce a deviation on the result. Method 3, 4 and 5 have a system approach, the goal is to reach 94% order line service level, while preventing individual order line service levels below 80%. In method 3, 4 and 5 we use the error reduction functions to mimic the order line service level. Each of these methods will yield for each item an r and Q value, see table 5.2. For MIIAC and MIINLP we see the same results, this should be no surprise, both methods have the same model formulation,

MIIAC uses Lagrange multipliers to solve it, while MIINLP applies interior point optimization. MIISSC also has the same model formulation as method 3 and 5, but the heuristic does not guarantee the optimal solution.

	Item1		Item2		Item3	
Method	r	Q	r	Q	r	Q
1: Ind Nl S_2	5	22	3	5	4	11
2: Ind cP S_{OL}	9	31	6	6	4	11
3: MIIAC S_{OL}	12	20	3	5	6	11
4: MIISSC S_{OL}	13	20	3	5	6	11
5: MIINLP S_{OL}	12	20	3	5	6	11

Table 5.2: Example 5.2: Multi-item approach, r and Q values

For each of the found and rounded r and Q values we calculated the exact S_{OL} service level in case of compound Poisson demand, see table in 5.3.

Method	Item1	Item2	Item3	Weighted average	Cost
1: Ind Nl S_2	83.3%	78.5%	92.9%	86.2%	2.459
2: Ind cP S_{OL}	94.5%	94.6%	92.9%	93.9 %	3.369
3: MIIAC S_{OL}	95.6%	78.5%	98.2%	93.2 %	2.727
4: MIISSC S_{OL}	96.5%	78.5%	98.2%	93.6 %	2.747
5: MIINLP S_{OL}	95.6%	78.5%	98.2%	93.2 %	2.727

Table 5.3: Example 5.2: Multi-item example, exact S_{OL} and cost

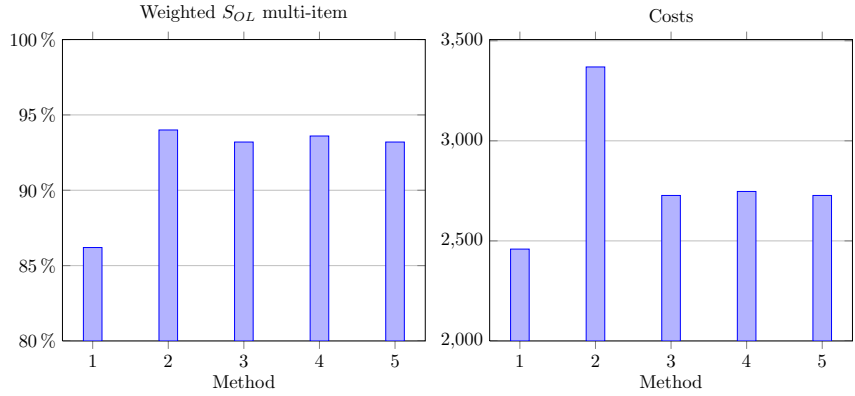

Figure 5.6: Example 5.2: Multi-item Weighted S_{OL} and C_v

Method 1 completely fails and the S_{OL} compound Poisson service level lies 8% below what was desired, and hence the cost is obviously lower. Due

to the discrete distribution aspect we see in method 2 that the exact 94% service level cannot be reached in method 2, for item 1 and 2 it is over 94%, while for item 3 it is lower, the weighted average is 93.9%. For method 3, 4 and 5 we do not respect the individual minimum service level of 80% for item 2, this is because the S_{OL} error reduction function still has a 1.5% error gap. MIIAC, MIISC and MIINLP are less than 1% below the 94% target, and the cost has a reduction of 18%, compared with method 2.

Example 5.3 MIIAC sensitivity analysis

In Figure 5.7 we show the costs of an individual item approach (method 2 out the previous example) and for a system approach (MIIAC method 3 out the previous example). We analyze the cost impact when shifting the target aggregate order line service level from 80% up to 98%. As we have set a minimum individual service level of 80%, it is logic there is no cost difference at 80%. In the theoretical case of 100% or when approaching 100% there will be no cost difference either, as each item will need a service level of 100%.

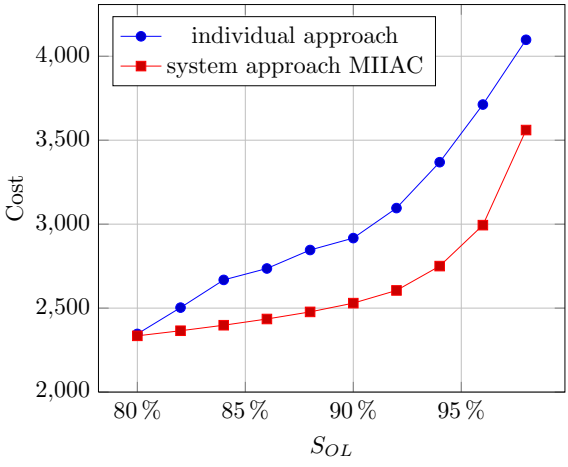

Figure 5.7: Example 5.2: MIIAC cost sensitivity analysis

In Figure 5.8 the relative cost savings for MIIAC compared with an individual approach are plotted. We know that at 80% and near 100% this is 0%. But now we can see that in our simple example the cost difference goes up to nearly 20% at $S_{OL} = 96\%$. From there on the cost savings decrease again very rapidly. But the major interest here is to see that the MIIAC method combined with the error reduction function allows to have reliable service levels and significant cost savings (12%-19%) in the range of service levels used very often in practice: between 90% and 98%.

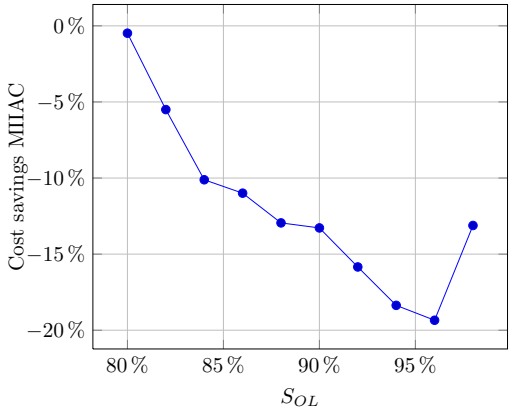

Figure 5.8: Example 5.2: MIIAC cost savings

5.8 Performance

In order to get an indication of the speed we did a small test case with 10, 100, 500 and 1.000 items. The problem had one aggregate service constraint and also individual service constraints. We solved theses cases with MIIAC, MIISSC and MIINLP. The results are given in Table 5.4.

Method	10 items	100 items	500 items	1.000 items
MIISSC	0.05	0.22	1.15	2.19
MIIAC	0.21	1.17	5.34	10.67
MIINLP	0.01	4.10	181.18	1265.53

Table 5.4: Performance multi-item methods: computation time in seconds

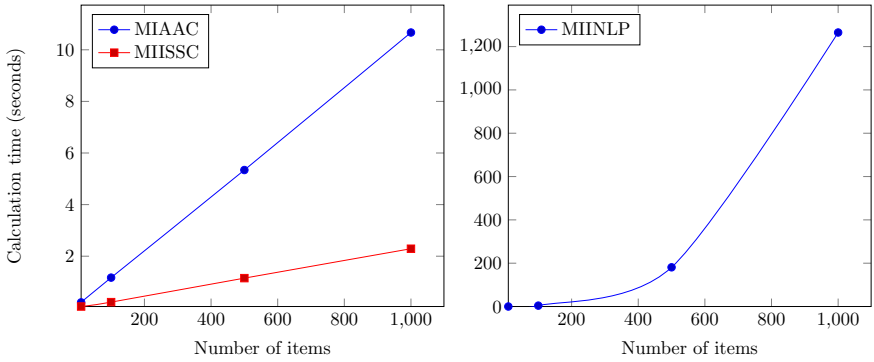

Figure 5.9: Performance multi-item solution methods

Figure 5.9 clearly shows that the calculation time for MIIAC and MIISSC grows linear with the number of items. The calculation time for MIISSC is 20% of the one needed for MIIAC. For the MIINLP the calculation time increases rather polynomial or exponentially. So MIISSC is 600 times faster than MIINLP for an instance of 1 000 items.

5.9 New contributions

We developed three methods for solving multi-item inventory problems with one or multiple aggregate and individual constraints.

The MIIAC algorithm is a general approach for a multi-item inventory problem with aggregate constraints. MIIAC makes use of Lagrange multipliers. MIIAC efficiently and effectively solves real and complex inventory situations in a calculation time that increases linear with the number of items. See contribution 13.

We developed a heuristic for a specific multi-item inventory problem with one aggregate constraint on a system service level and also an individual service constraint per item: MIISSC. Although less versatile than MIIAC, it is 5 times faster. See contribution 14.

We showed that real life large and complex multi-item inventory cases can be solved in non-linear mathematical programming engines (MIINLP). Making use of 'Optimization services' we could integrate the necessary statistical functions. This MIINLP approach has the advantage being even more flexible than MIIAC, the MIINLP computation time is higher than for MIISSC and MIIAC and it also increases much more rapidly with increasing number of items. See contribution 15.

5.10 Conclusions

Multi-item inventory models with aggregate constraint(s) were out of reach of inventory managers, but we made a step at closing the gap. So now they can benefit from the significant cost savings from a system approach on the one hand, but are also capable to integrate aggregate constraints such as investment, workforce, system service level or warehouse space. Depending on the needs we offer three approaches with different flavors of versatility and efficiency. All three can be combined with the error reduction functions, so guaranteeing a good match with Poisson and compound Poisson processes.

6
Real life cases

In this chapter we give two real life cases. The first is a wholesaler environment where we want to maximize the service level but where we are confronted with an investment and warehouse constraint. The MIIAC algorithm is used to solve this. The second case deals with a spare parts warehouse with a wish to reduce cost but maintain the service level. The MIISSC heuristic is used to optimize costs under given constraints. In both cases we can see a significant improvement using a system approach over an individual item approach.

6.1 A wholesaler case study

We will apply the MIIAC algorithm to deal with a multi-product inventory problem with an aggregate constraint. The wholesaler in this case supplies the pharmacies within a predefined region. An order from the pharmacist given in the morning over the internet is delivered in the afternoon. The company has a portfolio of 26.000 stock keeping units (SKU) and an average of 43.500 customer order lines per day with on average only 1,59 pieces per order line. As the sales margin is under pressure, the company felt the need to lower the inventory costs two years ago, but also to improve the fill rate service level, which was lower than the target of 97,5 %. The reasons for the reduced service level were the unpredictable customer demand pattern, the supplier reliability and quality of the products. At the same time the company was confronted with a rapidly increasing number of SKU's and a limited available warehouse space. This meant that in practice ad hoc solutions needed to be found to deal with the shortage of available warehouse space. A high service level is very important in this highly competitive market.

We will compare three scenario's:

1. AS IS: the original r and Q values
2. ABC: r and Q resulting from an ABC-categorized item optimization
3. MIIAC: r and Q values given by the MIIAC algorithm

For each of these scenario's we compute S_2 and $\overline{I}$ based upon (3.5) and (3.7). Scenario 1 uses the r and Q values that existed, for each item the fill rate S_2 and the average investment cost was calculated. This resulted in an average system fill rate of 96.7%.

In scenario 2 the company applied a differentiated item level computation. The order quantity Q was the EOQ (2.62). Based upon this order quantity a reorder point value r was computed with (3.5). An ABC revenue analysis was made to set a higher service level of 99,50% fill rate to the A category, 98% tot the B category items and 96% to the C category items. The A category comprises the fast movers, while the C-category is a set of slow movers. By going towards a fill rate based safety stock calculation that is differentiated over the items using an ABC analysis the investment reduces with 12% while reaching a system fill rate of 97,5%. As the needed space for inventory did not decrease, there was still a warehouse space issue.

In scenario 3, the company was looking to further optimize their inventory, but more importantly they were still confronted with the limited warehouse space. So in scenario 3 a system approach was used. As the wholesaler has an (r, Q) policy and the goal is to maximize the demand weighted fill rate

system service level (6.1a) given an inventory budget and the available warehouse space, the MIIAC algorithm was used. In the aggregate investment constraint (6.1b) we introduced the purchase cost (c_j), as the cost may differ significantly over the items. The warehouse space constraint is (6.1c) where w_j is the space required by item j. This aggregate warehouse space constraint is based upon the average used warehouse space and not the maximum used warehouse space. When focusing on the maximum used warehouse space, the problem is called the 'replenishment staggering problem'. This is much more difficult to solve. This replenishment staggering problem is discussed in section 2.4.1. The staggering problem can also be solved using a Lagrangian approach, but a 'normalizing factor' must be applied in the right hand side of the constraint, so in e_2. We computed e_2 by stating that the average inventory space in scenario 3 must be 10% smaller than the average inventory space in scenario 1, where we knew there was not enough space. This was based upon the warehouse manager's experience that a 10% decrease in needed space will be sufficient.

$$Maximize\ f(x) = \sum_{j=1}^{J} \lambda_j S_{2j} \tag{6.1a}$$

$$Subject\ to\ g_1(x) = \sum_{j=1}^{J} c_j \overline{I}_j \leq e_1 \tag{6.1b}$$

$$g_2(x) = \sum_{j=1}^{J} w_j \overline{I}_j \leq e_2 \tag{6.1c}$$

Scenario	Service level S_2	Investment	Decrease	Space
1: AS IS	96.7%	12.486.953	0%	✗
2: ABC	97.5%	10.986.844	12%	✗
3: MIIAC	97.5%	10.350.000	17%	✓

Table 6.1: Case 1: service levels and inventory costs

Using the MIIAC approach it is possible to create an optimal policy curve that indicates the maximum fill rate system service level for a certain inventory investment where the available warehouse space is not neglected. This system approach resulted in an additional 5% reduction in necessary inventory investment, or a 17% reduction versus the initial situation, while attaining the same system fill rate of 97,5%. It was also shown that a higher system fill rate of 98,5% can be reached if the inventory investment is not to be reduced. But even more important is the fact that these solutions already take into account the available warehouse space.

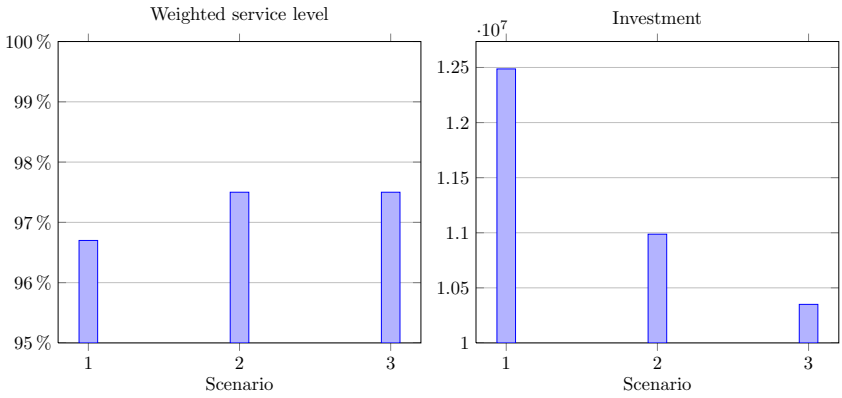

Figure 6.1: Spare parts case: service level and costs: 4 scenarios

In this case study we saw that an item approach combined with an ABC analysis (scenario 2) was able to reduce the inventory investment while attaining the target fill rate, but was unable to incorporate the limited warehouse space at the same time. When applying a MIIAC system approach, in scenario 3, both aggregate constraints, investment and warehouse space, can be satisfied and further cost or service improvement can be realized.

Before we started the creation of the scenarios, the main idea was to create a business case to ask management's approval for a warehouse expansion. This was still the conviction after seeing the results of scenario 2, although it gave the target service level and a better cost, we could also see that the needed warehouse space had not decreased. So there was still a need for costly warehouse expansion. Scenario 3 although provided the unexpected breakthrough. Now it was possible to include a warehouse space constraint, leading to a solution that will fit within the existing walls. The main expectation was that the service level target would no longer be reachable. It is here that the big surprise came. The MIIAC parameters were set so that the individual fill rates could be not lower than 90%. So there were now two big differences with scenario 2:

1. We allowed a bigger variance of fill rate over the items: 90%-99.99% (scenario 3) in stead of 96%-99.5% (scenario 2)
2. Within the range 90%-99.99% every value was possible in scenario 3, while in scenario 2 only 3 values were possible: 96%, 98% and 99.5% and the allocation was made on a rule of thumb

So in the next board meeting the following message could be brought: The awaited warehouse investment can be postponed, while we will be able to increase fill rate up to target!

6.2 Spare parts case

In this case we look at a company that is a diversified supplier of products and services for industry. As quality and reliability is key in the company's strategy, they want to see this also applied in their inventory management. From the site we analyzed they provide the European and African market with spare parts for their products. Headquarters has clear directives in decreasing inventory costs. This was the main reason to analyze and improve their inventory parameter settings.

We consider a portfolio of nearly 40 000 items. More than half of these items can be considered as slow moving items or as lumpy demand, where demand is zero most of the time. About 25% of the items can be seen as a steady demand, while another 15% has a quite large demand but with a large variation. We will consider the following four scenarios:

1. AS IS scenario: these are the r and Q values used before we started this exercise
2. Single item r optimization, only r is considered for change. It was asked to only adapt the safety stock. Q was to be the same as in scenario 1. r was adapted based upon an in-house software package, the equations were unknown.
3. Single item r and Q optimization: using corrected (r, Q) equations, see section 3.3.1
4. MIISSC heuristic: system optimization of r and Q values

We will use the S_2 definition (3.9) to evaluate the service level for each of the four scenarios, without error reduction functions. Currently in scenario 1 the demand weighted service level is 99.38% with a cost of 6.4 million.

The question was to decrease this inventory cost. The used cost parameters are: $h = 0.17c$, $k = 25$. There is a wide range in the purchase cost c, varying from 0.01, for the cheapest item, to over 100.000 for the most expensive item. Management wants to maintain the current service, but feels the need to decrease costs. Cost for each scenario is based upon (3.7) and (3.10).

For each of the scenarios we show the service level and the inventory cost, see table 6.2 and Figure 6.2.

Scenario 2 represents the initial question asked, to optimize solely the safety stocks. The safety stock is the reorder point plus the demand during lead time. So the only parameter we could adapt is the reorder point r. It was believed that the cost was mainly due to the safety stock, and not to the cycle stock, which is the result of the order quantities. We will see in the results that this is clearly a misconception. Recomputing the reorder point

with their software system, while maintaining Q, succeeds in reducing the inventory cost with 23%, the fill rate even increases up to 99.98%.

Scenario	S_2	Cost	Decrease
1: AS IS	99.38%	6.447.016	0%
2: Single item r optimization	99.98%	4.941.287	23%
3: Single item r & Q optimization	99.35%	2.793.448	57%
4: MIISSC	99.35%	1.848.510	71%

Table 6.2: Case 2: service levels and inventory costs

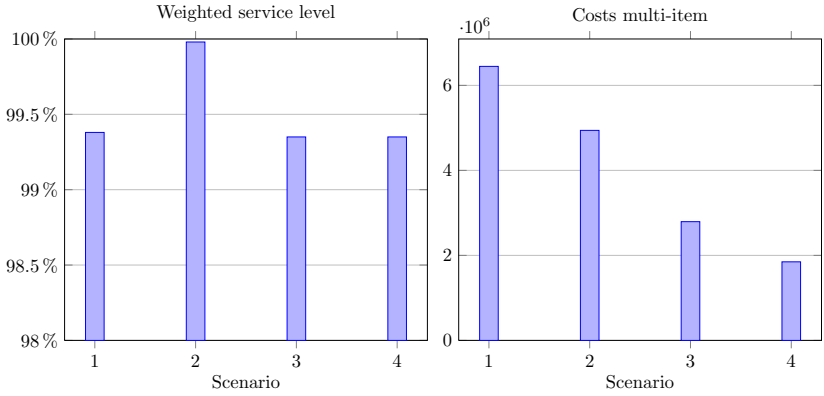

Figure 6.2: Spare parts case: service level and costs: 4 scenarios

Scenario 3 also changes the order quantities, where we simultaneously optimize the r and Q values achieving the identical service level for each item. We convinced the company to create this additional scenario as we knew its potential. This scenario 3 results in a cost reduction of 57% compared with the AS IS scenario. Compared with scenario 2, we see a 43% cost reduction. The achieved service level is 99.35%, this was set as goal, as no service level decrease compared with AS IS was acceptable for management. In this scenario each item has an individual service level constraint of $S_2 = 99.35\%$. So we can conclude here that the current order quantities are definitely not optimized. While initially everybody was strongly convinced that the order quantities could definitely not change, as they were the results of long negotiations over the past years, this was no longer the case when the results of scenario 3 were shown. Renegotiating the order quantities was now seen a business case of 1.2 million, being the difference between the cost in scenario 2 and 3. So some effort could be made to come closer to the optimal order quantities. The company also was given a priority list, as we could indicate the top 100 items where most of the cost difference was situated.

So scenario 2 and 3 tells us that the AS IS situation is not optimized. Taking first steps on single-item level gives a huge improvement. It must be said that this situation is unfortunately not uncommon in practice.

In scenario 4 we even took it one step further and used the MIISSC heuristic. Although it was clear that management would move from scenario 1 to a mix of scenario 2 and scenario 3: for most items the Q value will not change in the near future, only for the top 100 items negotiations will be started. We explained that differentiating the service levels over the items had another untouched potential. In order to estimate the potential benefit we also calculated these costs and service levels. MIISC results in a service level differentiation over the different items, see Figure 6.3. Nearly 50% of the items end up with a service level between 99% and 100%, we also see that 12% of items have a service level of 80%, which we had set as the minimum individual service level. The overall demand weighted service level is again 99.35% as in scenario 3.

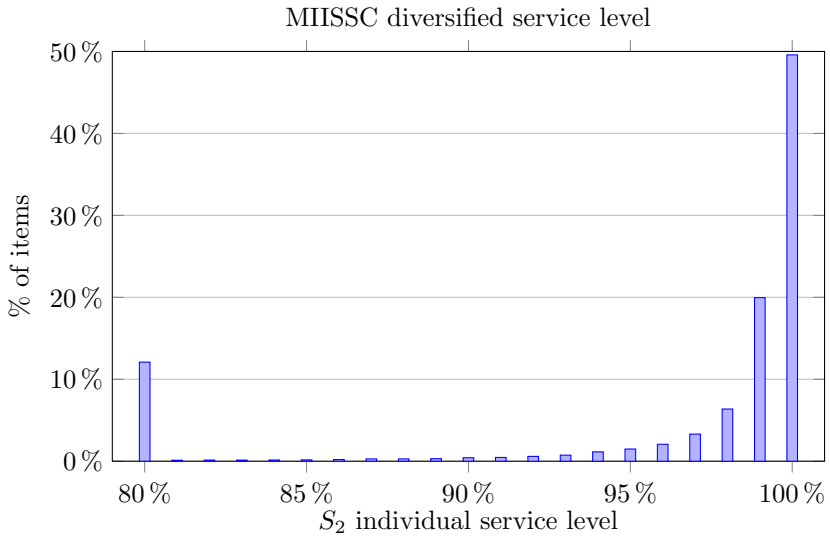

Figure 6.3: Spare parts case: MIISSC diversified service level

Realizing the cost savings from scenario 2, 3 and 4 will ask time, as current stocks need to descend to the new targets in a natural way. For items where this may take too long (up to five years and more), it can be decided to scrap certain stock.

The costs in scenario 4 further decrease with 0.9 million and we end up with a 71% cost decrease compared with the current AS IS situation. Compared with scenario 3 we have a 34% cost reduction, while having the

same overall demand weighted service level of 99.35%. Management was surprised that a further decrease would be possible, but nevertheless they considered this as unreachable for the moment.

So our conclusion here is that the current AS IS situation (scenario 1) has a very high cost for the achieved service level. Optimizing the safety stocks with their in-house software system, while still using the AS IS order quantities, can decrease the cost already significantly, but it leads to a fill rate higher than necessary (scenario 2). With scenario 3 we could explain that a combined optimization of r and Q will realize a cost less than half of the original, so we must state that re-evaluating the order quantities is at order, especially for the 'top 100' products. It was clear for management that renegotiating these order quantities was a necessity. Finally we demonstrated the MIISC heuristic as a system approach in scenario 4, differentiating the service level over the different items between 80% and 100%, which can realize a further cost benefit of 34% compared with scenario 3.

6.3 Conclusion

We have given two real life examples where we used a system approach. In practice we start from the current situation and first perform an individual item analysis and optimization, because this gives already a first result improvement. It is also a necessary step to convince the companies. This intermediate step also helps them to understand the additional value of a system approach. In our first case the system approach gave an additional 5% reduction in necessary inventory investment, knowing that a manual diversification was already applied. But more importantly we were able to integrate the warehouse space constraint in our solution.

In our second case we could realize an additional 34% cost reduction, going from an individual to a system approach. We can conclude that these efficiency gains are really appealing to inventory managers, but it requires the necessary time and intermediate steps to convince them.

7 Conclusions

Within a broad range of situations we see a practical need for an inventory system approach, rather than an item approach. This enables managers to realize their goals with an optimal mix between cost and service while confronted with limited resources such as workspace, workforce or investment.

7.1 Contributions

Throughout this dissertation we made several contributions to solve the multi-item inventory problem with aggregate constraint(s) and/or individual constraints more effectively, more accurately and more efficiently. Here is an overview of these contributions:

1. A comprehensive annotated literature review of multi-item inventory models, see contribution 1.
2. An order line service level: We provided equations for the popular order line service level in case of base-stock, (r, Q) and (s, S) policy, see contributions 2, 3 and 4.
3. Corrected and simplified (r, Q) KPI equations. We worked out a more accurate set of normal demand (r, Q) policy KPI's, especially average inventory and backorders profit greatly, see contribution 5. We developed an easy condition to check whether simplified equations can be used, without the risk of significant accuracy loss, see contribution 6.

4. (r, Q) and (s, S) KPI error analysis and error reduction functions. We provided an analysis of the approximation errors while using a normal demand (r, Q) approximation for a Poisson or compound Poisson demand (r, Q) or (s, S) policy. We established errors up to 60% for order line service level, see contribution 7. Based upon this analysis we created significantly improved approximation functions for the Poisson and compound Poisson exact KPI's based upon a normal distribution. This has the major benefit that we can still apply the simpler normal demand functions while having acceptable and significantly reduced errors. The average error in our realistic data set drops from more than 2% to 0.5% and less. The maximum error drops from 40%-50% to 5%. See contribution 8.
5. Closed form approximations for standard normal loss functions. We created the SN1OLF, SN2OLF, ISN1OLF and ISN2OLF algorithms. These are double precision rational approximations for respectively the standard normal first and second order loss function and their inverse functions. As these functions are in the core of inventory management in case of normal demand, these functions have a direct beneficial impact on the computation time. See contributions 9, 10, 11 and 12.
6. Multi-item aggregate constrained inventory solution methods. We created three methods for solving multi-item inventory problems. The MIIAC algorithm is a general approach for a multi-item inventory problem with aggregate and individual constraints making use of Lagrange multipliers, see contribution 13. MIISSC is a heuristic for a specific multi-item inventory problem with one aggregate and individual constraint on a system service level, see contribution 14. We also showed that real life large and complex multi-item inventory cases can be solved in non-linear mathematical programming engines (MIINLP), see contribution 15.

7.2 Future research

We realize that these results were only possible due to previous breakthroughs. Therefore we are also aware that these results are also only an intermediate step to the next scientific contributions.

The assumptions used for the creation of the error reduction functions, see section 3.6.1, can be changed to cover a broader set of scenarios. As such it can also be established when corrections are not needed, due to very small differences.

The closed form approximations for the standard normal loss functions can be used to create closed form functions directly solving equations (7.1)

and (7.2).

$$\left[\Phi^1(z) - \Phi^1(z + z_c) - c\right] = 0 \tag{7.1}$$

$$a\left[\Phi^1(z) - \Phi^1(z + z_c)\right] + b\left[\Phi^0(z) - \Phi^0(z + z_c)\right] - c = 0 \tag{7.2}$$

Finally we think that additional heuristics can be created for special conditions, in a way as MIISSC is a heuristic for a special condition. We think this has a direct benefit, because in practice these will be welcomed due to higher simplicity in implementation.

References

Abdel-Malek, L., Montanari, R., Meneghetti, D., 2008. The capacitated newsboy problem with random yield: The gardener problem. International Journal of Production Economics 115 (1), 113 – 127.

Abdel-Malek, L. L., Areeratchakul, N., 2007. A quadratic programming approach to the multi-product newsvendor problem with side constraints. European Journal of Operational Research 176 (3), 1607 – 1619.

Abdel-Malek, L. L., Montanari, R., 2005a. An analysis of the multi-product newsboy problem with a budget constraint. International Journal of Production Economics 97 (3), 296 – 307.

Abdel-Malek, L. L., Montanari, R., 2005b. On the multi-product newsboy problem with two constraints. Computers and Operations Research 32 (8), 2095 – 2116.

Axsäter, S., 2003. Supply Chain Operations: Serial and distribution inventory systems. Vol. 11 of Handbooks in Operations Research and Management Science. Elsevier, pp. 561–596.

Axsäter, S., Jun. 2006. Inventory Control (International Series in Operations Research & Management Science), 2nd Edition. Springer, Berlin.

Barrar, R., Loeb, H., 1970. On the remez algorithm for non-linear families. Numerische Mathematik 15, 382–391, 10.1007/BF02165509.

Bazaraa, M. S., Sherali, H. D., Shetty, C. M., 2006. Nonlinear programming, Theory and algorithms, Third edition. John Wiley & Sons, Hoboken, New Jersey.

Bera, U., Rong, M., Mahapatra, N., Maiti, M., 2009. A multi-item mixture inventory model involving random lead time and demand with budget constraint and surprise function. Applied Mathematical Modelling 33 (12), 4337 – 4344.

Bertsekas, D. P., 1996. Constrained Optimization and Lagrange Multiplier Methods. Athena Scientific, Belmont Massachusetts.

Boctor, F. F., 2010. Offsetting inventory replenishment cycles to minimize storage space. European Journal of Operational Research 203 (2), 321 – 325.

Boost, Jun. 2011. Minimax approximations and the remez algorithm. [online].

URL http://www.boost.org/doc/libs/1_44_0/libs/math/doc/sf_and_dist/html/math_toolkit/toolkit/internals2/minimax.html

Boylan, J. E., Johnston, F. R., 1994. Relationships between service level measures for inventory systems. The Journal of the Operational Research Society 45 (7), pp. 838–844.

Bryc, W., 2002. A uniform approximation to the right normal tail integral. Applied Mathematics and Computation 127 (2-3), 365 – 374.

Burden, R., Faires, J., 2005. Numerical analysis. Thomson Brooks/Cole.

Cody, W. J., 1970. A survey of practical rational and polynomial approximation of functions. Society for Industrial and Applied Mathematics 12, 400–423.

Cody, W. J., March 1993. Algorithm 715: Specfuna portable fortran package of special function routines and test drivers. ACM Trans. Math. Softw. 19, 22–30.

Cohen, M. A., Kleindorfer, P. R., Lee, H. L., Pyke, D. F., 1992. Multi-item service constrained (s, s) policies for spare parts logistics systems. Naval Research Logistics (NRL) 39 (4), 561–577.

De Schrijver, Steven, K., Aghezzaf, E.-H., Vanmaele, H., 2011a. Aggregate constrained inventory systems with independent multi-product demand: Control practices and theoretical limitations. International Journal of Production Economics (0), –, dOI: 10.1016/j.ijpe.2011.08.019.

De Schrijver, Steven, K., Aghezzaf, E.-H., Vanmaele, H., 2011b. Double precision rational approximation algorithm for the inverse standard normal first order loss function. Applied Mathematics and Computation.

De Schrijver, Steven, K., Aghezzaf, E.-H., Vanmaele, H., 2011c. Double precision rational approximation algorithms for the standard normal first and second order loss functions. Submitted to Applied Mathematics and Computation.

De Schrijver, Steven, K., Aghezzaf, E.-H., Vanmaele, H., 2012. Towards a systems approach to inventory management: Review of some basic models and analysis of approximation errors. In: Submitted to Proceedings of the 4th International Conference on Information Systems, Logistics and Supply Chain. ILS 2012 - Quebec (Canada), August 26-29.

Desmet, B., Aghezzaf, E., Vanmaele, H., 2009. A normal approximation model for safety stock optimization in a two-echelon distribution system. Journal of the Operational Research Society 61 (1), 156–163.

Desmet, B., Aghezzaf, E., Vanmaele, H., 2010. Safety stock optimisation in two-echelon assembly systems: normal approximation models. International Journal of Production Research 48 (19), 5767–5781.

DeVore, R., Lorentz, G., 1993. Constructive approximation. Grundlehren der mathematischen Wissenschaften. Springer-Verlag.

Dunham, C. B., 1975. Convergence of the fraser-hart algorithm for rational chebyshev approximation. Mathematics of Computation 29, 1078–1082.

Elbarbary, E. M., Elsayed, S. M., Youssef, I. K., Khodier, A. M., 2003. Restrictive chebyshev rational approximation and applications to heat-conduction problems. Applied Mathematics and Computation 136 (2-3), 395 – 403.

Everett, H., 1963. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. Operations Research 11 (3), 399–417.

Federgruen, A., Zheng, Y., 1992. An efficient algorithm for computing an optimal (r, q) policy in continuous review stochastic inventory systems. Operations research, 808–813.

Fourer, R., Ma, J., Martin, K., 2010. Optimization services: A framework for distributed optimization. Operations Research 58 (6), 1624–1636.

Fraser, W., Hart, J., 1962. On the computation of rational approximations to continuous functions. Communications of the ACM 5 (7), 401–403.

Gardner, E., Dannenbring, D., 1979. Using optimal policy surfaces to analyze aggregate inventory tradeoffs. Management Science 25 (8), 709–720.

Ghalebsaz-Jeddi, B., Shultes, B. C., Haji, R., 2004. A multi-product continuous review inventory system with stochastic demand, backorders, and a budget constraint. European Journal of Operational Research 158 (2), 456 – 469.

Goldberg, D., 1991. What every computer scientist should know about floating-point arithmetic. ACM Computing Surveys 23, 5–48.

Goyal, S. K., 1978. A note on "multi-product inventory situations with one restriction". The Journal of the Operational Research Society 29 (3), pp. 269–271.

Hadley, G., Within, T. M., 1963. Analysis of Inventory Systems. Prentice Hall, New Jersey.

Haksever, C., Moussourakis, J., 2005. A model for optimizing multi-product inventory systems with multiple constraints. International Journal of Production Economics 97 (1), 18 – 30.

Harris, F. W., 1913. How many parts to make at once. The Magazine of Management 10 (2), 135–136.

Harris, F. W., 1990. How many parts to make at once. Operations Research 38 (6), pp. 947–950.

Hill, R., Pakkala, T., 2007. Base stock inventory policies for a multi-item demand process. International Journal of Production Economics 109 (1-2), 137 – 148.

Hopp, W. J., Spearman, M. L., Zhang, R. Q., 1997. Easily implementable inventory control policies. Operations Research 45, 327–340.

Johnson, M. E., Lee, H. L., Davis, T., Hall, R., 1995. Expressions for item fill rates in periodic inventory systems. Naval Research Logistics (NRL) 42 (1), 57–80.

Kiani, M., Panareos, J., Psarakis, S., Saleem, M., 2008. Approximations to the normal distribution function and an extended table for the mean range of the normal variables. Journal of te Iranian Statistical Society 7, 57–72.

Kranenburg, A., Houtum, G. V., 2007. Effect of commonality on spare parts provisioning costs for capital goods. International Journal of Production Economics 108 (1-2), 221 – 227.

Lau, H.-S., Lau, A. H.-L., 1996. The newsstand problem: A capacitated multiple-product single-period inventory problem. European Journal of Operational Research 94 (1), 29 – 42.

Lenard, J., Roy, B., 1995. Multi-item inventory control: A multicriteria view. European Journal of Operational Research 87 (3), 685 – 692.

Lesch, S. M., Jeske, D. R., 2009. Some suggestions for teaching about normal approximations to poisson and binomial distribution functions. The American Statistician 63 (3), 274–277.

Linhart, J. M., October 2008. Algorithm 885: Computing the logarithm of the normal distribution. ACM Trans. Math. Softw. 35, 20:1–20:10.

Litinov, G. L., 1993. Approximate construction of rational approximations and the effect of error autocorrection. Russian Journal of Mathematical Physics 1, 313–352.

Mittchel, J. C., 1988. Multi-item inventory systems with a service objective. Operations Research 36, 747–755.

Niederhoff, J. A., 2007. Using separable programming to solve the multi-product multiple ex-ante constraint newsvendor problem and extensions. European Journal of Operational Research 176 (2), 941 – 955.

Özler, A., Tan, B., Karaesmen, F., 2009. Multi-product newsvendor problem with value-at-risk considerations. International Journal of Production Economics 117 (2), 244 – 255.

Page, E., Paul, R. J., 1976. Multi-product inventory situations with one restriction. Operational Research Quarterly (1970-1977) 27 (4), pp. 815–834.

Patriksson, M., 2008. A survey on the continuous nonlinear resource allocation problem. European Journal of Operational Research 185 (1), 1 – 46.

Peizer, D. B., Pratt, J. W., 1968. A normal approximation for binomial, f, beta, and other common, related tail probabilities, i. Journal of the American Statistical Association 63 (324), pp. 1416–1456.

Puerto, J., Fernandez, F., 1998. Pareto-optimality in classical inventory problems. Naval Research Logistics (NRL) 45 (1), 83–98.

R, 2011. The r project for statistical computing.
URL `http://www.r-project.org/`

Ralston, A., Rabinowitz, P., 2001. A first course in numerical analysis. Dover books on mathematics. Dover Publications.

Remez, E. Y., 1934a. Sur la determination des polynomes d'approximation de degre donnee. Comm. Soc. Math. Kharkov 10.

Remez, E. Y., 1934b. Sur le calcul effectiv des polynomes d'approximation des tschebyscheff. Compt. Rend. Acade. Sc. 199 199.

Remez, E. Y., 1934c. Sur un procede convergent d'approximations successives pour determiner les polynomes d'approximation. Compt. Rend. Acad. Sc. 198.

Rosenblatt, M. J., 1981. Multi-item inventory system with budgetary constraint: a comparison between the lagrangian and the fixed cycle approach. International Journal of Production Research 19 (4), 331–339.

Rosenblatt, M. J., Rothblum, U. G., 1990. On the single resource capacity problem for multi-item inventory systems. Operations Research 38 (4), 686–693.

Rosling, K., 2002a. Inventory cost rate functions with nonlinear shortage costs. Operations Research, 1007–1017.

Rosling, K., 2002b. The square-root algorithm for single-item inventory optimization. Tech. rep., Working paper, School of Industrial Engineering, Växjö University, SE-351 95 Växjö, Sweden.

Ross, S., 1996. Stochastic processes, second edition. Wiley series in probability and statistics: Probability and statistics. Wiley.

Ross, S., 2009. Introduction to Probability Models, 10th edition. Elsevier Science.

Schneider, H., Rinks, D. B., 1989. Optimal policy surfaces for a multi-item inventory problem. European Journal of Operational Research 39 (2), 180 – 191.

Schrady, D. A., Choe, U. C., 1971. Models for multi-item continuous review inventory policies subject to constraints. Naval Research Logistics Quarterly 18 (4), 451–463.

Sherbrooke, C. C., 2004. Optimal inventory modeling of systems: multi-echelon techniques, second edition. Kluwer Academic Publishers Group, Dordrecht, The Netherlands.

Shoup, V., Jun. 2011. Ntl: A library for doing number theory. [online]. URL `http://shoup.net/ntl/`

Silver, E., Pyke, D., Peterson, R., 1998. Inventory management and production planning and scheduling. Wiley.

Song, J.-S., Zipkin, P., 2003. Supply chain operations: Assemble-to-order systems. In: Graves, S., de Kok, A. (Eds.), Supply Chain Management: Design, Coordination and Operation. Vol. 11 of Handbooks in Operations Research and Management Science. Elsevier, pp. 561 – 596.

Starr, M., Miller, D., 1962. Inventory Control: Theory and Practice. Prentice-Hall, Englewood Cliffs, NJ.

Thonemann, U. W., Brown, A. O., Hausman, W. H., 2002. Easy quantification of improved spare parts inventory policies. Management Science 48 (9), pp. 1213–1225.

Vairaktarakis, G. L., 2000. Robust multi-item newsboy models with a budget constraint. International Journal of Production Economics 66 (3), 213 – 226.

Vedder, J. D., 1987. Simple approximations for the error function and its inverse. American Journal of Physics 55 (8), 762–763.

Wachter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57.

Waissi, G. R., Rossin, D. F., 1996. A sigmoid approximation of the standard normal integral. Applied Mathematics and Computation 77 (1), 91 – 95.

Waters, C., 2003. Inventory control and management. J. Wiley.

Wichura, M. J., 1988. Algorithm as 241: The percentage points of the normal distribution. Journal of the Royal Statistical Society. Series C (Applied Statistics) 37 (3), pp. 477–484.

Wilson, E., Hilferty, M., 1931. The distribution of chi-square. proceedings of the National Academy of Sciences of the United States of America 17 (12), 684.

Wilson, R. H., 1934. A scientific routine for stock control" harvard business review. Harvard Business Review 13, 116–128.

Withers, C. S., Nadarajah, S., 2010. Repeated integrals of the univariate normal as a finite series with the remainder in terms of moran's functions. Statistics, 1–10.

Zhang, B., Xu, X., Hua, Z., 2009. A binary solution method for the multi-product newsboy problem with budget constraint. International Journal of Production Economics 117 (1), 136 – 141.

Zhao, X., Fan, F., Liu, X., Xie, J., September 2007. Storage-space capacitated inventory system with (r, q) policies. Oper. Res. 55, 854–865.

Zhou, Y., hong Chen, X., run Wang, Z., 2008. Optimal ordering quantities for multi-products with stochastic demand: Return-cvar model. International Journal of Production Economics 112 (2), 782 – 795.

Zipkin, P., August 1986a. Inventory service-level measures: convexity and approximation. Manage. Sci. 32, 975–981.

Zipkin, P., 1986b. Stochastic leadtimes in continuous-time inventory models. Naval Research Logistics Quarterly 33 (4), 763–774.

Zipkin, P. H., 2000. Foundations of Inventory Management. McGraw-Hill, Signapore.

A

Approximation functions and parameters

A.1 Remez algorithm

In section 2.2.1 we briefly introduced the Remez, here we give some more insights in the Remez algorithm. The minimax rational function $R(x)$ is defined to be the function that yields the smallest maximal value of the error function. Chebyshev showed that there is a unique minimax solution for $R(x)$ that has the following properties (Ralston and Rabinowitz (2001)):

- If $R(x)$ is a polynomial of degree N, then there are $N + 2$ unknowns: the $N + 1$ coefficients of the polynomial, and maximal value of the error function.
- The error function has $N + 1$ roots, and $N + 2$ extrema (minima and maxima).
- The extrema alternate in sign, and all have the same magnitude.

If we know the location of the extrema of the error function ϵ then we can write $N + 2$ simultaneous equations where ϵ is the maximal error term, and x_i are the abscissa values of the $N + 2$ extrema of the error function. It is then trivial to solve the simultaneous equations to obtain the polynomial coefficients and the error term (A.1).

$$R(x_i) + (-1)^i \epsilon = f(x_i) \text{ where } i = 1..N+2 \tag{A.1}$$

The Remez algorithms are discussed in DeVore and Lorentz (1993) and Ralston and Rabinowitz (2001). The Remez method is an iterative technique which, given a broad range of assumptions, will converge on the extrema of the error function, and therefore will give the minimax solution. Before we can begin the Remez method, we must obtain an initial value for the location of the extrema of the error function. We could "guess" these, but a much closer first approximation can be obtained by first constructing an interpolated polynomial approximation to $f(x)$. In order to obtain the $N+1$ coefficients of the interpolated polynomial we need $N+1$ points $(x_0...x_N)$, with our interpolated form passing through each of those points that yields N+1 simultaneous equations:

$$f(x_i) = P(x_i) = c_0 + c_1 x_i ... + c_N x_i^N \tag{A.2}$$

Which can be solved for the coefficients $c_0...c_N$ in $P(x)$. Obviously this is not a minimax solution, indeed our only guarantee is that $f(x)$ and $P(x)$ touch at $N+1$ locations, away from those points the error may be arbitrarily large. However, we would clearly like this initial approximation to be as close to $f(x)$ as possible, and it turns out that using the zeros of an orthogonal polynomial as the initial interpolation points is a good choice. We use the zeros of a Chebyshev polynomial as these are particularly easy to calculate, these are the Chebyshev nodes. We prefer these Chebyshev nodes above equidistant nodes, as equidistant nodes produce the Runge effect. This Runge phenomenon is an oscillation at the interval borders in case of high degree polynomial interpolation, see Ralston and Rabinowitz (2001). Next there are two steps that are executed iteratively until the control points are located at the extrema of the error function, and then we have found the minimax solution. These two steps are:

1. Obtain the error term ϵ and the coefficients of $P(x)$

2. Locate the extrema of the new approximation $P(x)$ from the previous step

This method can be extended to a rational approximation (A.3). $P(x)$ and $Q(x)$ are polynomials and then we proceed as before, except that now we have $N+M+2$ unknowns if $P(x)$ is of order N and $Q(x)$ is of order M. This assumes that $Q(x)$ is normalized so that its leading coefficient is 1, giving $N+M+1$ polynomial coefficients in total, plus the error term ϵ. However now the equations become non-linear for the error term ϵ, which

complicates the solution method as we need to solve the set of non-linear equations iteratively until a stable value for ϵ is found.

$$f(x) = R(x) = \frac{P(x)}{Q(x)} \tag{A.3}$$

A.2 Error reduction functions tables

A.2.1 Maximum and average errors

	Scenarios								
KPI	1	2	3	4a	4b	5a	5b	6a	6b
IrQ	50,84%	50,84%	2,56%						
IsS	50,84%	50,84%	19,02%	7%	7%	7%	7%	4,5%	4,5%
BrQ	100%	18,74%	24,71%	10%	10%	6%	6%	4%	4%
BsS	100%	18,69%	24,65%	10%	10%	7%	7%	4%	4%
OFsS			390%	200%	200%	35%	35%		
S3rQ	100%	7,04%	9,2%	6%	6%	4,5%	4,5%	3%	3%
S3sS	100%	11,72%	9,2%	6%	6%	5%	5%	3%	3%
S2rQ	58,2%	12,39%	14,8%	8%	8%	5%	5%	5%	5%
S2sS	58,2%	12,39%	14,8%	8%	8%	7%	7%	7%	6,37%
SOLrQ	44,61%	16,17%	16,8%	8%	8%	6%	6%	6%	6%
SOLsS	44,61%	14,81%	15,9%	8%	8%	6%	6%	5,5%	5,5%

Table A.1: Maximum errors $\epsilon_{max|\overline{I}|}$, $\epsilon_{max|\overline{B}|}$ & $\epsilon_{max|S|}$: Scenarios 1-6

	Scenarios								
KPI	1	2	3	4a	4b	5a	5b	6a	6b
IrQ	3,99%	4,12%	0,33%						
IsS	6,84%	6,77%	3,2%	1,39%	1,25%	1,21%	1,09%	0,85%	0,66%
BrQ	5,32%	3,63%	4,36%	2,33%	1,87%	1,53%	1,16%	1,16%	0,76%
BsS	4,74%	2,84%	3,61%	2,92%	2,22%	1,75%	1,28%	1,14%	0,7%
OFsS			11.93%	5.74%	5.35%	3.45%	1.73%		
S3rQ	2,42%	0,66%	1,08%	0,59%	0,57%	0,51%	0,44%	0,41%	0,3%
S3sS	2,51%	0,7%	0,73%	0,68%	0,59%	0,45%	0,42%	0,4%	0,31%
S2rQ	2,22%	1,8%	2,21%	0,89%	0,83%	0,73%	0,57%	0,56%	0,43%
S2sS	1,92%	1,51%	1,93%	0,89%	0,82%	0,75%	0,56%	0,52%	0,39%
SOLrQ	2,4%	2,2%	2,58%	1,28%	1,15%	1,13%	0,78%	0,74%	0,65%
SOLsS	2,07%	1,9%	2,27%	1,22%	1,08%	0,94%	0,77%	0,73%	0,61%

Table A.2: Average errors $\epsilon_{avg|\overline{I}|}$, $\epsilon_{avg|\overline{B}|}$ & $\epsilon_{avg|S|}$: Scenarios 1-6

A.2.2 Error reduction function parameters

In order to calculate the error reduction functions we use (3.48)-(3.51). We need to calculate p_a, p_b and p_c using

- (3.52)-(3.54) for $\overline{I}$, $\overline{B}$ and the service levels S
- (3.55)-(3.57) for $\overline{OF}$

In the following tables we give the $p_{a1} - p_{a9}$, $p_{b1} - p_{b13}$ and $p_{c1} - p_{c13}$ values for δ_{S2}, δ_{S3}, δ_{SOL}, $\delta_{\overline{I}}$, $\delta_{\overline{B}}$ and $\delta_{\overline{OF}}$ for an (r, Q) and (s, S) policy.

Param	s2sS δ_{1a}	s2sS δ_{1b}	s2sS δ_{2a}	s2sS δ_{2b}	s2sS δ_{3b}	s2sS δ_{3a}
p_{a1}	0	0	0,62079872	0,1599473	1,0798454	1,3967634
p_{a2}	0	0	0,12802276	-0,57096543	0,91040687	1,323997
p_{a3}	0	0	-20,034212	0,72213321	-5,378452	-9,5274545
p_{a4}	0	0	-49,531247	-40,017852	-28,399061	-36,045843
p_{a5}	0	0	-3,1482104	-3,2059023	-2,9431094	-5,1367495
p_{a6}	0	0	36,026886	-12,801874	0,59132196	9,0097137
p_{a7}	0	0	36,703659	55,768683	34,372128	38,424411
p_{a8}	0	0	0	0	-0,14036164	-0,1395666
p_{a9}	0	0	0	0	-0,91854392	-1,1488489
p_{b1}	-0,02611578	-0,074707104	-0,090256163	-0,072040932	-0,15649184	-0,17234494
p_{b2}	-0,39295316	-0,088121154	0	-0,043230975	-0,11692107	-0,13102561
p_{b3}	4,4158436	2,7172644	4,0626909	2,8118799	2,6979054	3,0054568
p_{b4}	0	0,0019196106	0	0,0011752242	0,00082906157	0
p_{b5}	0,019977146	0,0048107489	0	0,0049695396	0,0017346291	0
p_{b6}	2,1137474	1,6672442	5,5701984	4,4659515	3,7824873	5,1203535
p_{b7}	-0,10157257	0,0091226001	0	0,032904793	0,036329303	0
p_{b8}	-7,410916	-2,8350503	-5,7912643	-2,5199876	-2,1401526	-2,9176387
p_{b9}	2,6837517	-0,026093258	-2,4261323	-3,280375	-2,7530877	-3,214286
p_{b10}	0	0	0	0	0,0078792799	0
p_{b11}	0	0	0	0	0,094479321	0,12325378
p_{b12}	0	0	0	0	0,33400085	-0,51776544
p_{b13}	0	0	0	0	-0,19454628	0,50003518
p_{c1}	0	-0,063525496	0	-0,0071437738	-0,00227371	0
p_{c2}	0	0,00077584365	0	0,00026803271	-0,00021250232	0
p_{c3}	0	0,000094147634	0	-0,000055976107	-0,000064628204	0
p_{c4}	0	-0,000035129026	0	0,00000030004384	-0,00000097659247	0
p_{c5}	0	-0,037811223	0	-0,0083671567	-0,0070232835	0
p_{c6}	0	0,027375788	0	0,0062666771	0,0045740367	0
p_{c7}	0	0,0002613652	0	0,010794375	0,0046186448	0
p_{c8}	-0,0029257815	0,038203518	0	-0,0012884843	0,0047034847	0
p_{c9}	0,013107402	0,046439026	0	0,0031299817	0,0036350312	0
p_{c10}	0	0,0010210386	0	0,0024085224	0,0024714804	0
p_{c11}	0	-0,0084033388	0	-0,0027621457	-0,0017252219	0
p_{c12}	0	0	0	0	-0,00046685486	0
p_{c13}	0	0	0	0	0,000023300055	0

Table A.3: Error reduction function parameters S_2 (s, S)

Param	s2rQ δ_{1b}	s2rQ δ_{1a}	s2rQ δ_{2b}	s2rQ δ_{2a}	s2rQ δ_{3b}	s2rQ δ_{3a}
p_{a1}	0	0	0,319345	0,58264678	1,3068164	2,2438518
p_{a2}	0	0	-0,14218018	0,71534621	1,1916032	2,1018105
p_{a3}	0	0	-7,8907152	-27,110399	-5,69654	-22,308692
p_{a4}	0	0	-59,762032	-48,817051	-56,129956	-34,142493
p_{a5}	0	0	-10,355889	-23,213402	-9,2795231	-19,960706
p_{a6}	0	0	5,0696127	58,796136	-3,3440988	25,930044
p_{a7}	0	0	72,099316	40,0512	71,648701	45,059747
p_{a8}	0	0	0	0	-0,21660837	-0,21502194
p_{a9}	0	0	0	0	-1,1380356	-1,6803797
p_{b1}	-0,093056357	-0,025548024	-0,11393864	-0,084645242	-0,18721526	-0,2032524
p_{b2}	0,04867344	0,024706683	-0,12417076	-0,094639771	-0,19038537	-0,1325155
p_{b3}	2,6715921	1,8876832	4,6291679	5,4161753	4,0641117	4,388622
p_{b4}	0,0020777944	0	0,0016722105	0	0,0017622894	0
p_{b5}	-0,0040073297	0	0,0057130999	0	0,0051953383	0
p_{b6}	1,6447414	2,6554164	6,1509592	6,6159906	6,0273045	4,7483159
p_{b7}	-0,011862679	0,013296669	-0,062070529	-0,2361067	-0,018703533	0
p_{b8}	-3,4403478	-2,0493715	-7,1283813	-11,683471	-5,0713104	-6,6014941
p_{b9}	0,55228586	-1,2321784	-1,8629891	1,2794969	-3,0855901	-0,62462837
p_{b10}	0	0	0	0	0,0034387417	0
p_{b11}	0	0	0	0	0,0863032	0,11437209
p_{b12}	0	0	0	0	0,084609035	0,92880496
p_{b13}	0	0	0	0	-0,077634809	-0,71173895
p_{c1}	-0,059868984	-0,022121814	-0,048293748	0	-0,015728501	0,03957095
p_{c2}	0,00098715684	0	0,00064660185	0	-0,000045354969	0
p_{c3}	0,00036582933	0	0,00020343543	0	0,0000020999789	0
p_{c4}	-0,000051599837	0	-0,000017642764	0	-0,0000075524917	0
p_{c5}	-0,04933468	0	-0,024821085	0	-0,015265794	0
p_{c6}	0,029943003	0	0,013592632	0	0,0087347614	0
p_{c7}	0,0020444964	0	-0,00048029309	0	0,0053503178	0
p_{c8}	0,027481707	0,018575703	0,02745706	0	0,012794724	-0,039567334
p_{c9}	0,058392313	0,015879836	0,037858452	0,011040824	0,012528778	0
p_{c10}	0,0006286459	0	0,00022471902	0	0,0029106283	0
p_{c11}	-0,0057044102	0	-0,0023789469	0	-0,0022140348	0
p_{c12}	0	0	0	0	-0,00045734151	0
p_{c13}	0	0	0	0	0,000022824387	0

Table A.4: Error reduction function parameters S_2 (r, Q)

Param	solrQ δ_{1b}	solrQ δ_{1a}	solrQ δ_{2b}	solrQ δ_{2a}	solrQ δ_{3b}	solrQ δ_{3a}
p_{a1}	0	0	0,69304572	1,1386248	1,8695746	2,0096875
p_{a2}	0	0	-0,49655237	0,75921572	1,6232623	1,6950568
p_{a3}	0	0	-16,583902	-37,863546	-18,17976	-20,522426
p_{a4}	0	0	-73,593654	-59,091644	-68,680361	-59,450898
p_{a5}	0	0	-9,6870649	-19,241753	-16,109663	-21,682771
p_{a6}	0	0	28,179534	65,26442	18,402429	18,600763
p_{a7}	0	0	71,451382	52,989003	82,295512	81,575836
p_{a8}	0	0	0	0	-0,1931702	-0,17905432
p_{a9}	0	0	0	0	-1,3068424	-1,3919834
p_{b1}	-0,13502253	-0,051127719	-0,17205385	-0,14427946	-0,25211473	-0,20194119
p_{b2}	0,20292933	0	-0,038207738	-0,049839461	-0,27915822	-0,27171325
p_{b3}	2,7780417	3,2170714	5,5112844	5,9813171	5,7136177	5,6059237
p_{b4}	0,0028229168	0	0,0020503299	0	0,0014088256	0
p_{b5}	-0,011190191	0	0,0048034974	0	0,0091516859	0,010653448
p_{b6}	1,7122237	2,5288408	7,5777913	7,1311986	8,1305118	7,8474796
p_{b7}	-0,028996949	0	-0,10454289	0	-0,063883248	-0,037919489
p_{b8}	-4,1115344	-5,8470184	-9,4049023	-10,338384	-8,5747618	-8,2768653
p_{b9}	0,97157652	1,4931449	-1,982907	-1,4611739	-3,3329951	-3,4425888
p_{b10}	0	0	0	0	0,0021086	-0,0090364055
p_{b11}	0	0	0	0	0,11384496	0,099286812
p_{b12}	0	0	0	0	-0,36691561	-0,47740656
p_{b13}	0	0	0	0	0,30272232	0,41608523
p_{c1}	-0,16087068	-0,10301964	-0,15054173	-0,048512801	-0,11655432	-0,078485268
p_{c2}	0,001925227	0	0,0015424546	0	0,00085523308	0
p_{c3}	0,00067314505	0	0,00045114229	0	0,00019934309	0
p_{c4}	-0,00010486939	0	-0,000066188166	0	-0,000044067262	0
p_{c5}	-0,10350144	-0,023138145	-0,076560567	0	-0,063676544	-0,049894364
p_{c6}	0,062975271	0,011059809	0,046388721	0	0,039180756	0,025364021
p_{c7}	-0,017052662	-0,060077067	-0,017920381	-0,043159059	-0,00041884938	0
p_{c8}	0,097046664	0,098028167	0,099733485	0,048341894	0,082243808	0,067466939
p_{c9}	0,13882216	0,084293184	0,11362056	0,043923377	0,082113792	0,051951869
p_{c10}	-0,00051077774	0	-0,00073500703	0	0,0026553054	0,0073128753
p_{c11}	-0,012250117	0	-0,0097755154	0	-0,010102833	-0,004524725
p_{c12}	0	0	0	0	-0,00065924394	-0,0000010245204
p_{c13}	0	0	0	0	0,000032919509	0

Table A.5: Error reduction function parameters S_{OL} (r, Q)

Param	solsS δ_{1b}	solsS δ_{1a}	solsS δ_{2b}	solsS δ_{2a}	solsS δ_{3a}	solsS δ_{3b}
p_{a1}	0	0	0,467753	0	2,0188102	1,6047157
p_{a2}	0	0	-0,62381432	0	1,7456738	1,4809897
p_{a3}	0	0	-8,3000401	-7,5222445	-17,365721	-9,901217
p_{a4}	0	0	-78,024238	-64,291931	-60,03781	-66,13594
p_{a5}	0	0	-9,118183	-27,321977	-13,624565	-11,808639
p_{a6}	0	0	6,9033246	16,883768	17,886888	2,6149926
p_{a7}	0	0	89,307234	81,263497	68,918499	83,646533
p_{a8}	0	0	0	0	-0,080646435	-0,14984375
p_{a9}	0	0	0	0	-1,5722694	-1,2696389
p_{b1}	-0,12094627	0	-0,16964522	0	-0,1919155	-0,24487291
p_{b2}	-0,034701614	0	-0,20919519	0	-0,048718894	-0,27565277
p_{b3}	3,3806048	1,7308892	5,7764932	2,6644645	3,1068135	4,4942627
p_{b4}	0,002845863	0	0,0029507562	0	0	0,0013560846
p_{b5}	0,0031633904	0	0,014033752	0	0	0,0083151585
p_{b6}	1,6245828	2,927518	7,5568163	8,875388	6,8950591	7,247074
p_{b7}	-0,026997998	0	-0,083969775	-0,070279616	0	-0,014953498
p_{b8}	-4,5945376	-2,1751484	-9,1087691	-6,0176849	-3,7798189	-6,001812
p_{b9}	1,0996987	-1,1645099	-2,4885522	-3,9639563	-4,7702216	-4,1129281
p_{b10}	0	0	0	0	0	0,010487004
p_{b11}	0	0	0	0	0,11263307	0,1319167
p_{b12}	0	0	0	0	1,5565025	0,23278798
p_{b13}	0	0	0	0	-1,075476	-0,085658465
p_{c1}	-0,1311776	-0,1069731	-0,16225023	-0,13621486	-0,11010993	-0,15357573
p_{c2}	0,0020868316	0	0,0020510924	0	0	0,0011368941
p_{c3}	0,00030487165	0	0,0003143049	0	0	0,00021905194
p_{c4}	-0,00010782086	0	-0,00009273536	0	0	-0,000064822988
p_{c5}	-0,12011277	-0,021604686	-0,10871587	-0,027941029	-0,027575727	-0,087837502
p_{c6}	0,082366833	0,0096659676	0,074126475	0,012509152	0,012659412	0,057779277
p_{c7}	0,016860695	-0,0569233	0,0069348589	-0,059197018	-0,050634706	0,0023113938
p_{c8}	0,06213046	0,10150033	0,094714791	0,12913954	0,10354824	0,1072372
p_{c9}	0,11396067	0,083199378	0,11998647	0,093172242	0,082278443	0,11004475
p_{c10}	0,0048782829	0	0,0023721561	0	0	0,0021967392
p_{c11}	-0,023486002	0	-0,021553579	0	0	-0,016816743
p_{c12}	0	0	0	0	0	-0,00092234322
p_{c13}	0	0	0	0	0	0,000046074473

Table A.6: Error reduction function parameters S_{OL} (s, S)

Param	IsS δ_{1b}	IsS δ_{1a}	IsS δ_{2b}	IsS δ_{2a}	IsS δ_{3b}	IsS δ_{3a}
p_{a1}	-0,000000061726575	0	-0,000024597061	0	-0,000022233502	0
p_{a2}	0	0	-0,000035166878	0	-0,00021778966	0
p_{a3}	0	0	-0,0005388069	0	0,066003069	0,050709921
p_{a4}	0	0	0,008050581	0,0081718164	0,002656812	0
p_{a5}	0	0	0,00011652671	0	0,00048410896	0
p_{a6}	0	0	0,0050781063	-0,00019068446	0,010987296	0
p_{a7}	0	0	-0,011486639	-0,0081263923	-0,014667448	0
p_{a8}	0	0	0	0	0,000032648677	0
p_{a9}	0	0	0	0	0,00004010176	0
p_{b1}	-0,0025764719	0	-0,0039686866	0	-0,070117446	-0,051510448
p_{b2}	-0,0098426339	0	-0,0094142031	0	-0,018005136	0
p_{b3}	-0,92690419	-0,99953821	-0,89648272	-0,9921362	-1,0079009	-1,0847093
p_{b4}	0,00010682364	0	0,00010617388	0	0,000032749003	0
p_{b5}	0,00045497906	0	0,00016799714	0	-0,00016463736	0
p_{b6}	0,034611098	0,059871312	0,092009885	0,1433803	0,019713545	-0,023427678
p_{b7}	-0,0014323344	0	-0,0013833724	0	0,0013257499	0
p_{b8}	-0,11287128	0	-0,13619071	-0,01723058	-0,0059184347	0,033278629
p_{b9}	0,015742283	-0,059891933	-0,0389248	-0,13989362	-0,049067208	0,016956699
p_{b10}	0	0	0	0	0,035636702	0,027091235
p_{b11}	0	0	0	0	0,069766739	0,051149464
p_{b12}	0	0	0	0	0,059859018	0,17951705
p_{b13}	0	0	0,000000066766483	0	-0,00028688556	-0,086893296
p_{c1}	-0,44421604	-0,2243474	-0,3512382	-0,12701685	0,44515235	0,27352032
p_{c2}	-0,0096858895	0	-0,010120986	0	-0,0017744491	0
p_{c3}	0,0055107123	0	0,0016409903	0	-0,019426889	0
p_{c4}	0,0010038532	0	0,00071139628	0	0,00075656267	0
p_{c5}	1,9058952	0,83211798	1,8816769	0,86764862	1,3216582	0,97312677
p_{c6}	-1,1923422	-0,18780714	-1,1043952	-0,24677476	-0,70856314	-0,61612252
p_{c7}	-0,84888648	0,08943598	-0,84138088	-0,008298677	0,29621927	0,48035222
p_{c8}	0,73456158	0,16232721	0,65818496	0,081245945	-0,37826953	-0,31361945
p_{c9}	0,50816013	0,33300936	0,4602371	0,35322605	-0,34648177	-0,18888714
p_{c10}	-0,1471501	0	-0,12125449	0,0070439095	0,09488898	0,10277405
p_{c11}	0,46374796	0	0,4176159	0,052638078	0,2636648	0,32075334
p_{c12}	0	0	0	0	0,016601098	0
p_{c13}	0,00000013871821	0	0	0	-0,00088358873	0

Table A.7: Error reduction function parameters $\overline{I}$ (s, S)

Param	S3rQ δ_{1b}	S3rQ δ_{1a}	S3rQ δ_{2b}	S3rQ δ_{2a}	S3rQ δ_{3b}	S3rQ δ_{3a}
p_{a1}	0	0	0,085399496	0	1,0943654	1,3880446
p_{a2}	0	0	-0,15165073	0	0,50062075	0,34580753
p_{a3}	0	0	-1,1708071	-2,3393569	-4,9100706	-3,9502013
p_{a4}	0	0	-1,267889	-2,9987214	-7,1374419	-12,383092
p_{a5}	0	0	-2,0602956	-0,74499835	-1,2671549	2,3029312
p_{a6}	0	0	-5,8637482	4,4041362	6,9338997	7,9256682
p_{a7}	0	0	9,8011412	-3,7629032	2,892984	0,24917512
p_{a8}	0	0	0	0	-0,29892799	-0,39800407
p_{a9}	0	0	0	0	-1,043527	-1,3654603
p_{b1}	0,0061516398	0	-0,016611751	0	0,0057108984	0,044169686
p_{b2}	-0,013114407	0	-0,093060786	0	0,13472509	0,18601419
p_{b3}	1,1160387	1,0243127	2,1242357	1,3851264	0,95816674	0,62462838
p_{b4}	-0,00017220305	0	0,0001548575	0	0,000090312202	0
p_{b5}	-0,00014256374	0	0,004414882	0	-0,0020432094	0
p_{b6}	-0,18793433	-0,24104327	-0,066495961	0	0,5990843	0,9212463
p_{b7}	0,058776026	0,062914086	0,048068125	0	0,080109281	0
p_{b8}	0,19080206	0,45380749	-1,1456559	-0,18011099	0,9114942	0,52832674
p_{b9}	0,23937879	0,15444144	0,64479804	0,64127494	-1,1010966	-0,70875438
p_{b10}	0	0	0	0	-0,0035062859	0
p_{b11}	0	0	0	0	-0,015508881	-0,046687169
p_{b12}	0	0	0	0	1,465662	3,5532961
p_{b13}	0	0	0	0	-1,0667048	-2,690616
p_{c1}	0,077337854	0,035087078	0,12639855	0,048173773	0,035873378	0
p_{c2}	0,000095523076	0	0,00018049235	0	-0,00094787924	0
p_{c3}	-0,000053914397	0	-0,0001708423	0	-0,000029536141	0
p_{c4}	0,0000027533576	0	-0,00000048333429	0	-0,00000094691379	0
p_{c5}	0,00010907406	0	-0,013796107	0	0,0011532527	0
p_{c6}	0,0013185739	0,001257514	0,012239387	0	-0,00044558679	0
p_{c7}	0,036266244	0	0,067668835	0,0040243892	0,005939494	0
p_{c8}	-0,079818236	-0,035094308	-0,13195245	-0,049013198	-0,016030193	0
p_{c9}	-0,03461982	0	-0,059730544	0	-0,010310413	0,0056714571
p_{c10}	0,0041737555	0	0,0087774449	0	0,00163337	0
p_{c11}	-0,0016978361	0	-0,0068410228	0	0,00010514429	0
p_{c12}	0	0	0	0	-0,0010492133	0
p_{c13}	0	0	0	0	0,000052424791	0

Table A.8: Error reduction function parameters S_3 (r, Q)

Param	S3sS δ_{1b}	S3sS δ_{1a}	S3sS δ_{2b}	S3sS δ_{2a}	S3sS δ_{3b}	S3sS δ_{3a}
p_{a1}	0	0	-0,023523489	-0,066557217	0,79716786	1,1004365
p_{a2}	0	0	-0,22239121	-0,079148757	0,24563603	0,11116558
p_{a3}	0	0	2,561479	1,0607154	-2,5401893	-2,5189868
p_{a4}	0	0	-5,3174679	-9,0124531	-5,1786972	-4,2311883
p_{a5}	0	0	-3,0142827	0,85775502	-2,2230304	-1,4826028
p_{a6}	0	0	-13,108538	-3,5843943	0,75442799	-1,2026959
p_{a7}	0	0	16,398976	3,9693038	6,2873342	4,7480286
p_{a8}	0	0	0	0	-0,210914	-0,31146123
p_{a9}	0	0	0	0	-0,76441535	-1,0611184
p_{b1}	0,023261821	0,010602112	0,029995251	0,015383004	0,022651616	0,042505199
p_{b2}	-0,041788549	0	-0,063557997	0,0013438501	0,040855584	0,15363745
p_{b3}	0,96039315	0,69788195	0,88400159	0,74214382	0,8651097	0,66173617
p_{b4}	-0,00054507491	0	-0,00059914455	0	-0,0004558522	0
p_{b5}	0,00135193	0	0,0041701925	0	0,001538229	0
p_{b6}	-0,39202681	-0,66614885	0,9275488	0,82861836	0,40951903	-0,0270674
p_{b7}	0,063909296	0,082563833	0,080496624	0	0,082639597	0,073666401
p_{b8}	0,53672128	1,5132097	1,4155337	1,5956779	1,0643775	1,1108969
p_{b9}	0,27151237	-0,034507398	-1,4759625	-1,1659123	-0,96896191	-0,39122065
p_{b10}	0	0	0	0	0,00095402484	0
p_{b11}	0	0	0	0	-0,0076935887	-0,045377121
p_{b12}	0	0	0	0	1,3172217	2,898966
p_{b13}	0	0	0	0	-0,81698144	-2,0588137
p_{c1}	0,12062611	0,04830026	0,024222868	0	0,059882246	0,067215047
p_{c2}	0,00011392892	0	-0,00010053671	0	-0,0010037635	0
p_{c3}	-0,00023221194	0	-0,0000834355	0	-0,00010328889	0
p_{c4}	0,0000022411815	0	0,0000019777199	0	0,0000047278894	0
p_{c5}	-0,0065806764	0	0,0051643193	0	0,0049922801	0
p_{c6}	0,0090091832	0	-0,0011499499	0	-0,0016880004	0
p_{c7}	0,061551946	0	0,0073354859	-0,0041390341	0,014291823	0,021083482
p_{c8}	-0,12366612	-0,048292576	-0,021018733	0	-0,038026028	-0,067333684
p_{c9}	-0,059343686	0	-0,01341274	0,0059837697	-0,023130176	-0,020932926
p_{c10}	0,0079681805	0	0,0013872326	0	0,0023631857	0,001771745
p_{c11}	-0,006278154	0	-0,00082902164	0	-0,00064241617	0
p_{c12}	0	0	0	0	-0,001049244	0
p_{c13}	0	0	0	0	0,00005241951	0

Table A.9: Error reduction function parameters S_3 (s, S)

Param	BsS δ_{1b}	BsS δ_{1a}	BsS δ_{2b}	BsS δ_{2a}	BsS δ_{3b}	BsS δ_{3a}
p_{a1}	0	0	-0,013141666	0	0,045610567	0,083894246
p_{a2}	0	0	0,10585247	0	0,11102559	0
p_{a3}	0	0	-0,86105751	0	-1,1640106	-0,27618939
p_{a4}	0	0	-12,907645	-8,5711248	-8,1741138	-0,73045135
p_{a5}	0	0	-16,1782	-15,27667	-12,572759	-11,377109
p_{a6}	0	0	-0,55247016	0	6,4528212	5,5457076
p_{a7}	0	0	19,15342	3,6548146	5,2353	-13,957536
p_{a8}	0	0	0	0	0,0084453453	0
p_{a9}	0	0	0	0	-0,047178105	-0,065098304
p_{b1}	0,029368322	0,0026452142	0,021304269	0	-0,040287323	-0,090527551
p_{b2}	-0,15345475	-0,046744296	-0,33246519	0	-0,1972893	0
p_{b3}	-0,30596058	-0,40753976	0,38005522	-1,4801646	0,81426622	-0,10143568
p_{b4}	-0,0011518666	0	0,00062093455	0	0,00035270567	0
p_{b5}	0,008048889	0	0,0079543219	0	0,0014229548	0
p_{b6}	-1,4405609	-1,2337302	20,930056	12,034436	12,377779	-1,3464277
p_{b7}	0,56047723	0	33,050652	32,073722	25,962811	23,949538
p_{b8}	-2,0192791	-2,7303711	3,607392	2,2814716	-10,128964	-11,441936
p_{b9}	2,8699676	4,6387139	-36,734813	-6,7824148	-10,181715	27,660661
p_{b10}	0	0	0	0	-0,0087498648	0
p_{b11}	0	0	0	0	0,044580822	0,062227979
p_{b12}	0	0	0	0	0,030530765	0,037409253
p_{b13}	0	0	0	0	-0,011086771	0
p_{c1}	1,6750431	1,1355501	-7,8517908	-3,8335535	-4,1658243	2,1158533
p_{c2}	-0,55997324	0	-16,871833	-16,787725	-13,388373	-12,572273
p_{c3}	-0,0046452963	0	-0,002794364	0	0,00035266287	0
p_{c4}	0,00074707857	0	-0,00071466375	0	-0,0004036302	0
p_{c5}	0,46624503	0,33976156	0,50813039	1,136255	0,2764656	0,16047315
p_{c6}	-0,11049862	-0,044612443	0,10467248	-0,50680569	0,031000334	0
p_{c7}	2,3286767	2,9445753	-2,7239683	-1,5952473	3,923346	6,0625832
p_{c8}	-3,1213502	-4,5750724	17,378558	3,1286008	4,8410043	-13,774365
p_{c9}	-0,2580148	0	-0,19037294	0,38149443	-0,047928577	0,029283657
p_{c10}	0,052139526	0	0,053062277	-0,12342129	0,020907908	0
p_{c11}	-0,011740347	0	-0,079852988	0,27986063	-0,020380849	0,047080175
p_{c12}	0	0	0	0	-0,00038045208	0
p_{c13}	0	0	0	0	0,000020391285	0

Table A.10: Error reduction function parameters $\overline{B}$ (s, S)

Param	BrQ δ_{1b}	BrQ δ_{1a}	BrQ δ_{2b}	BrQ δ_{2a}	BrQ δ_{3b}	BrQ δ_{3a}
p_{a1}	0	0	0,033552024	0,0084996797	0,091572982	0,098727271
p_{a2}	0	0	0,066897257	0	0,095133477	0
p_{a3}	0	0	-1,5711771	-0,96803604	-1,7359944	-0,92395767
p_{a4}	0	0	-9,3049324	-2,8590993	-7,2254122	-8,7392022
p_{a5}	0	0	-24,086373	-40,819616	-16,196139	-11,149293
p_{a6}	0	0	-2,3642817	0	5,1178729	8,5216404
p_{a7}	0	0	25,130535	16,127741	9,0294157	-4,0585292
p_{a8}	0	0	0	0	-0,00059425586	0
p_{a9}	0	0	0	0	-0,052032554	-0,058419036
p_{b1}	0,020054038	-0,0033394728	-0,040679921	-0,0098007613	-0,10127791	-0,09396777
p_{b2}	-0,22610445	-0,04272605	-0,21751161	-0,057497187	-0,17850458	-0,050869315
p_{b3}	-0,30889157	-0,55950106	0,90322299	0,42281643	1,3180802	0,52434557
p_{b4}	-0,00033768972	0	0,00018888977	0	0,00029815249	0
p_{b5}	0,011624369	0	0,0053384655	0	0,0014943247	0
p_{b6}	-1,3027915	-1,6872465	14,160648	0	11,086973	14,59415
p_{b7}	-0,097990912	0	48,315479	82,119669	32,714948	22,748066
p_{b8}	-1,1556993	-2,0722247	8,9812892	3,6545952	-6,1257914	-15,325987
p_{b9}	2,1050176	4,7108023	-49,537036	-32,369029	-18,649202	8,0460466
p_{b10}	0	0	0	0	0,00066983096	0
p_{b11}	0	0	0	0	0,050484555	0,056865999
p_{b12}	0	0	0	0	0,04905653	0
p_{b13}	0	0	0	0	-0,035305342	0
p_{c1}	1,5190271	1,692528	-4,8581299	2,8560255	-3,9106454	-5,8975889
p_{c2}	0,097215411	0	-24,226568	-41,296012	-16,516449	-11,594704
p_{c3}	-0,0061694587	0	-0,0014870891	0	0,00052671721	0
p_{c4}	0,00011035401	0	-0,000087907438	0	-0,00018595935	0
p_{c5}	0,49107953	0,3571005	0,52348689	0,35305486	0,21779782	0,24547528
p_{c6}	0,031845643	-0,031361668	0,045272614	0	0,1225353	0
p_{c7}	1,38206	2,4147437	-6,2579147	-3,2463954	1,2546723	7,1482715
p_{c8}	-2,2970276	-4,7722824	24,299702	16,09246	9,5730315	-4,0884021
p_{c9}	-0,24837688	0	0,00018992301	0	0,036861917	0
p_{c10}	0,059479723	0	0,0072992099	0	0,013293022	0,017389878
p_{c11}	-0,073772699	0,029479354	-0,0046283332	0	-0,030366051	0
p_{c12}	0	0	0	0	0,00001266313	0
p_{c13}	0	0	0	0	-0,00000032137084	0

Table A.11: Error reduction function parameters $\overline{B}$ (r, Q)

Param	OFsS δ_{1b}	OFsS δ_{1a}	OFsS δ_{2b}	OFsS δ_{2b}
p_{a1}	0	0	-0.40184933	-0.37245727
p_{a2}	0	0	-0.39893309	-0.43560539
p_{a3}	0	0	1.0439325	0.96323606
p_{a4}	0	0	-0.22094488	-0.1558756
p_{a5}	0	0	0.0095930268	0.0066710507
p_{a6}	0	0	-0.183109	-0.201526
p_{a7}	0	0	0.023855756	0.025503617
p_{a8}	0	0	0	0
p_{a9}	0	0	0	0
p_{b1}	1.0825233	1.0443514	1.0948724	1.0695131
p_{b2}	0.0011437009	0.0008733351	0.057753116	0.070466173
p_{b3}	-0.0037754807	-0.0022813356	-0.1307431	-0.089241301
p_{b4}	0.000025363408	0	0.000054158232	0
p_{b5}	-0.08007622	-0.073589227	-0.053196367	-0.049419443
p_{b6}	0.00020387933	0	0.0249364	0
p_{b7}	-0.000013508497	0	-0.0010740496	0
p_{b8}	0.0035132831	0.00099487197	0.023068394	0.032656664
p_{b9}	-0.00056869978	0	-0.0028905064	-0.0035057537
p_{b10}	0	0	0	0
p_{b11}	0	0	0	0
p_{b12}	0	0	0	0
p_{b13}	0	0	0	0
p_{c1}	-0.0013839936	0	-0.0019658246	0
p_{c2}	0.000051429025	0	0.000071330515	0
p_{c3}	0.0021895006	0.0017203621	0.0011365446	0
p_{c4}	-0.000049320344	0	-0.000044573323	0
p_{c5}	0.0071996099	0	0.0088980897	0
p_{c6}	-0.012176639	0.00053031108	-0.014680838	0.0036977038
p_{c7}	0.0015231389	0	0.0021901296	0
p_{c8}	-0.00011910586	0	-0.0001683209	0
p_{c9}	0.00037049671	0	0.00068804045	0
p_{c10}	-0.00038345406	0	-0.00067332631	0
p_{c11}	0.0027653244	-0.0011086211	0.0043885429	-0.0050080494
p_{c12}	0	0	0	0
p_{c13}	0	0	0	0

Table A.12: Error reduction function parameters $\overline{OF}$ (s, S)

A.3 SN1OLF rational functions

A.3.1 SN1OLF rational function expressions

The parameters for the SN1OLF rational functions are given in the next section A.3.2.

$$p = e(((((((a_7x + a_6)x + a_5)x + a_4)x + a_3)x + a_2)x + a_1)x + a_0) \\ /(((((((b_7x + b_6)x + b_5)x + b_4)x + b_3)x + b_2)x + b_1)x + 1) \quad \text{(A.4)}$$

$$p = e(((((((c_7x + c_6)x + c_5)x + c_4)x + c_3)x + c_2)x + c_1)x + c_0) \\ /(((((((d_6x + d_5)x + d_4)x + d_3)x + d_2)x + d_1)x + 1) \quad \text{(A.5)}$$

$$p = e((((((e_6x + e_5)x + e_4)x + e_3)x + e_2)x + e_1)x + e_0) \\ /(((((((f_6x + f_5)x + f_4)x + f_3)x + f_2)x + f_1)x + 1) \quad \text{(A.6)}$$

e in the previous equations is given by (4.18).

A.3.2 SN1OLF rational function parameters

Here we provide the list of used parameters for the rational functions of the SN1OLF algorithm and its rational function expressions from A.3.1.

$a_0 = 0.39894228040143268$
$a_1 = 0.21002474394780039$
$a_2 = 0.071246581285295085$
$a_3 = 0.013032473433200309$
$a_4 = 0.0014595736472943335$
$a_5 = 0.68547041219742437e-4$
$a_6 = 0.99733431803299761e-7$
$a_7 = -0.26957666631539149e-8$

$b_1 = 1.7797680988671933$
$b_2 = 1.4091972145738764$
$b_3 = 0.64572332747815564$
$b_4 = 0.18572649551025895$
$b_5 = 0.033713766857832489$
$b_6 = 0.0036017759390713289$
$b_7 = 0.00017644910556614172$

$c_0 = 0.062738277955091465$
$c_1 = 0.03502297980288093$
$c_2 = 0.0086444588653607395$
$c_3 = 0.0010417848156028285$
$c_4 = 0.51961903750684434e-4$
$c_5 = -0.78871551226211378e-8$
$c_6 = 0.32785620480866639e-9$
$c_7 = -0.68891223622269787e-11$

$d_1 = 1.2376563024482419$
$d_2 = 0.65808751220578398$
$d_3 = 0.19293566790434162$
$d_4 = 0.033000199681586271$
$d_5 = 0.0031350224464661629$
$d_6 = 0.00012992843915945728$

$e_0 = 0$
$e_1 = 0.39894228040143246$
$e_2 = 22.685111028866376$
$e_3 = 434.7910643487663$
$e_4 = 3356.8807938223652$
$e_5 = 9642.0097656947831$
$e_6 = 6953.32433232037$

$f_1 = 59.863140718098526$ $f_2 = 1254.4489976092465$
$f_3 = 11384.852193724395$ $f_4 = 44847.385892906192$
$f_5 = 66740.214462756024$ $f_6 = 24612.052740704975$

A.4 SN2OLF rational functions

A.4.1 SN2OLF rational function expressions

The parameters for the SN2OLF rational functions are given in the next section A.4.2.

$$p = e(((((((a_7x + a_6)x + a_5)x + a_4)x + a_3)x + a_2)x + a_1)x + a_0) \\ /(((((((b_7x + b_6)x + b_5)x + b_4)x + b_3)x + b_2)x + b_1)x + 1) \quad \text{(A.7)}$$

$$p = e(((((((c_7x + c_6)x + c_5)x + c_4)x + c_3)x + c_2)x + c_1)x + c_0) \\ /(((((((d_7x + d_6)x + d_5)x + d_4)x + d_3)x + d_2)x + d_1)x + 1) \quad \text{(A.8)}$$

$$p = (e/r)(((((((e_7x + e_6)x + e_5)x + e_4)x + e_3)x + e_2)x + e_1)x + e_0) \\ /(((((((f_7x + f_6)x + f_5)x + f_4)x + f_3)x + f_2)x + f_1)x + 1) \quad \text{(A.9)}$$

e in the previous equations is given by (4.18).

A.4.2 SN2OLF rational function parameters

Here we provide the list of used parameters for the rational functions of the SN2OLF algorithm and its rational function expressions from A.4.1.

$a_0 = 0.25$ $a_1 = 0.059466962529246364$
$a_2 = 0.018355620244174282$ $a_3 = 0.0013812459438037644$
$a_4 = 0.00014355195520570905$ $a_5 = -0.42377117322615563e-5$
$a_6 = 0.23809047340616304e-6$ $a_7 = -0.66469958200808304e-8$

$b_1 = 1.8336369717227178$ $b_2 = 1.4994837406863502$
$b_3 = 0.71174545900099654$ $b_4 = 0.21283752915627627$
$b_5 = 0.040355035917229408$ $b_6 = 0.0045311760763075409$
$b_7 = 0.00023537531946841132$

$c_0 = 0.021312722656493838$ $c_1 = 0.010215604212941093$
$c_2 = 0.0022887091738377789$ $c_3 = 0.00025389013286717049$
$c_4 = 0.11978235751305706e-4$ $c_5 = -0.34553072087589492e-8$
$c_6 = 0.13207941962153137e-9$ $c_7 = -0.25413412919933679e-11$

$d_1 = 1.4230202937400878$ $d_2 = 0.89399309288656039$
$d_3 = 0.32221715492195554$ $d_4 = 0.072173572962398247$
$d_5 = 0.010084050800734187$ $d_6 = 0.00081764995576340292$
$d_7 = 0.29873078267110186e-4$

$e_0 = 0$
$e_1 = 0.39894228040143267$
$e_2 = 33.297276519443342$
$e_3 = 998.54933699193513$
$e_4 = 13241.10184554912$
$e_5 = 76008.291350745479$
$e_6 = 147125.44161757116$
$e_7 = 3819.8748949681348$

$f_1 = 89.463894791843534$
$f_2 = 2994.7753704691611$
$f_3 = 47553.297230451145$
$f_4 = 373929.25975228526$
$f_5 = 1369924.7218087889$
$f_6 = 1858297.0898555859$
$f_7 = 236391.73034752403$

A.5 ISN1OLF rational functions

A.5.1 ISN1OLF rational function expressions

Here we list the rational functions used in the Algorithm ISN1OLF. The parameters for the SN2OLF rational functions are given in the next section A.5.2.

$$\begin{aligned} z = -p + \left[(\exp(-0.5p^2)/p^2\right] * \\ ((((((((((m_{10}x + m_9)x + m_8)x + m_7)x + m_6)x + m_5)x \\ + m_4)x + m_3)x + m_2)x + m_1)x + m_0) \\ /((((((((((n_{10}x + n_9)x + n_8)x + n_7)x \\ + n_6)x + n_5)x + n_4)x + n_3)x + n_2)x + n_1)x + 1) \end{aligned} \quad \text{(A.10)}$$

$$\begin{aligned} z = -p + \left[(\exp(-0.5p^2)/\sqrt{p}\right] * \\ (((((((k_7x + k_6)x + k_5)x + k_4)x + k_3)x + k_2)x + k_1)x + k_0) \\ /(((((((l_7x + l_6)x + l_5)x + l_4)x + l_3)x + l_2)x + l_1)x + 1) \end{aligned} \quad \text{(A.11)}$$

$$\begin{aligned} z = ((((((i_6x + i_5)x + i_4)x + i_3)x + i_2)x + i_1)x + i_0) \\ /(((((j_5x + j_4)x + j_3)x + j_2)x + j_1)x + 1) \end{aligned} \quad \text{(A.12)}$$

$$\begin{aligned} z = (((((g_5x + g_4)x + g_3)x + g_2)x + g_1)x + g_0) \\ /((((h_4x + h_3)x + h_2)x + h_1)x + 1) \end{aligned} \quad \text{(A.13)}$$

$$\begin{aligned} z = y * ((((((((e_8x + e_7)x + e_6)x + e_5)x \\ + e_4)x + e_3)x + e_2)x + e_1)x + e_0) \\ /(((((((((f_8x + f_7)x + f_6)x + f_5)x \\ + f_4)x + f_3)x + f_2)x + f_1)x + 1) \end{aligned} \quad \text{(A.14)}$$

$$z = y * ((((((((c_8x + c_7)x + c_6)x + c_5)x + c_4)x + c_3)x + c_2)x + c_1)x + c_0) / ((((((((d_8x + d_7)x + d_6)x + d_5)x + d_4)x + d_3)x + d_2)x + d_1)x + 1) \quad \text{(A.15)}$$

$$z = y * ((((((((a_8x + a_7)x + a_6)x + a_5)x + a_4)x + a_3)x + a_2)x + a_1)x + a_0) / ((((((((b_8x + b_7)x + b_6)x + b_5)x + b_4)x + b_3)x + b_2)x + b_1)x + 1) \quad \text{(A.16)}$$

A.5.2 ISN1OLF rational function parameters

Here we provide the list of used parameters for the rational functions of the ISN1OLF algorithm and its rational function expressions from A.5.1.

$a_0 = 1.310444679913372$
$a_1 = 1.0496342982777808$
$a_2 = 0.34297377276691108$
$a_3 = 0.058806303584142083$
$a_4 = 0.0056707052143933359$
$a_5 = 0.00030646561311550538$
$a_6 = 0.87299508529791606e-5$
$a_7 = 0.11359029676757079e-6$
$a_8 = 0.47715414930209652e-9$

$b_1 = 0.77935982713146056$
$b_2 = 0.24949458983424002$
$b_3 = 0.04218517946986447$
$b_4 = 0.0040342579952242503$
$b_5 = 0.00021712896462726042$
$b_6 = 0.61753952795739417e-5$
$b_7 = 0.8032129209966964e-7$
$b_8 = 0.33739809465660943e-9$

$c_0 = 0.58234615354415903$
$c_1 = 2.3954910046771972$
$c_2 = 3.7644259282127282$
$c_3 = 3.1822730248163338$
$c_4 = 1.5804886582031085$
$c_5 = 0.45927351465091796$
$c_6 = 0.071752521699595843$
$c_7 = 0.0051073685232919929$
$c_8 = 0.00011544405570833181$

$d_1 = 2.8868777130708606$
$d_2 = 3.7190165916935734$
$d_3 = 2.7459319764867306$
$d_4 = 1.2389548635486364$
$d_5 = 0.33776812350821943$
$d_6 = 0.051177121304169432$
$d_7 = 0.0036125038564975782$
$d_8 = 0.81625833768166845e-4$

$e_0 = 3.0018269039209338$
$e_1 = 87.774236056726657$
$e_2 = 989.7575959597692$
$e_3 = 5425.7139678862877$
$e_4 = 15009.178138029023$
$e_5 = 19803.67505848015$
$e_6 = 10861.978269562645$
$e_7 = 2360.7711979594225$
$e_8 = 394.32281881668326$

$f_1 = 31.891192048993958$
$f_2 = 399.96189801714692$
$f_3 = 2505.8535715458245$
$f_4 = 8237.9586584846684$
$f_5 = 13666.751461068495$
$f_6 = 10067.57909732112$
$f_7 = 2583.8131935382122$
$f_8 = 292.57249970310298$

$g_0 = 0.61175758417034084e-16$
$g_1 = 0.062499999999999994$
$g_2 = -0.0072754785571224347$
$g_3 = 0.00020727510842806252$
$g_4 = 0.82125690103447695e-7$
$g_5 = -0.21022601430286476e-7$

$h_1 = -0.14134154943904849$
$h_2 = 0.0055971987449862891$
$h_3 = -0.31892520349077855e-4$
$h_4 = -0.88983818772332 07e-6$

$i_0 = -0.49846544058069863e-16$
$i_1 = -0.25000000000000002$
$i_2 = -0.15242142496267809$
$i_3 = -0.030445417368412181$
$i_4 = -0.0023029650958645681$
$i_5 = -0.68276543115638487e-4$
$i_6 = -0.12757147423548785e-5$

$j_1 = 0.70942126995107004$
$j_2 = 0.17264183638706689$
$j_3 = 0.016757888840674101$
$j_4 = 0.00058599116642757624$
$j_5 = 0.76994691214814959e-5$

$k_0 = 0.24488851831921104$
$k_1 = 0.48029713677686631$
$k_2 = 0.18301189125291796$
$k_3 = -0.026160178329977429$
$k_4 = 0.021184219173857513$
$k_5 = -0.0029987754713519703$
$k_6 = 0.0007233393146827096$
$k_7 = -0.44262578270966203e-4$

$l_1 = 2.5410054607988286$
$l_2 = 2.0418856609447443$
$l_3 = 0.5529951895300247$
$l_4 = 0.055678606510953158$
$l_5 = 0.017367680496451343$
$l_6 = 0.0062574676815150778$
$l_7 = 0.00039215993062993774$

$m_0 = 0.18756233208103297$
$m_1 = 0.31676099852893919$
$m_2 = 0.18050826111441451$
$m_3 = 0.075146244143085796$
$m_4 = 0.036805415000645036$
$m_5 = 0.0093534945621974929$
$m_6 = 0.0034409217608655729$
$m_7 = 0.00063248029913634361$
$m_8 = 0.00019426953835475993$
$m_9 = 0.16897829742340167e-4$
$m_{10} = 0.80030048496041983e-5$

$n_1 = 1.1106105568550152$
$n_2 = 0.52698392742028986$
$n_3 = 0.23366691487529792$
$n_4 = 0.09983456392019021$
$n_5 = 0.031404493662692849$
$n_6 = 0.007561621875344939$
$n_7 = 0.0020956017866871155$
$n_8 = 0.00047749988216206908$
$n_9 = 0.47477680185174772e-4$
$n_{10} = 0.19898093244659233e-4$

A.6 ISN2OLF rational functions

A.6.1 ISN2OLF rational function expressions

Here we list the rational functions used in the Algorithm ISN1OLF. The parameters for the SN2OLF rational functions are given in the next section A.6.2.

$$z = -\sqrt{2p-1} - [(((k_3x + k_2)x + k_1)x + k_0) \\ /((l_2x + l_1)x + 1)]/(p^2 exp(p)) \quad \text{(A.17)}$$

$$z = \sqrt{2p-1} - [(((((((((i_8x + i_7)x + i_6)x + i_5)x + i_4)x + i_3)x + i_2)x + i_1)x + i_0) / (((((((j_7x + j_6)x + j_5)x + j_4)x + j_3)x + j_2)x + j_1)x + 1)]/(p^2 exp(p)) \quad \text{(A.18)}$$

$$z = (p - 0.25)((((((((((g_{10}x + g_9)x + g_8)x + g_7)x + g_6)x + g_5)x + g_4)x + g_3)x + g_2)x + g_1)x + g_0) / ((((((((h_9x + h_8)x + h_7)x + h_6)x + h_5)x + h_4)x + h_3)x + h_2)x + h_1)x + 1) \quad \text{(A.19)}$$

$$z = (0.25 - p)((((((e_6x + e_5)x + e_4)x + e_3)x + e_2)x + e_1)x + e_0) / ((((((f_6x + f_5)x + f_4)x + f_3)x + f_2)x + f_1)x + 1) \quad \text{(A.20)}$$

$$z = y * (((((((((c_9x + c_8)x + c_7)x + c_6)x + c_5)x + c_4)x + c_3)x + c_2)x + c_1)x + c_0) / ((((((((d_9x + d_8)x + d_7)x + d_6)x + d_5)x + d_4)x + d_3)x + d_2)x + d_1)x + 1) \quad \text{(A.21)}$$

$$z = y * (((((((((a_9x + a_8)x + a_7)x + a_6)x + a_5)x + a_4)x + a_3)x + a_2)x + a_1)x + a_0) / (((((((b_8x + b_7)x + b_6)x + b_5)x + b_4)x + b_3)x + b_2)x + b_1)x + 1) \quad \text{(A.22)}$$

A.6.2 ISN2OLF rational function parameters

Here we provide the list of used parameters for the rational functions of the ISN2OLF algorithm and its rational function expressions from A.6.1.

$a_0 = 5.2198298531688834$
$a_1 = 7.1201527678911184$
$a_2 = 4.0739724299280747$
$a_3 = 1.2669804338888258$
$a_4 = 0.23180079430882858$
$a_5 = 0.025237719381674411$
$a_6 = 0.0015819133851575586$
$a_7 = 0.52650219825908774e-4$
$a_8 = 0.79459303700632415e-6$
$a_9 = 0.38024526839979055e-8$

$b_1 = 1.0673186810264216$
$b_2 = 0.46729981833312222$
$b_3 = 0.10740205955064567$
$b_4 = 0.013760939980712562$
$b_5 = 0.0009683499066951873$
$b_6 = 0.3481642746061499e-4$
$b_7 = 0.54994180705908756e-6$
$b_8 = 0.26887279808569303e-8$

$c_0 = 0.39809887620896078$
$c_1 = 2.2684791327198945$
$c_2 = 3.508852142313804$
$c_3 = 2.9327917474980908$
$c_4 = 1.4398260352034985$
$c_5 = 0.37513333511116429$
$c_6 = 0.01101914722332671$
$c_7 = -0.023767990201524592$
$c_8 = -0.006273971048499002$
$c_9 = -0.00044295558464790916$

$d_1 = 1.7407135490813085$
$d_2 = 1.579510089960357$
$d_3 = 0.8178759912045096$
$d_4 = 0.23316641869675973$
$d_5 = 0.013450822538275304$
$d_6 = -0.012934400112314594$
$d_7 = -0.0040067813159069966$
$d_8 = -0.00031280736215242744$
$d_9 = -0.6008115162653259e-8$

$e_0 = 3.2418704589022945$
$e_1 = 8.0517058410168211$
$e_2 = 7.2511010225147658$
$e_3 = 2.8781817211447518$
$e_4 = 0.48581400669268295$
$e_5 = 0.027546558547237258$
$e_6 = 0.00023309131706558357$

$f_1 = 2.83278290598764$
$f_2 = 3.0221254939939149$
$f_3 = 1.5074963751663713$
$f_4 = 0.35382835486750113$
$f_5 = 0.03423435350038521$
$f_6 = 0.00091098059445654428$

$g_0 = -2.5066282746310005$
$g_1 = -4.0758059576080205$
$g_2 = -2.648577122745974$
$g_3 = -0.88496508978927867$
$g_4 = -0.16349523069287772$
$g_5 = -0.016802187200224274$
$g_6 = -0.00092374517397341856$
$g_7 = -0.24739020466270336e-4$
$g_8 = -0.26075688491270463e-6$
$g_9 = -0.50763803346039848e-9$
$g_{10} = 0.38410357558821514e-12$

$h_1 = 1.8223608640512509$
$h_2 = 1.353705293439588$
$h_3 = 0.52993788827705204$
$h_4 = 0.1183052764283123$
$h_5 = 0.01527861628071871 5$
$h_6 = 0.0011113982886871742$
$h_7 = 0.42453596359550705e-4$
$h_8 = 0.73510591959860863e-6$
$h_9 = 0.39466243968973455e-8$

$i_0 = 0.14020972127662181$
$i_1 = 0.068079860885949756$
$i_2 = -0.048768499520436906$
$i_3 = 0.010568466878658061$
$i_4 = -0.0011641165848097204$
$i_5 = 0.71693389721573075e-4$
$i_6 = -0.23713811542172557e-5$
$i_7 = 0.33058654513583253e-7$
$i_8 = -0.41371292946094178e-12$

$j_1 = 0.49841377673084252$
$j_2 = -0.34045426916659383$
$j_3 = 0.071551678401172788$
$j_4 = -0.007669174370693392$
$j_5 = 0.00046011838108922945$
$j_6 = -0.14822897041059011e-4$
$j_7 = 0.20107322967847599e-6$

$k_0 = 0.1547407329026946$
$k_1 = -0.008439916324944443$
$k_2 = 0.00011207299681885017$
$k_3 = 0.13345453831138152e-9$

$l_1 = -0.052691158482935122$
$l_2 = 0.0006815223132616962$

B

Model extensions and derivatives

B.1 Stochastic leadtimes

The customer demand can be stochastic, but also the leadtime is not necessarily constant and can also be stochastic. Zipkin (2000) gives a good overview of the impact of a stochastic leadtime. We can consider three categories:

1. Independent stochastic leadtimes
2. Limited-capacity supply systems
3. Exogenous sequential supply systems

The first category, independent stochastic leadtimes, is a parallel processing system, with infinite capacity that works independently. In this case an order that is placed later than a considered order may arrive earlier, so the FIFO (first in first out) principle is not respected. Here leadtime uncertainty has no impact in case of a base-stock system with Poisson demand and little impact in case of an (r, Q) system. For an (r, Q) system that uses a normal approximation, we need to redefine σ^2 by (B.1). Here L_2 indicates a random variable that is the minimum of two independent copies of L, so $L_2 = min\left\{L^1, L^2\right\}$.

$$\sigma^2 = \lambda E[L] + min\left\{\lambda(q-1)(E[L] - E[L_2]), \lambda^2 V[L]\right\} \tag{B.1}$$

Systems with limited capacity can have several forms (single process, series, parallel, assembly, ...). As there may bottlenecks due to limited capacity, this has a direct impact on the leadtime and the system performance

indicators. Here each order's leadtime depends on the encountered bottlenecks and work in progress in the system that depends on previous orders, this system is called endogenous. These limited capacity cases are not considered in this study.

The third considered category is the exogenous sequential supply systems, where the effect of our own orders on the whole system can be neglected. A special case here is the exogenous sequential supply system, where the FIFO rule is respected for the placed orders, but where we assume an incapacitated system. If we approximate D, that incorporates the effect of variation on customer demand and variation on leadtime, we have the following approximation for the mean (B.2) and variation (B.3) of D_L in case of stochastic leadtime.

$$\nu = E[D_L] = \lambda E[L] \tag{B.2}$$

$$\sigma^2 = V[D_L] = \Psi^2 E[L] + \lambda^2 V[L] \tag{B.3}$$

Zipkin (1986b) showed that the net inventory IN and the inventory position IP distributions in the standard continuous-time model with backorders, see B.4, are valid under broad conditions of stochastic leadtime if the leadtimes are generated according a specific but plausible scenario. These include cases where the leadtime consists of travel time and also settings where a supplier processes orders according to a queuing process, provided many orders arrive also from other independent sources. This mechanism is to ensure orders never cross. The demand processes can be of general classes of compound-counting processes and it also stands in a variety of order policies: (r, Q) and (s, S).

$$IL(\infty) = IP(\infty) - D_L(\infty) \tag{B.4}$$

B.2 Periodic review

For all the previous discussed inventory models we assumed continuous review. Where in the early days this was not possible due to the lack of support IT systems to do the continuous monitoring of the stock, nowadays we no longer see this as a reason to prevent applying a continuous review system. On the other hand there may be other reasons to prefer a periodic review, which is easier to manage and which does not require an expensive IT system. A periodic review system allows you to evaluate weekly your stock, which could allow you to group some orders from the same supplier and will enable us to group the receiving of goods. So here we will give a brief explanation on the impact of a periodic review while still applying a continuous time-line.

We start from the base-stock policy, because this is the basis of the (r, Q) and (s, S) order policy as we have seen. In periodic review base-stock

policy where we have a review period T and then order a quantity to bring the inventory position IP up to the target stock level s. As explained in Zipkin (2000) the net inventory process **IN** does not have a true limiting distribution, due to the cyclic behavior of the order policy, but it does has a long-run frequency distribution: $\underline{\text{IN}}$. In between review point ($0 \leq t \leq T$), the relation between the net inventory IN and the inventory on order IO is still valid, see B.5. If we denote the uniform mixture of demand over the interval $[0, T)$ as $\underline{D}$, we can state B.6.

$$IN(t) = s - IO(t) \tag{B.5}$$

$$\underline{IN} = s - \underline{D} \tag{B.6}$$

From here on there is a full resemblance with the classic base-stock policy, we only need to redefine the leadtime variable demand. Here we use a conditional probability mass function where $g(d|t)$ is a function of d depending on the parameter t, and t can vary over the range $[0, T)$.

$$g(d|t) = (\nu t)^d e^{-\nu t}/d! \tag{B.7}$$

$$\begin{aligned} g_{\underline{D}}(d) &= \frac{1}{T}\int_0^T g(d|t+L)dt \\ &= \frac{1}{\lambda T}\left[G^0(d|L+T) - G^0(d|L)\right] \end{aligned} \tag{B.8}$$

$$G^0_{\underline{D}}(d) = \frac{1}{\lambda T}\left[G^1(d+1|L+T) - G^1(d+1|L)\right] \tag{B.9}$$

$$G^1_{\underline{D}}(d) = \frac{1}{\lambda T}\left[G^2(d|L+T) - G^2(d|L)\right] \tag{B.10}$$

Using these redefined demand functions we end up with the same definitions for the KPI's:

$$\overline{A} = G^0_{\underline{D}}(s-1) \tag{B.11}$$

$$\overline{B} = G^1_{\underline{D}}(s) \tag{B.12}$$

$$\overline{P} = \lambda \overline{A} \tag{B.13}$$

$$\overline{I} = s - \lambda(L + \frac{T}{2}) + \overline{B} \tag{B.14}$$

For the normal approximation of this periodic review model we can apply the same approach.

For other combinations of periodic review and how to model them we refer to Rosling (2002a). Johnson et al. (1995) test a variety of approximations in a periodic inventory system and they also show that simpler fill rate expressions perform poorly and underestimate the real fill rate.

B.3 KPI partial derivatives

We will first give some overall derivative rules for r and Q for the first and second order normal loss function: (B.15)-(B.22).

$$\frac{\partial \Phi^2(z_{(r+0.5)})}{\partial r} = -\frac{\Phi^1(z_{(r+0.5)})}{\sigma} \tag{B.15}$$

$$\frac{\partial \Phi^2(z_{(r+0.5)})}{\partial Q} = 0 \tag{B.16}$$

$$\frac{\partial \Phi^1(z_{(r+0.5)})}{\partial r} = -\frac{\Phi^0(z_{(r+0.5)})}{\sigma} \tag{B.17}$$

$$\frac{\partial \Phi^1(z_{(r+0.5)})}{\partial Q} = 0 \tag{B.18}$$

$$\frac{\partial \Phi^2(z_{(r+0.5+Q)})}{\partial r} = -\frac{\Phi^1(z_{(r+0.5+Q)})}{\sigma} \tag{B.19}$$

$$\frac{\partial \Phi^2(z_{(r+0.5+Q)})}{\partial Q} = -\frac{\Phi^1(z_{(r+0.5+Q)})}{\sigma} \tag{B.20}$$

$$\frac{\partial \Phi^1(z_{(r+0.5+Q)})}{\partial r} = -\frac{\Phi^0(z_{(r+0.5+Q)})}{\sigma} \tag{B.21}$$

$$\frac{\partial \Phi^1(z_{(r+0.5+Q)})}{\partial Q} = -\frac{\Phi^0(z_{(r+0.5+Q)})}{\sigma} \tag{B.22}$$

B.3.1 Q partial derivatives

Here we make use of the following KPI definitions: (5.9)-(5.16). We also use the normal function derivative rules: (B.15)-(B.22).

Average backorders

$$\begin{aligned}\frac{\partial \widetilde{B}_{cP}}{\partial Q} &= -p_{(\overline{B}d)}\frac{\sigma^2}{Q^2}\left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)})\right] \\ &\quad + p_{(\overline{B}d)}\frac{\sigma^2}{Q}\left[\frac{\Phi^1(z_{(r+0.5+Q)})}{\sigma}\right]\end{aligned} \tag{B.23a}$$

$$\begin{aligned}= -p_{(\overline{B}d)}\frac{\sigma^2}{Q^2}\Bigg[&\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) \\ &- \frac{Q\Phi^1(z_{(r+0.5+Q)})}{\sigma}\Bigg]\end{aligned} \tag{B.23b}$$

Average inventory

$$\frac{\partial \widetilde{I}_{cP}}{\partial Q} = \frac{p_{(\overline{I}d)}}{2} - p_{(\overline{I}d)}\frac{\sigma^2}{Q^2}\left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)})\right] + p_{(\overline{I}d)}\frac{\sigma^2}{Q}\left[\frac{\Phi^1(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.24a}$$

$$= \frac{p_{(\overline{I}d)}}{2} - p_{(\overline{I}d)}\frac{\sigma^2}{Q^2}\left[\Phi^2(z_{(r+0.5)}) - \Phi^2(z_{(r+0.5+Q)}) - \frac{Q\Phi^1(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.24b}$$

Fill rate service level

$$\frac{\partial \widetilde{S}_{2cP}}{\partial Q} = p_{(S2b)}\frac{\sigma}{Q^2}\left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)})\right] - p_{(S2b)}\frac{\sigma}{Q}\left[\frac{\Phi^0(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.25a}$$

$$= p_{(S2b)}\frac{\sigma}{Q^2}\left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) - \frac{Q\Phi^0(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.25b}$$

Ready rate service level

$$\frac{\partial \widetilde{S}_{3cP}}{\partial Q} = p_{(S3b)}\frac{\sigma}{Q^2}\left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)})\right] - p_{(S3b)}\frac{\sigma}{Q}\left[\frac{\Phi^0(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.26a}$$

$$= p_{(S3b)}\frac{\sigma}{Q^2}\left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) - \frac{Q\Phi^0(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.26b}$$

Order line service level

$$\frac{\partial \widetilde{S}_{OLcP}}{\partial Q} = p_{(SOLb)}\frac{\sigma}{Q^2}\left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)})\right] - p_{(SOLb)}\frac{\sigma}{Q}\left[\frac{\Phi^0(z_{(r+0.5+Q)})}{\sigma}\right] \tag{B.27a}$$

$$= p_{(SOLb)} \frac{\sigma}{Q^2} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) - \frac{Q\Phi^0(z_{(r+0.5+Q)})}{\sigma} \right] \tag{B.27b}$$

Order frequency

$$\frac{\partial \widetilde{OF}_{cP}}{\partial Q} = \frac{-p_{(\overline{OF}b)}}{Q^2} \tag{B.28a}$$

Stockout frequency

$$\frac{\partial \widetilde{A}_{cP}}{\partial Q} = -\frac{\partial \widetilde{S}_{3cP}}{\partial Q} \tag{B.29a}$$

$$= -p_{(S3b)} \frac{\sigma}{Q^2} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) - \frac{Q\Phi^0(z_{(r+0.5+Q)})}{\sigma} \right] \tag{B.29b}$$

Average new backorders

$$\frac{\partial \widetilde{P}_{cP}}{\partial Q} = -\lambda \frac{\partial \widetilde{S}_{2cP}}{\partial Q} \tag{B.30a}$$

$$= -\lambda p_{(S2b)} \frac{\sigma}{Q^2} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) - \frac{Q\Phi^0(z_{(r+0.5+Q)})}{\sigma} \right] \tag{B.30b}$$

B.3.2 r partial derivatives

Here we make use of the following KPI definitions: (5.9)-(5.16). We also use the normal function derivative rules: (B.15)-(B.22).

Average backorders

$$\frac{\partial \widetilde{B}_{cP}}{\partial r} = -p_{(\overline{B}d)} \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right] \tag{B.31a}$$

Average inventory

$$\frac{\partial \widetilde{I}_{cP}}{\partial r} = p_{(\bar{I}d)} \left[1 - \frac{\sigma}{Q} \left[\Phi^1(z_{(r+0.5)}) - \Phi^1(z_{(r+0.5+Q)}) \right] \right] \tag{B.32a}$$

Fill rate service level

$$\frac{\partial \widetilde{S}_{2cP}}{\partial r} = \frac{p_{(S2b)}}{Q} \left[\Phi^0(z_{(r+0.5)}) - \Phi^0(z_{(r+0.5+Q)}) \right] \tag{B.33a}$$

Ready rate service level

$$\frac{\partial \widetilde{S}_{3cP}}{\partial r} = \frac{p_{(S3b)}}{Q} \left[\Phi^0(z_{(r+0.5)}) - \Phi^0(z_{(r+0.5+Q)}) \right] \tag{B.34a}$$

Order line service level

$$\frac{\partial \widetilde{S}_{OLcP}}{\partial r} = \frac{p_{(SOLb)}}{Q} \left[\Phi^0(z_{(r+0.5)}) - \Phi^0(z_{(r+0.5+Q)}) \right] \tag{B.35a}$$

Order frequency

$$\frac{\partial \widetilde{OF}_{cP}}{\partial r} = 0 \tag{B.36a}$$

Stockout frequency

$$\frac{\partial \widetilde{A}_{cP}}{\partial r} = -\frac{\partial \widetilde{S}_{3cP}}{\partial r} \tag{B.37a}$$

$$= -\frac{p_{(S3b)}}{Q} \left[\Phi^0(z_{(r+0.5)}) - \Phi^0(z_{(r+0.5+Q)}) \right] \tag{B.37b}$$

Average new backorders

$$\frac{\partial \widetilde{P}_{cP}}{\partial r} = -\lambda \frac{\partial \widetilde{S}_{2cP}}{\partial r} \tag{B.38a}$$

$$= -\lambda \frac{p_{(S2b)}}{Q} \left[\Phi^0(z_{(r+0.5)}) - \Phi^0(z_{(r+0.5+Q)}) \right] \tag{B.38b}$$

B.3.3 Lagrange function

$$Z(r,Q,\zeta,\xi) = \sum_{j=1}^{J} f_j(r_j,Q_j) + \sum_{n=1}^{N} \zeta_n \left[\sum_{j=1}^{J} g_{anj}(r_j,Q_j) - e_{an} \right] + \sum_{o=1}^{O}\sum_{j=1}^{J} \xi_{oj} \left[g_{soj}(r,Q) - e_{io} \right] \tag{B.39}$$

$$\begin{aligned} Z(r,Q,\zeta,\xi) = & \left(\sum_{j=1}^{J} \Big[\alpha_{\overline{I}j}\widetilde{I}_{cPj} + \alpha_{\overline{OF}j}\widetilde{OF}_{cPj} + \alpha_{\overline{B}j}\widetilde{B}_{cPj} + \alpha_{\overline{A}j}\widetilde{A}_{cPj} + \alpha_{\overline{P}j}\widetilde{P}_{cPj} \right. \right. \\ & \left. \left. + \alpha_{S2j}\widetilde{S}_{2cPj} + \alpha_{S3j}\widetilde{S}_{3cPj} + \alpha_{SOLj}\widetilde{S}_{OLcPj} + \alpha_{rj}r + \alpha_{Qj}Q \Big] \right) \\ & + \sum_{n=1}^{N} \zeta_n \left(\sum_{j=1}^{J} \Big[\beta_{\overline{I}nj}\widetilde{I}_{cPnj} + \beta_{\overline{OF}nj}\widetilde{OF}_{cPnj} + \beta_{\overline{B}nj}\widetilde{B}_{cPnj} \right. \right. \\ & + \beta_{\overline{A}nj}\widetilde{A}_{cPnj} + \beta_{\overline{P}nj}\widetilde{P}_{cPnj} + \beta_{S2nj}\widetilde{S}_{2cPnj} + \beta_{S3nj}\widetilde{S}_{3cPnj} \\ & \left. \left. + \beta_{SOLnj}\widetilde{S}_{OLcPnj} + \beta_{rnj}r + \beta_{Qnj}Q \Big] - e_{an} \right) \\ & + \left(\sum_{o=1}^{O}\sum_{j=1}^{J} \xi_{oj} \Big[\gamma_{\overline{I}oj}\widetilde{I}_{cPoj} + \gamma_{\overline{OF}oj}\widetilde{OF}_{cPoj} + \gamma_{\overline{B}oj}\widetilde{B}_{cPoj} \right. \right. \\ & + \gamma_{\overline{A}oj}\widetilde{A}_{cPoj} + \gamma_{\overline{P}oj}\widetilde{P}_{cPoj} + \gamma_{S2oj}\widetilde{S}_{2cPoj} + \gamma_{S3oj}\widetilde{S}_{3cPoj} \\ & \left. \left. + \gamma_{SOLoj}\widetilde{S}_{OLcPoj} + \gamma_{roj}r + \gamma_{Qoj}Q - e_{io} \Big] \right) \end{aligned} \tag{B.40}$$

We introduce the θ abbreviation:

$$\theta_{Xj} = \alpha_{Xj} + \sum_{n=1}^{N} \beta_{Xnj}\zeta_n + \sum_{o=1}^{O} \gamma_{Xnj}\xi_{oj} \tag{B.41}$$

Now we can simplify the Z equation:

$$Z(r,Q,\zeta,\xi) = \sum_{j=1}^{J} \Big[\theta_{\overline{I}j}\widetilde{I}_{cPj} + \theta_{\overline{OF}j}\widetilde{OF}_{cPj} + \theta_{\overline{B}j}\widetilde{B}_{cPj} + \theta_{\overline{A}j}\widetilde{A}_{cPj} + \theta_{\overline{P}j}\widetilde{P}_{cPj} + \theta_{S2j}\widetilde{S}_{2cPj} + \theta_{S3j}\widetilde{S}_{3cPj} + \theta_{SOLj}\widetilde{S}_{OLcPj} + \theta_{rj}r + \theta_{Qj}Q\Big] - \sum_{n=1}^{N}\zeta_n e_{an} - \sum_{o=1}^{O}\xi_{oj}e_{io} \tag{B.42}$$

B.3.4 Lagrange derivatives

$$\frac{\partial Z}{\partial r_j} = 0, \quad j = 1,..,J \tag{B.43}$$

$$\frac{\partial Z}{\partial Q_j} = 0, \quad j = 1,..,J \tag{B.44}$$

$$\frac{\partial Z}{\partial \zeta_n} = 0, \quad n = 1,..,N \tag{B.45}$$

$$\frac{\partial Z}{\partial \xi_{oj}} = 0, \quad o = 1,..,O, \quad j = 1,..,J \tag{B.46}$$

We make use of the following abbreviations:

$$\Phi^0_r = \Phi^0(z_{(r+0.5)j}) \tag{B.47}$$

$$\Phi^0_{rQ} = \Phi^0(z_{(r+0.5+Q)j}) \tag{B.48}$$

$$\Phi^1_r = \Phi^1(z_{(r+0.5)j}) \tag{B.49}$$

$$\Phi^1_{rQ} = \Phi^1(z_{(r+0.5+Q)j}) \tag{B.50}$$

$$\Phi^2_r = \Phi^2(z_{(r+0.5)j}) \tag{B.51}$$

$$\Phi^2_{rQ} = \Phi^2(z_{(r+0.5+Q)j}) \tag{B.52}$$

B.3.4.1 Lagrange derivatives r

$$\frac{\partial Z}{\partial r_j} = \theta_{\overline{I}j}\frac{\partial \widetilde{I}_{cPj}}{\partial r_j} + \theta_{\overline{OF}j}\frac{\partial \widetilde{OF}_{cPj}}{\partial r_j} + \theta_{\overline{B}j}\frac{\partial \widetilde{B}_{cPj}}{\partial r_j} + \theta_{\overline{A}j}\frac{\partial \widetilde{A}_{cPj}}{\partial r_j} + \theta_{\overline{P}j}\frac{\partial \widetilde{P}_{cPj}}{\partial r_j} + \theta_{S2j}\frac{\partial \widetilde{S}_{2cPj}}{\partial r_j} + \theta_{S3j}\frac{\partial \widetilde{S}_{3cPj}}{\partial r_j} + \theta_{SOLj}\frac{\partial \widetilde{S}_{OLcPj}}{\partial r_j} + \theta_{rj} \tag{B.53}$$

$$\frac{\partial Z}{\partial r_j} = \theta_{\overline{I}j} p_{(\overline{I}d)j} \left[1 - \frac{\sigma_j}{Q_j}\left[\Phi_r^1 - \Phi_{rQ}^1\right]\right] + \theta_{\overline{OF}j} 0 + \theta_{\overline{B}j}(-p_{(\overline{B}d)j})\frac{\sigma_j}{Q_j}\left[\Phi_r^1 - \Phi_{rQ}^1\right] + \theta_{\overline{A}j}(-\frac{p_{(S3b)j}}{Q_j})\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{\overline{P}j}(-\lambda\frac{p_{(S2b)j}}{Q_j})\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{S2j}\frac{p_{(S2b)j}}{Q_j}\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{S3j}\frac{p_{(S3b)j}}{Q_j}\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{SOLj}\frac{p_{(SOLb)j}}{Q_j}\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{rj} \tag{B.54}$$

$$\frac{\partial Z}{\partial r_j} = -\left(\theta_{\overline{I}j} p_{(\overline{I}d)j} + \theta_{\overline{B}j} p_{(\overline{B}d)j}\right)\frac{\sigma_j}{Q_j}\left[\Phi_r^1 - \Phi_{rQ}^1\right] + \Big((\theta_{S2j} - \theta_{\overline{P}j}\lambda_j)p_{(S2b)j} + (\theta_{S3j} - \theta_{\overline{A}j})p_{(S3b)j} + \theta_{SOLj}p_{(SOLb)j}\Big)\frac{1}{Q_j}\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta_{rj} + \theta_{\overline{I}j}p_{(\overline{I}d)j} \tag{B.55}$$

$$\frac{\partial Z}{\partial r_j} = -\left(\theta_{\overline{I}j} p_{(\overline{I}d)j} + \theta_{\overline{B}j} p_{(\overline{B}d)j}\right)\sigma_j\left[\Phi_r^1 - \Phi_{rQ}^1\right] + \Big((\theta_{S2j} - \theta_{\overline{P}j}\lambda_j)p_{(S2b)j} + (\theta_{S3j} - \theta_{\overline{A}j})p_{(S3b)j} + \theta_{SOLj}p_{(SOLb)j}\Big)\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \left(\theta_{rj} + \theta_{\overline{I}j}p_{(\overline{I}d)j}\right)Q_j = 0 \tag{B.56}$$

$$\theta^1 = -\left(\theta_{\overline{I}j} p_{(\overline{I}d)j} + \theta_{\overline{B}j} p_{(\overline{B}d)j}\right)\sigma_j \tag{B.57}$$

$$\theta^2 = (\theta_{S2j} - \theta_{\overline{P}j}\lambda_j)p_{(S2b)j} + (\theta_{S3j} - \theta_{\overline{A}j})p_{(S3b)j} + \theta_{SOLj}p_{(SOLb)j} \tag{B.58}$$

$$\theta^3 = Q_j\left(\theta_{rj} + \theta_{\overline{I}j}p_{(\overline{I}d)j}\right) \tag{B.59}$$

$$\theta^1\left[\Phi_r^1 - \Phi_{rQ}^1\right] + \theta^2\left[\Phi_r^0 - \Phi_{rQ}^0\right] + \theta^3 = 0 \tag{B.60}$$

B.3.4.2 Lagrange derivatives Q

$$\frac{\partial Z}{\partial Q_j} = \theta_{\overline{I}j}\frac{\partial \widetilde{I}_{cPj}}{\partial Q_j} + \theta_{\overline{OF}j}\frac{\partial \widetilde{OF}_{cPj}}{\partial Q_j} + \theta_{\overline{B}j}\frac{\partial \widetilde{B}_{cPj}}{\partial Q_j} + \theta_{\overline{A}j}\frac{\partial \widetilde{A}_{cPj}}{\partial Q_j} + \theta_{\overline{P}j}\frac{\partial \widetilde{P}_{cPj}}{\partial Q_j} + \theta_{S2j}\frac{\partial \widetilde{S}_{2cPj}}{\partial Q_j} + \theta_{S3j}\frac{\partial \widetilde{S}_{3cPj}}{\partial Q_j} + \theta_{SOLj}\frac{\partial \widetilde{S}_{OLcPj}}{\partial Q_j} + \theta_{Qj} \tag{B.61}$$

$$\begin{aligned}\frac{\partial Z}{\partial Q_j} = \theta_{\overline{I}j}\Big(&\frac{p_{(\overline{I}d)j}}{2} \\ &- p_{(\overline{I}d)j}\frac{\sigma_j^2}{Q_j^2}\left[\Phi^2(z_{(r+0.5)j}) - \Phi^2(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^1(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{\overline{OF}j}\frac{-p_{(\overline{OF}b)j}}{Q_j^2} \\ &+ \theta_{\overline{B}j}\Big(-p_{(\overline{B}d)j}\frac{\sigma_j^2}{Q_j^2}\left[\Phi^2(z_{(r+0.5)j}) - \Phi^2(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^1(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{\overline{A}j}\Big(-p_{(S3b)j}\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{\overline{P}j}\Big(-\lambda p_{(S2b)j}\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{S2j}\Big(p_{(S2b)j}\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{S3j}\Big(p_{(S3b)j}\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) \\ &+ \theta_{SOLj}\Big(p_{(SOLb)}\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) \right. \\ &\qquad \left. - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right]\Big) + \theta_{Qj}\end{aligned} \tag{B.62}$$

$$\begin{aligned}\frac{\partial Z}{\partial Q_j} = &\frac{p_{(\overline{I}d)j}\theta_{\overline{I}j}}{2} - \left(\theta_{\overline{I}j}p_{(\overline{I}d)j} + \theta_{\overline{B}j}p_{(\overline{B}d)j}\right)\frac{\sigma_j^2}{Q_j^2}\left[\Phi^2(z_{(r+0.5)j}) - \Phi^2(z_{(r+0.5+Q)j}) \right. \\ &\left. - \frac{Q_j\Phi^1(z_{(r+0.5+Q)j})}{\sigma_j}\right] + \left(-\theta_{\overline{A}j}p_{(S3b)j} - \theta_{\overline{P}j}\lambda p_{(S2b)j} + \theta_{S2j}p_{(S2b)j}\right. \\ &\left. + \theta_{S3j}p_{(S3b)j} + \theta_{SOLj}p_{(SOLb)}\right)\frac{\sigma_j}{Q_j^2}\left[\Phi^1(z_{(r+0.5)j}) - \Phi^1(z_{(r+0.5+Q)j}) \right. \\ &\left. - \frac{Q_j\Phi^0(z_{(r+0.5+Q)j})}{\sigma_j}\right] + \theta_{Qj} + \theta_{\overline{OF}j}\frac{-p_{(\overline{OF}b)j}}{Q_j^2}\end{aligned} \tag{B.63}$$

$$\begin{aligned}\frac{\partial Z}{\partial Q_j} = &\frac{p_{(\overline{I}d)j}\theta_{\overline{I}j}}{2} - \left(\theta_{\overline{I}j}p_{(\overline{I}d)j} + \theta_{\overline{B}j}p_{(\overline{B}d)j}\right)\frac{\sigma_j^2}{Q_j^2}\left[\Phi_r^2 - \Phi_{rQ}^2 - \frac{Q_j\Phi_{rQ}^1}{\sigma_j}\right] \\ &+ \left((\theta_{S2j} - \theta_{\overline{P}j}\lambda_j)p_{(S2b)j} + (\theta_{S3j} - \theta_{\overline{A}j})p_{(S3b)j} + \theta_{SOLj}p_{(SOLb)j}\right)\frac{\sigma_j}{Q_j^2}\left[\Phi_r^1 \right. \\ &\left. - \Phi_{rQ}^1 - \frac{Q_j\Phi_{rQ}^0}{\sigma_j}\right] + \theta_{Qj} - \frac{\theta_{\overline{OF}j}p_{(\overline{OF}b)j}}{Q_j^2} \\ = &\,0\end{aligned} \tag{B.64}$$

$$Q_j^2\left(\frac{p_{(\overline{I}d)j}\theta_{\overline{I}j}}{2}+\theta_{Qj}\right)=\left(\theta_{\overline{I}j}p_{(\overline{I}d)j}+\theta_{\overline{B}j}p_{(\overline{B}d)j}\right)\sigma_j^2\left[\Phi_r^2-\Phi_{rQ}^2-\frac{Q_j\Phi_{rQ}^1}{\sigma_j}\right]$$
$$-\left((\theta_{S2j}-\theta_{\overline{P}j}\lambda_j)p_{(S2b)j}+(\theta_{S3j}-\theta_{\overline{A}j})p_{(S3b)j}\right.$$
$$\left.+\theta_{SOLj}p_{(SOLb)j}\right)\sigma_j\left[\Phi_r^1-\Phi_{rQ}^1-\frac{Q_j\Phi_{rQ}^0}{\sigma_j}\right]+\theta_{\overline{OF}j}p_{(\overline{OF}b)j} \tag{B.65}$$

$$Q_j^2=\left(\frac{2}{p_{(\overline{I}d)j}\theta_{\overline{I}j}+2\theta_{Qj}}\right)\left[\left(\theta_{\overline{I}j}p_{(\overline{I}d)j}+\theta_{\overline{B}j}p_{(\overline{B}d)j}\right)\sigma_j^2\left[\Phi_r^2-\Phi_{rQ}^2-\frac{Q_j\Phi_{rQ}^1}{\sigma_j}\right]\right.$$
$$-\left((\theta_{S2j}-\theta_{\overline{P}j}\lambda_j)p_{(S2b)j}+(\theta_{S3j}-\theta_{\overline{A}j})p_{(S3b)j}+\theta_{SOLj}p_{(SOLb)j}\right)\sigma_j\left[\Phi_r^1\right.$$
$$\left.\left.-\Phi_{rQ}^1-\frac{Q_j\Phi_{rQ}^0}{\sigma_j}\right]+\theta_{\overline{OF}j}p_{(\overline{OF}b)j}\right] \tag{B.66}$$

$$Q_j^2=\theta^4\left[\Phi_r^2-\Phi_{rQ}^2-\frac{Q_j\Phi_{rQ}^1}{\sigma_j}\right]+\theta^5\left[\Phi_r^1-\Phi_{rQ}^1-\frac{Q_j\Phi_{rQ}^0}{\sigma_j}\right]+\theta^6 \tag{B.67}$$

$$\theta^4=\frac{2\sigma_j^2\left(\theta_{\overline{I}j}p_{(\overline{I}d)j}+\theta_{\overline{B}j}p_{(\overline{B}d)j}\right)}{p_{(\overline{I}d)j}\theta_{\overline{I}j}+2\theta_{Qj}} \tag{B.68}$$

$$\theta^5=\frac{-2\sigma_j\left[(\theta_{S2j}-\theta_{\overline{P}j}\lambda_j)p_{(S2b)j}+(\theta_{S3j}-\theta_{\overline{A}j})p_{(S3b)j}+\theta_{SOLj}p_{(SOLb)j}\right]}{p_{(\overline{I}d)j}\theta_{\overline{I}j}+2\theta_{Qj}} \tag{B.69}$$

$$\theta^6=\frac{2\theta_{\overline{OF}j}p_{(\overline{OF}b)j}}{p_{(\overline{I}d)j}\theta_{\overline{I}j}+2\theta_{Qj}} \tag{B.70}$$

$$Q_j=\sqrt{\theta^4\left[\Phi_r^2-\Phi_{rQ}^2-\frac{Q_j\Phi_{rQ}^1}{\sigma_j}\right]+\theta^5\left[\Phi_r^1-\Phi_{rQ}^1-\frac{Q_j\Phi_{rQ}^0}{\sigma_j}\right]+\theta^6} \tag{B.71}$$

B.3.4.3 Lagrange derivatives aggregate constraints

$$\frac{\partial Z}{\partial\zeta_n}=\sum_{j=1}^{J}\left[\beta_{\overline{I}nj}\widetilde{I}_{cPnj}+\beta_{\overline{OF}nj}\widetilde{OF}_{cPnj}+\beta_{\overline{B}nj}\widetilde{B}_{cPnj}+\beta_{\overline{A}nj}\widetilde{A}_{cPnj}\right.$$
$$+\beta_{\overline{P}nj}\widetilde{P}_{cPnj}+\beta_{S2nj}\widetilde{S}_{2cPnj}+\beta_{S3nj}\widetilde{S}_{3cPnj}+\beta_{SOLnj}\widetilde{S}_{OLcPnj}$$
$$\left.+\beta_{rnj}r+\beta_{Qnj}Q\right]-e_{an} \tag{B.72}$$

Making use of (B.72) in step (B.73b) and next in step (B.73c) we replace ζ_n through a reformulation of (5.35).

$$\zeta_n = \frac{\sum_{j=1}^{J} \zeta_n \beta_{\overline{A}nj} \widetilde{A}_{cPnj}}{\sum_{j=1}^{J} \beta_{\overline{A}nj} \widetilde{A}_{cPnj}} \tag{B.73a}$$

$$= \sum_{j=1}^{J} \zeta_n \beta_{\overline{A}nj} \widetilde{A}_{cPnj} / e_{an} \tag{B.73b}$$

$$= \sum_{j=1}^{J} \left[\frac{-(\theta_j^1 \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right] + \theta_j^{2-\zeta} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right] + \theta_j^3)}{\theta_j^{2+\zeta} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right]} \beta_{\overline{A}nj} \widetilde{A}_{cPnj} \right] / e_{an} \tag{B.73c}$$

B.3.4.4 Lagrange derivatives individual constraints

$$\frac{\partial Z}{\partial \xi_{oj}} = \gamma_{\overline{I}oj} \widetilde{I}_{cPoj} + \gamma_{\overline{OF}oj} \widetilde{OF}_{cPoj} + \gamma_{\overline{B}oj} \widetilde{B}_{cPoj} + \gamma_{\overline{A}oj} \widetilde{A}_{cPoj} + \gamma_{\overline{P}oj} \widetilde{P}_{cPoj} + \gamma_{S2oj} \widetilde{S}_{2cPoj} + \gamma_{S3oj} \widetilde{S}_{3cPoj} + \gamma_{SOLoj} \widetilde{S}_{OLcPoj} + \gamma_{roj} r + \gamma_{Qoj} Q - e_{io} \tag{B.74}$$

$$\xi_{oj} = \frac{\xi_n}{1} \tag{B.75a}$$

$$= \frac{\xi_{oj} \gamma_{\overline{A}oj} \widetilde{A}_{cPoj}}{\gamma_{\overline{A}oj} \widetilde{A}_{cPoj}} \tag{B.75b}$$

$$= \xi_{oj} \gamma_{\overline{A}oj} \widetilde{A}_{cPoj} / e_{io} \tag{B.75c}$$

$$= \left[\frac{-(\theta_j^1 \left[\Phi_{rj}^1 - \Phi_{rQj}^1\right] + \theta_j^{2-\xi} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right] + \theta_j^3)}{\theta_j^{2+\xi} \left[\Phi_{rj}^0 - \Phi_{rQj}^0\right]} \gamma_{\overline{A}oj} \widetilde{A}_{cPoj} \right] / e_{io} \tag{B.75d}$$