How to Make the Most Productive Intervention in a Complex Economic System

Thesis

UNRESTRICTED

How to Make the Most Productive Intervention in a Complex Economic System

Bjorn Madsen

A thesis submitted in partial fulfilment for the degree of

Doctor of Philosophy

Of the Open University

March 2015

The Design Group,
Faculty of Maths, Computing and Technology

Date of Submission: 25 July 2014
Date of Award: 13 April 2015.

Abstract

Information about supply and demand propagates through supply chains in a queueing network with people and computers as batch information processors.

As each batch processor delays propagation of information whilst pursuing optimal local decisions, the effect is delay and distortion of the information that is used to commit resources to actions in the supply chain.

This thesis investigates the effect of delay and imperfect information as a *source of error*, to establish the case for change in research focus from optimal exploitation of physical constraints to optimal exploitation of information.

In the context of real world supply chains, the thesis asks "How does one make the most productive intervention in a complex economic system?" and pursues a meta-intervention which perpetually minimises the discovered error-term.

Evidence from literature indicates that agent-based modelling permits real-time peer-to-peer communication and distributed optimisation.

Based on the literature the research project designs and develops an agent-based model which operates in real-time without batch-processes and can perform incremental multi-objective optimisation under realistic (chronologically progressive) conditions for decision making.

The agent based model is then used to investigate two real-world supply chains, as case studies, which reveals a significant improvement of profitability and order-fulfilment.

The thesis concludes that agent-based modelling is a very promising direction for "making the most productive intervention" as it reduces delay to a minimum. Finally it recommends that continuous improvement of decision making methods is a role better suited for humans, rather than operational decision making where computers cope much better with the high amount of detailed information.

Acknowledgements

I gratefully acknowledge the contributions of the sponsors at LEGO System A/S who made this research project possible.

A special gratitude must go to Professor George Rzevski and Professor Petr Skobelev who have been my guiding stars navigating through new fields of knowledge, exploring ideas and for providing priceless feedback.

Thanks are also due to Professors Jeffrey Johnson, Anthony Lucas-Smith, for their support throughout the Ph. D.

The teams at LEGO, Barloworld Supply Chain Software, Open University, Smart Solutions & Multi-Agent Technology Ltd., and in particular: Kim Gynther Nielsen, Poul Jorgensen, Alexander Tsarev, Andrey Eremeev, Dmitry Ochkov, Jim Wilson, Frank Gesoff, Richard Forest, Jon Tucker, Francois Viljoen & Evrim Ovunc.

Without their contributions this Ph.D. would not have materialised.

The hallmark of science is not to predict,
but to explain how things work.

Herbert A. Simon

Table of Contents

List of figures

List of tables

Glossary & Acronyms

Agent – an abstracted representation of a person, organisation or object that can take action (Russell & Norvig 2009).

API – Application Programming Interface

APO – Advanced Planning & Optimisation

APS – Advanced Planning Systems

DC – Distribution Centre

DB – Database

ERP – Enterprise Resource Planning

FMCG – Fast Moving Consumer Goods.

FTL – Full Truck Load

IT – Information Technology

KPI – Key Performance Indicator.

LBR – LEGO Brand Retail, a division in the LEGO Group

MAS – Multi-Agent System

MRP – Manufacturing Requirements Planning

NSCM – New Supply Chain Model

OMT – Order Management team

OTIF – KPI for quantity delivered On-Time In Full.

PSO – Particle Swarm Optimiser

POS – Point of Sales data

RDN – Resource-Demand Network

SAP ECC 6.0 – SAP GmbH ERP used by LEGO System.

SC – Supply Chain

SCM – Supply Chain Management

SKU – Stock Keeping Unit.

S&OP – Sales & Operations Planning

Chapter 1 – Introduction

Overview – This chapter sets out the context for the dissertation: the motivation for its conception e.g. career in supply chain management and the motivation for pursuing the Doctorate. The logic (story line) and structure of the thesis is outlined, so the readers can anticipate what they will be reading about in the chapters that follow, and how they are linked.

1.1 Motivation

The motivation for this research is drawn from observations made in November 2007, where the author participated in an executive logistics meeting at LEGO Systems headquarters in Denmark.

The main concern was what that there was 4 weeks to the height of the toy industry's sales season, Christmas, and there was € 80million in retail value of stock in the distribution centre. The products would still be possible to sell in January and onwards, but the customers were calling customer service and complaining that their order fulfilment was less than 100%. In other words the customers were still ordering the products in stock, but somehow LEGO did not release the available stock to fulfil the customer orders.

The order management organisation was aware of the pending orders from reports from LEGOs enterprise resource planning (ERP) system. The warehouse staff was operating at ≈82% workload and could ship more if needed, but without release of the stock, the assignment of orders to stock could not happen, and without assignment, deliveries could not be created which the warehouse could act upon. Every part of the supply chain was trimmed to the highest efficiency and demand for the stock was visible, yet the information that was supposed to trigger the physical activities in the supply chain was not propagating.

LEGO Systems is well known across industries for having an excellent team of well qualified engineers and a generally high education level compared to other family owned businesses of similar size and maturity, so the obvious question of "why?" was approached with rapid systematic analysis of the processes which would reveal the bottleneck, so actions could be taken and results produced.

It turned out that the bottleneck was the process of decision making: In order to minimise the risk associated with taking unsold goods back from the retailers after the Christmas season, stocks were assigned to contracts whereby the business unit that said it was going to sell the stock, also assured that it would sell the stock. To cope with evolving expectations, a process had been agreed upon in which the markets negotiated with the global planning on how to assign the available stock. This decision process was democratic, interactive but ultimately not suitable for quick response to the market.

The dilemma occurred: Supply chains can be incredibly rapid in their response to changes in activities, but if the processes which transform the information about demand delay the ability for the supply chain to take action, the demand signal will become outdated and the quality of the decisions made in the SC degrade. On the other hand, if the ability to respond to changes in market demand, was built to suit the peak demand, relatively expensive production capacity would as a result be underutilised the rest of the year. Adjusting the capacity adaptive could be done, but not at notice shorter than a month. So there was the dilemma: How should the management model conclude what the right trade-off should be between all the constraints? The ERP system takes these constraints for granted and attempts to exploit them. The ERP system does not provide a model for simulation.

1.2 Research statement

Given the boundaries of the supply chain as a complex economic system, the research statement which this thesis explores is:

How does one make the most productive intervention in a complex economic system?

1.3 Contributions

This thesis makes the following contributions:

1. It provides a critique of existing approaches to supply chain optimisation, which:
    a. Departs from claims of optimisation when the information processing is batch based.
    b. Highlights problems with retrospective analysis, which does not recognise the chronologically progressive nature of decision making
    c. Makes the case for change in focus in the field of SCM from physical activities to information.
2. It recasts the supply chain coordination problem as an agent based model which is c-competitive and free of batch processing (referred to as the new supply chain model).
3. It provides a detailed specification of the new supply chain model, which is implemented by professional software engineers. The implementation redefines the role of the supply chain manager from decision making to continuous improvement of the decision making method, which is a novel form of meta-intervention in the field.
4. It delivers two case studies hosted by well-known global corporations, which illustrate significant results on key performance indicators of relevance to industry.

The reader should therefore – as a conclusion – anticipate that the most productive intervention in a complex economic system is a meta-intervention which deploys a set of strategies, which elevates the role of the human decision maker from "making decisions" to "improving the decision making method" and at the same time use computerised agent based models (ABM) for decision-making. The thesis is based on the evidence that the human decision making process currently is the most dominant obstacle for improved industrial productivity and therefore makes the case to engage people in the creative exploration and development of opportunities for the ABM to exploit when required. The symbiosis of raw rigorous information processing capability in software combined with human domain expertise establishes a context for a continuous improvement of decision making methods without the present days delay in information exchange and decision making which currently produces measurable losses to the modern business.

1.4 Structure

The thesis is structured as follows:

Chapter 1 sets the context out for the dissertation: The motivation for its conception (e.g. career in supply chain management and the motivation for pursuing a doctorate of the type presented), the logic (story line) and the structure of the thesis is presented so that the readers can anticipate what they will be reading about in the chapters that follow, and how they are linked.

Chapter 2 introduces the Supply Chain Management literature and develops the Research Question(s).

Chapter 3 comments on the theory of optimisation problems and describes the strengths and weaknesses of key mathematical approaches from John von Neumann to the present. It describes the SC problem associated with the challenge of solving the multi-echelon time variant knapsack problem and includes a discussion of how to compare methods based on "division and conquer", evolution based algorithms and message parsing systems. It highlights how faster computing and greater storage can allow for some progress but at a fundamental level, an obstacle for a major breakthrough, is the pursuit of local algorithmic performance and not system performance of the complex system as a whole.

Chapter 4 returns to the desire to tackle long-standing problems in supply chain management. It describes the research approach which is based on recasting the problems as an Agent Based Model and testing the model with case studies. It discusses the strengths and limitations of that approach, and explains why case study is an appropriate choice for this dissertation. It introduces the hypothesis that complexity-based approaches have something to offer and cites prior work motivating that hypothesis.

Chapter 5 describes the problems faced by LEGO before the research project started. It describes LEGO's (and the sector's) supply chain management approach in which "information is processed in a queueing network" and shows the consequence of this idea.

Chapter 6 performs a critical analysis of the situation described in previous chapters and shows the reasoning that leads to the formulation of a new approach to SCM that is focused on information and not physical logistics. This explains the issue of batch processing in a network of queues and the significance of the temporal dimension. It explains in detail the advantages of the new approach in respect to removal of obstacles for higher performance, such as batch processing including the significance of the time components of the human-computer interactive process.

Chapter 7 describes the implementation of the research ideas for a retailer: LEGO Brand Retail. The case study describes the simulations and how they were designed and implemented in detail. The results are presented, described and critiqued in detail and highlights the challenges of real-world implementation.

Chapter 8 describes the implementation of the research ideas for a fast moving consumer goods manufacturer, as a case study, which reinforces the core elements of the thesis as already introduced. This case study illustrates repeatability of the results in a much more complex supply chain.

Chapter 9 returns to consideration of batch processing of information in a queueing network. It discusses what can be learned from the three case studies. It gives a concise and detailed presentation of reflections on the computer-human aspects and concludes that the thinking concerned with system design in regard to computer speeds and human interventions are two parts of one problem.

Chapter 10 presents the main conclusions and recommend issues for further research. The contribution of the work to, primarily supply chain management, literature is summarised.

1.5 Publications

The following publications support the thesis. The author in bold is 1st author.

B. Madsen, P. Skobelev, G. Rzevski, and A. Tsarev, "Real-Time Multi-agent Forecasting and Replenishment Solution for LEGOs Branded Retail Outlets," in *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2012, pp. 451–456.

B. Madsen, G. Rzevski, P. Skobelev, and A. Tsarev, "A Strategy for Managing Complexity of the Global Market and Prototype Real-Time Scheduler for LEGO Supply Chain," *Int. J. Softw. Innov.*, vol. 1, no. 2, pp. 28–39, 2013.

B. Madsen, "Design & Deployment of an Enterprise Grade Real-time Multi Agent System for Supply Chain Synchronization," in *12th IEEE/ACIS International Conference on Computer and Information Science*, 2013, pp. 77–82.

Case study III: Real-time Manufacturing (FMCG) is to be published at Complex Systems, may 2015, as:
B. Madsen, "Complex Adaptive Software for FMCG", in WIT Transactions on Modelling and Simulation, vol. 58, 2015.

The full length literature review (appendix A.3 Literature review on supply chain management (extension)) is to be published in *Marik, V., et.al. "Adaptive Ramp Up for Manufacturing – ARUM", Springer, 2015* as Chapter 2. *"State-of-the art in Planning and Scheduling in Manufacturing"*.

Chapter 2 – Literature review: Supply Chain Management

Introduction – This chapter introduces the Supply Chain Management literature and develops the Research Question(s) with emphasis on optimisation methods to solve supply chain problems such as the knapsack problem.

2.1 Review question

During the past 15 years in the industry as supply chain consultant, I have observed that Supply Chain Management does not address effective communication, though it is much occupied with making logistics effective. Management attention is given to optimisation of the *activities* in the supply chain, instead of attending to the *information* which trigger the activities in the supply chain in the first place. The emphasis as it has been experienced, is, on efficient execution of activities (doing things right) in contrast to effective organisation of the activities (doing the right things).

This chapter reviews the literature with a perspective to clarify the topic of:

Are applications of optimisation methods within the domain of Supply Chain Management (SCM) grounded in the flow of information or are they focused on optimal exploitation of available information?

The term "Supply Chain Management" (SCM) is accredited to Keith Oliver in (Oliver & Weber 1982, pp.63–75) and referred to as

> *a supply chain is defined as a set of three or more entities (organizations or individuals) directly involved in the upstream and downstream flows of products, services, finances, and/or information from a source to a customer.* (Also cited in Mentzer et al. 2001, p.4; cited again recently in Ellram & Cooper 2014, pp.8–9).

Ellram & Cooper (2014) add that whilst "supply chain" is used as an umbrella term for more descriptive terms such as supply networks, demand chain and demand networks a stronger critique falls upon the term "management", where the authors note that *"despite the academic conundrum surrounding* [the exact definition of] *SCM* [,] *companies kept implementing SCM practices as they saw fit"* (Ellram & Cooper 2014, p.9) .

2.2 Timeline of SCM

As supply chain management is cross-disciplinary, the theme of supply chain management is informed by different perspectives which contribute to enrich the conclusions drawn from its pursuit of influence.

Year	Author	Contributing perspective
1982	Oliver	Supply Chain Management (SCM) formulated as cross functional process
1986	Smith	SCM as "Six Sigma" quality assurance program
1997	Vidal	Strategic production and distribution models as models for optimisation of SCM
1997	Cooper	SCM as interactive process of planning, execution and control of coordinated business processes.
1998	Lambert	SCM as Executive role
1998	Swaminathan	Technical difficulties with modelling supply chain dynamics
1999	Chen	SCM as unilateral negotiation process
2000	Fox	SCM as agency
2000	Angerhofer	SCM as dynamic system
2000	Christopher	The agile supply chain as competitive model in volatile markets
2001	Choi	The supply chain as complex system
2002	Ellram	The financial impact SCM
2003	Juttner	SCM and its influence on risk of disruption of supplies
2004	Christopher	SCM as risk mitigation process
2004	Goldratt	SCM as process of continuous improvement
2005	Christopher	SCM and difficulties with cost-accounting
2005	Christopher	Skills required for SCM
2005	Stadtler	How advanced planning can be used in SCM
2006	Gattorna	SCM as multi-dimensional business relationship
2006	Lau	Requirements for coordinated scheduling in a distributed system
2006	Wickers	Supply Chain Responsiveness as competitive advantage
2007	Huang	Supply Chain as platform for mass customisation
2007	Rzevski	Agile SCM as a call for revision of ERP & APS methods
2007	Pathak	Evolutionary dynamics of supply network topologies
2010	Smith	Modelling of resilience in supply chain
2010	Ivanov	Reconfiguration issues in so-called adaptive supply chains
2010	Allesina	The relevance of supply chain metrics for performance management
2013	Singh	How information propagates in the Supply Chain
2014	Ellram	30 years of SCM and still no theoretical foundation

Table 1 Timeline of contributions to supply chain management (SCM)

Several authors (Aitken 1998; Christopher 2005; Gattorna 2006) have attempted to rebrand "Supply Chain Management" as "Supply Network Management". Others, such as Choi et al. (2001, p.365) argue that it would be more informative "to recognize supply networks as a complex adaptive system (CAS)" and refer to "complex adaptive supply network" as "a collection of firms that seek to maximize their individual profit and livelihood by exchanging information, products, and services with one another". On the other hand, as economists use similar terms to Choi et al. (2001) for complex economic system (Blume & Durlauf 2005), the discussion can enter a blur of where the border may be between economics and SCM. This thesis takes the stance that Supply Chains are well recognised examples of Complex Economic Systems (Choi et al. 2001; Rzevski 2011) and though the three decade old title of a *chain* is less illustrative than *networks*, it has ingrained its meaning in the scientific community as application of methods which enable the delivery of a value proposition through a network of processes (Ellram & Cooper 2014). SCM also draws references to the field of Management Science, which Beer (1984) refers to as the "the business use of operations research". Organisation Theory is also used, for example with reference to Mintzberg (1983) with emphasis on the responsibility of coordinated decision making between the functions and departments. At the core of the SCM which Oliver & Weber (1982) defined, SCM as a field was justified through its cross functional role, however as reported by (Ellram & Cooper 2014) no unifying theoretical framework has been established.

A place where this is transparent is in SCM's usage of optimisation methods, which have been imported from management science and operations research. These methods have been developed with an explicit focus on a particular problem, such as:

I. network design,
II. transactions and the information system which supports the transactions, or
III. managerial function

Several authors discuss these perspectives, provide extensive critiques, but none appear to bind them into a unified framework which could form the foundation for theory development (Vidal & Goetschalckx 1997; Cooper et al. 1997; Lambert & Cooper 2000; Lambert et al. 1998; Angerhofer & Angelides 2000; Huang et al. 2003; Huang & Zhang 2007; Gunasekaran & Ngai 2004; Gunasekaran & Ngai 2005; Meixell & Gargeya 2005; Stadtler & Kilger 2005; Stadtler 2005; Burgess et al. 2006; Melo et al. 2009; Christopher 2001; Christopher 2005; Christopher & Rutherford 2004; Allesina et al. 2010; Oliveira & Gimeno 2014). These discussed perspectives are outlined in the three tables below.

Network design	1. The number, location, capacity and type of facilities (plants and warehouses) 2. The choice of sources of supply and demand, and contractual terms 3. Transportation modes, choice of routes and possible channels 4. Macroeconomic conditions (stability, security, transparency and trade culture)

Table 2 Modelling and optimisation perspectives involved in network design.

Transactions	1. Microeconomic decisions made by members of the chain of peers exchanging information (including information systems), which trigger logistic activities. 2. Organisation of business processes which balance supply with demand, such as planning and control of: 2.1. Procurement of raw materials, 2.2. Inventory, including coordination of production and shipping between facilities (routing), work-in-progress and finished goods. 2.3. Allocation of available stock to confirmed demand. 3. National interests such as customs declaration and operations associated with cross-border/cross-trade zone transactions (taxes, duties, exchange rates, trade barriers, transfer prices).

Table 3 Modelling and optimisation perspectives involved transaction processing

Managerial function	Strategic capabilities 1. Financial planning, such as evaluation of investments in 1st and 2nd tier suppliers 2. Comparative studies of alternative service model- and supply network-designs.
	Tactical allocations (connecting capacities to need for transactions) 1. Planning of material and resource requirements based on demand, bills-of-materials, lead time and available production and delivery methods. 2. Allocation of resources to committed portfolio of activities
	Operational continuity (completion of pending transactions) 1. Completion of order management, pick, pack & dispatch including administrative obligations 2. Preparation and complementation of production and maintenance 3. Resolution of conflicts caused by unexpected events.

Table 4 Modelling and optimisation perspectives (strategic, tactical and operational)

2.3 Critique of models used by SCM

Vidal & Goetschalckx's (1997) summarise their critique of model of supply chains and logistic systems as follows:

> "The [...] considerations allow us to claim that there exist many research opportunities for developing more comprehensive global supply chain models that include BOM constraints, more stochastic factors, and qualitative aspects that are very important within a global environment. Specific opportunities for research are the following:
>
> - explicit inclusion of more stochastic features in modeling international supply chains;
> - consideration of vendor and transportation channel reliability in the selection of vendors and transportation channels;
> - inclusion of customer service level as part of the set of constraints;
> - explicit modeling of potential economies of scale existing in interactional supply chains;
> - simulation of qualitative factors, such as the general infrastructure of a country;
> - differentiation of products by country;
> - determination of adequate excess capacity in different countries;
> - coordination of commodity flows, cash flow, and information flow within an international environment;
> - modeling of alliances and multi-company network configurations; and
> - development of specialized methods of solution." (Vidal & Goetschalckx 1997, pp.14–15)
>
> "it is easy to conclude that there exists a lack of features in the existing strategic models for the design of supply chains [...]. The main drawback of these models is the fact that most uncertainties are not considered in the formulations. In addition, there does not exist a formal and consistent way to represent BOM constraints. Moreover, some international factors, such as exchange rates, taxes, and duties are not fully described by the existing models." (Vidal & Goetschalckx 1997, p.15).

Stadtler & Kilger (2005) summarise this in in the preface of their third edition of *Supply Chain Management and Advanced Planning*:

> "The field of Supply Chain Management (SCM) and Advanced Planning has evolved tremendously since the first edition was published in 2000. SCM concepts have conquered industry – most industry firms appointed supply chain managers and are "managing their supply chain". Impressive improvements have resulted from the application of SCM concepts and the implementation of Advanced Planning Systems (APS). **However, in the last years many SCM projects and APS implementations failed or at least did not fully meet expectations**. Many firms are just "floating with the current" and are applying SCM concepts without considering all aspects and fully understanding the preconditions and consequences."

The methods used for transforming information into decisions are emphasised by their focus on some more or less explicit set of objective functions. Whilst the rigorous treatment of this subject is multi-objective optimisation (Coello 2006; Fu

2002), the more soft or inspirational is presented as leading the agents of the supply chain towards joint coordinated efforts of delivering the customer value proposition (Porter 2008; Christopher 2005; Gattorna 2006). Across the literature performance indicators are used to indicate the relative ability of agents to work towards the set objective functions.

2.4 Optimisation methods used in SCM

The dimensions of the objective functions typically belong to the classical MBA/M.Sc. SCM curriculum and range across customer experience, profit (revenue, costs) and the ability to execute at strategic, tactical and operational levels:

1. Factors which influence customer expectations:
   1.1. Brand expectations, reputation.
   1.2. Value-proposition means of product/service/image differentiation
2. Factors which influence revenue:
   2.1. Trade enabling factors, such as availability of service and efforts/cost of trade, i.e. being visible in the market.
   2.2. Order fulfilment rate: right time, right product, right location, right quantity and to right terms & conditions.
3. Factors which influence costs:
   3.1. Fixed and variable -production, -facility, -vendor/order, -transport and –production line costs; including costs associated with hedging, volume contracts and loans.
   3.2. Cost of capital from work-in-progress, inventory (pipeline-, cycle- and safety-stock) and excess inventory caused by lack of influence to coordinate/forecast demand, including lack of supplier reliability.
   3.3. Taxes, duties and other regulatory fees including licensing fees of IP-rights.
   3.4. Depreciation of obsolete and overdue products
   3.5. Government subsidies (cost reduction)
4. Factors which influence ability to execute at all levels:
   4.1. Human resources, talent
   4.2. Information systems
   4.3. Human/computer interaction

Table 5 Dimensions used to characterise objective functions of the supply chain

Badole et al. (2012) provide the latest and most extensive review of 690 papers[1] with detailed insight in the publications of papers which concern supply chain models, with focus on particular problems and the method used for solving the problem. Badole et al. (2012) find papers of supply chain models in 24 journals with 53.97% (of 302 papers) in the International Journal of Production Economics and the European Journal of Operational Research. The diversity of methods used is an extensive mix of 17 methods (Genetic algorithm, system dynamics, mathematics, linear programming, game theory, simulation, Taguchi methods, dynamic sequencing, fuzzy sets, mixed integer and linear programming, sensitivity analysis, Markov chains, petri net, agent based simulation, Lagrangian mechanics, ant colony optimisation, artificial neural network) for 3 key problem categories

[1] In (Badole et al. 2012, p.78) citations [59] and [64] are duplicates with errors in the authors title, so whilst the work covers a lot of papers, there is still opportunity for improvement of rigour.

(planning supply and demand, operational planning/scheduling and network design). The two tables below provides a summary of applied quantitative methods in literature.

#	**Method**	Supply & Demand Planning[2]	Scheduling[3]	Supply Network Design
1	Stochastic approximation[4]	2		
2	Ranking and selection	2		
3	Game Theory	4		1
4	Markov chain	3	2	
5	Petri net	1	4	1
6	Fuzzy Logic	3	3	2
7	Combinatorial optimisation	1	1	
8	Simulated annealing		3	
9	Dynamic Programming (divide and conquer)		2	2
10	Artificial Neural Network	1		1
11	Lagrangian relaxation	2		1
12	Mixed integer and linear programming	9	5	17
13	Monte Carlo simulation	1		1
14	Discrete event simulations (DES)	7	4	3
15	DES with system dynamics	1	6	1
16	Genetic algorithms	2	15	3
17	Tabu Search	1	1	
18	Particle Swarm optimisers		6	
19	Ant Colony Optimisers	1	2	1
20	Agent Based Models	1	43	1

Table 6 Methods and focus

[2] Including forecasting

[3] Including travelling salesman's problem and its derived routing problems

[4] Including iterative attempts to identify extrema which can only be estimated, not computed

#	Author
1	(Robbins & Monro 1951; Nemirovski et al. 2009)
2	(Runarsson & Yao 2000; Chan & Chung 2013; Giovannucci et al. 2007)
3	(Neumann et al. 1944; Huang & Zhang 2007; Shoham & Leyton-Brown 2008; Caro & Martinez-de-Albeniz 2010)
4	(Srivastava 2007; Shoham & Leyton-Brown 2008; De Boer & Boer 2000)
5	(Viswanadham & Raghaven 2000; Badole et al. 2012; Van der Aalst 1998; Biswas & Narahari 2004)
6	(Chan & Chung 2013; Bollen et al. 2010)
7	(Bidot et al. 2008; Giovannucci et al. 2007)
8	(Chan & Chung 2013; Kirkpartick et al. 1983; Iridia et al. 1997)
9	(Johnson 1954; Bellman 1986; Wu et al. 1999; Vidal & Goetschalckx 1997)
10	(Grljevic & Bosnjak 2011; Bollen et al. 2010; Astor & Adami 2000)
11	(Badole et al. 2012; Lidestam & Ronnqvist 2011)
12	(Badole et al. 2012; Shapiro 2007; Vidal & Goetschalckx 1997)
13	(Badole et al. 2012; Shapiro 2007; Vidal & Goetschalckx 1997)
14	(Chan & Chung 2013; Tako & Robinson 2011; Moon & Phatak 2005; Mönch et al. 2011; Shapiro 2007)
15	(Chan & Chung 2013; Angerhofer & Angelides 2000; Tako & Robinson 2011; Pathak et al. 2007)
14	(Power & Sharda 2007; Moon & Phatak 2005; Matuszek & Mleczko 2009; Klemmt et al. 2009; Mönch et al. 2011)
15	(Siebers et al. 2010)
16	(Konak et al. 2006; Horn et al. 1994; Coello 2000; Ghosh & Dehuri 2004; Poli et al. 2008; Slak et al. 2011; Chan & Chung 2013; Dimitrov & Baumann 2011)
17	(Badole et al. 2012; Chan & Chung 2013)
18	(Martinez & Coello 2011; Engelbrecht 2005; Zhang et al. 2011; Mohemmed et al. 2007; Fidanova 2005; Chan & Chung 2004; Silva et al. 2002)
19	(Meuleau & Dorigo 2002; Dorigo 1992; Ilie et al. 2010; Iridia et al. 1996; Bakhouya & Gaber 2007)
20	(Anosike & Zhang 2007; Max Gath, Stefan Edelkamp 2013; Siebers et al. 2010; Andreev et al. 2007; Madsen et al. 2012; Akanle & Zhang 2008; Allan 2009; Leitao & Vrba 2011; Chatfield et al. 2007; Ivanov et al. 2010; Leitão 2009; Turgay 2009; Zhang et al. 2006; Gath et al. 2013; Brintrup 2010; Skobelev 2011; Lau et al. 2006; Holmgren 2008; Chan & Chung 2013; Smith 2010; Neagu et al. 2006; Fox et al. 2000)

Table 7 Authors and methods. Some authors compare several methods.

From this overview it should be noted that the only method which may incorporate the rest is Agent Based Modelling (ABM). ABM is an established modelling paradigm in which software objects (agents) interact with their virtual environment to pursue a set of goals. The agent is thereby a software component with its on execution process and the environment is the swarm constituted by other agents. What distinguishes ABM from other modelling paradigms is the interaction, which probably is the single most important characteristic of complex adaptive software. Coordination is performed using exchange of messages with rich content. The only way to change the inner state of the agent, is through message exchange which the agent interprets single threaded. This stands in contrast to object oriented programming which does not recognise the encapsulation pursued in ABM, whereby synchronization locks are needed to avoid that two or more hyper-threads change the inner state of the object. The most characteristic difference is that agents in agent-based models suspend and continue operation autonomously (without prescription from the system designer). A modern multi-threaded computer thereby hands over the control between agents as if they were independent "lightweight" computing threads or micro- or nano-services (Wooldridge & Ciancarini 2001; Wooldridge 2009; Russell & Norvig 2009).

At the highest level of abstraction one can argue that ABM is a goal pursuing system which combines discrete event processing through message exchange amongst agents, with internal rules of achieving state updates[5]. The internal methods can thereby use all 20 methods, including ABM within or integrated with other ABM. Several authors therefore conclude that ABM is the way forward, with statements such as:

> "Agent technology has been recognized as a promising paradigm for next generation manufacturing systems." (Shen et al. 2006, p.415)

The mixture could indicate that the SCM community still is experimenting with methods. Evidence of this hypothesis is that, for example, that few authors are publishing papers on more than one problem solving method, and none publish for more than four methods, and that several authors spend sections to argue in order to obtain peer-acceptance of the notion of "optimality" when dealing with multi-objective optimisation problems, as optimality does not constitute the classical mathematical optimum (Coello 2006, p.29).

Melo et al. (2009) attempts to systematically explore what supply chain models have been focused on and discover that 75% of the literature is mainly focused on costs, compared to 9% multiple objectives and 16% on profit (Melo et al. 2009, p.408). Despite the critique of transparency of the models which were used, which was raised by Vidal & Goetschalckx in 1997, Melo et al. (2009) argue that a clear and specific algorithm can be traced in 75% of the articles when associated with facility location problems, though they declare that in "most of them the structure of the supply chain network is considerably simplified" (Melo et al. 2009, p.409). They add:

> "In addition to these findings, we note that the large majority of location models within SCM is mostly cost-oriented. This somewhat contradicts the fact that SCND[6] decisions involve large monetary sums and investments are usually evaluated based on their return rate."... "...Moreover, substantial investments lead to a period of time without profit. Companies may wish to invest under the constraint that a minimum return will be gradually achieved." ... "By considering profit-oriented objective functions, it also makes sense to understand, anticipate and react to customer behaviour in order to maximize profit or revenue. This means bringing revenue management ideas into strategic supply chain planning." Melo et al. (2009, p.410)

---

[5] This reference will be referred to later in the thesis.

[6] Supply Chain Network Design

So with evaluation of cost, and not profitability, the SC model will deliver a logically flawed advice to the business manager. This gives a clear guideline for future models: If they are not based on profitability, they will be wrong.

A second area where this has been discussed in great depth is in the critique of the accounting methods which provide the numerical values by which optimal decisions are determined[7]. This has been raised by several authors for the field of supply chain management:

- Theory of constraints (Goldratt & Cox 2004; Watson et al. 2007) provide a critique of producing widgets, without considering throughput.
- Lean Thinking (Womack 2008; Cunningham & Fiume 2003; Pepper & Spedding 2010; Christopher & Lee 2004; Christopher 2004) provides a critique of accounting practices which neglect explicit accounting of costs associated with activities which are not value adding to the customer. Examples of such "waste" refer to unnecessary transportation (i), inventory (ii), movements of units (iii), waiting or idle time of assets (iv), "overproduction" which is production of stock-keeping units (SKUs) which are not dedicated to an specific customer order (v) , "over processing" which is making objects of quality beyond the required quality standard (vi) and finally, defects (vii) which is production that results in scrapable SKUs.
- Six Sigma (invented by Bill Smith at Motorola in 1986, cited in Christopher & Rutherford 2004; Raisinghani et al. 2005; Pepper & Spedding 2010) which minimise wastage through process control where all process output is predictable with 6 sigma.
- Route maps to 4R's (Christopher 2005) which pursue SC-responsiveness(i), -reliability(ii), -resilience (iii) and trustworthy SC-relationships (iv) through a set of development projects which in combination results in a supply chain that is capable of providing a competitive advantage.

Often the meta-interventions apply methods from mathematics, similar to what functional specialisation has pursued, though with the more lateral focus that enables functional departments to collaborate more efficiently as the product of a process of continuous improvements instead of improvement through optimisation as a single discrete change. However as coupling of meta-interventions with simulation has not been observed in the literature, this domain is left unchartered. The inclusion of results of meta-interventions in optimisation models is, to some degree, implicit as the performance characteristics of process is used as inputs in optimisation models. The literature does not present any explicit examples of multiple persons collaborating on different parts of the supply chain model to clarify whether their actions are sub-optimising or resolving global bottlenecks:

> "While there is an abundance of SC management literature, it is realized that research at the inter-organizational level is less prevalent. However, the objective of SCM is to integrate all the firms in the value chain and treat them as a single entity (global supply chain). **Notwithstanding, the current research has failed to look at that perspective of the SCM.**" (Badole et al. 2012, p.75)

[7] For a detailed example of the accounting methods, please see the appendix on Accounting Principles.

This may be due to the practice that each researcher works either on a modelling problem (providing overview) or an optimisation problem (identifying optimality) in isolation, and thereby does not need to couple their SC-model interactively with other researchers. Practitioners echo this hypothesis, in their summaries: Fowler & Rose (2004) synthesise the key challenges for practical exploitation of the modelling and simulation methods as:

1. *An order of magnitude reduction in problem solving cycles*
2. *Development of real-time simulation-based problem solving capability*
3. *True Plug-and-Play Interoperability of Simulations and Supporting Software within a Specific Application Domain*
4. *Greater Acceptance of Modeling & Simulation within Industry*

And, Shen et al. (2006) state six requirements for what they call "next generation manufacturing systems" where they refer to systems used for practical exploitation of potential benefits for the supply chain as a whole:

> *R1. Full integration of heterogeneous software and hardware systems within an enterprise, a virtual enterprise, or across a supply chain;*
>
> *R2. Open system architecture to accommodate new subsystems (software or hardware) or dismantle existing subsystems "on the fly";*
>
> *R3. Efficient and effective communication and cooperation among departments within an enterprise and among enterprises;*
>
> *R4. Embodiment of human factors into manufacturing systems;*
>
> *R5. Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments;*
>
> *R6. Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the working environment.* (Shen et al. 2006, p.416)

These industry requirements contrast the relevance of academic publications which claims successful solutions to synthetic[8] problems. Shen et al. (2006) reflect on the research with self-criticism:

> *"Many researchers (particularly Ph.D. students) working on agent-based manufacturing are still focusing on the fundamental research to enhance the rationality or intelligence of software agents and develop more efficient and effective coordination and negotiation mechanisms. While this kind of research is important and still needed, we believe that the future R&D work should focus on the integration of agent-based planning and scheduling systems with existing systems used in manufacturing enterprises. The most important integration is with real time data collection systems, ... Another important integration is with existing ERP and MRP systems. Note that a certification is required for integrating or interfacing with some commercial ERP/MRP systems.* ***Only when such integrations are achieved and validated in industrial***

[8] I found it more appropriate to write "synthetic problems" than irrelevant problems

settings, will the agent technology be widely applied in manufacturing industry. (Shen et al. 2006, p.427)

2.5 Conclusions

This chapter provided a review of the supply chain management literature, in attempt to answer the review question:

> Are applications of optimisation methods within the domain of Supply Chain Management (SCM) grounded in the flow of information or are they focused on optimal exploitation of available information?

Supply Chain Management attempts to make the most productive intervention in the complex economic system it can exert influence on as a cross-disciplinary activity exploiting the best practices from operations research, management science, meta-interventions and its associated methods for incremental optimisation of "supply chain performance".

From a systems perspective the most productive intervention in a supply chain (as a complex economic system) ***can only be a meta-intervention*** characterised by:

(i) The information available to the decision maker,
(ii) The resources which may be mobilized at a particular point in time, and
(iii) The time it will take for the decision maker to transform the available information into a conclusion about what to do, and finally
(iv) When to communicate the made decision(s) to peers who need to incorporate the decision in their own plans.

From this perspective it is secondary which model that may be used to transform the information into a decision. Two methods which both can complete the transformation of the available information in the time between two events will compete on rigour, parsimony or precision, but only after the four characteristic terms (i-iv) have been considered. Yet of several algorithms identified, most implementations where based on batch-processing which disregards the incremental nature of updates to the information repository.

The answer to the review question is that the literature points towards that SC-models are "focused on optimal exploitation of available information". This happens despite the published critiques of SC-models which raises professional concerns about the (unjustified) reductions of SC-model to suit the optimisation libraries. It also raises concerns about myopic usage of available data, ignorant of absent data which could have informed the model. An example hereof was the extensive focus on cost-models instead of profit-models. Evidence drawn from meta-interventions also reveal that collaboration on SC-models is absent, whereby meta-interventions at best are limited to local interests, and not SC-wide collaboration. The review thereby informs the research objective by:

- Providing an overview of:
  - Optimisation perspectives (network, transactions, functions)
  - Optimisation focus (customer expectations, revenue, costs and ability to execute)
  - Optimisation methods used for SC-modelling (Table 6 Methods and focus).

- Proposing that the lack of ability to collaborate on SC-models may be the main obstacle for coupling models together to "make most productive intervention" across legal entities in order to improve decision making.
- Raising the critique of lack of ability to connect supply chain models with existing system, so that simulation and execution is combined as decision support.

Chapter 3 – Literature review: Optimisation

Introduction – This chapter comments on the theory of optimisation problems and describes the strengths and weaknesses of mathematical approaches from John von Neumann to the present. It describes the challenge of the knapsack problem and includes a discussion of "divide and conquer" algorithms and Dynamic Programming. It presents how faster computing and greater storage can allow for some progress but at a fundamental level, and barring a major breakthrough, algorithmic approaches will never succeed.

3.1 Origins of multi-objective optimisation

The origins of optimisation, is often attributed Leonid Kantorovich's methods (1939) which George Dantzig rephrased as the simplex algorithm for solving constraints based optimisation problems (1947) for the agricultural sector. However optimisation as an effective methods for detection of extrema in mathematical structures is described by Isaac Newton in Principia Mathematica (1687) where minimisation of the error term is known commonly as Newton's method[9]. Joseph Fourier also presents the principles of Fourier analysis in "Treatise on the propagation of heat in solid bodies" [10] to explain convergence. Gauss also presented a method in private correspondence in 1823 to Seidel who published the Gauss-Seidel method for successive displacement in matrix calculation in 1874.[11] More recently, John von Neumann's minimax theorem categorises optimal strategies for games, which is extended radically in Neumann et al. (1944). Single objective optimisation has now become a classical discipline, whilst multi-objective optimisation and verification of solutions in large solution landscapes remains a major research area (Coello 2006). Detection of the convex hull of solutions (Pareto efficiency) and comparison of multi-modal solutions also remain a growing research area. Determining the better strategy for effective identification of *the solution set* is central to analysis of computational complexity (Arora & Barak 2009) with roots in work from Turing (1936), Johnson (1954), Karp (1972) and Bellman (1986). These methods were designed so that one (and only one) mathematician could calculate the result given enough time was available. However as the digital computer allowed for acceleration of the computation by orders of magnitude, applications started to grow faster than the computer power available leading to pursuit of optimisation of the optimisation methods. Inspired by models of the human brain, models suitable for decentralised computation started to emerge (Neumann et al. 2000).

Johnson (1954) & Bellman (1986) both attempted to solve problems by dividing larger problems into their atomic parts, solve these, and aggregate the component solutions. Whilst this is suitable for some optimisation problems, the method is

---

[9] Cited in Cormen et al. (2009) but is also taught to grammar school students in Denmark in 5th grade.

[10] Mémoire sur la propagation de la chaleur dans les corps solides, présenté le 21 décembre 1807 à l'Institut national – Nouveau Bulletin des sciences par la Société philomatique de Paris I (6). Paris: Bernard. March 1808. pp. 112–116.

[11] P.278-281 in Carl Friedrich Gauss' 1903 *Werke*, published by Koniglichen Gesellschaft der wissenschaften, cited in http://www.encyclopediaofmath.org/ where the original source is available in German on http://gdz.sub.uni-goettingen.de.

only applicable for supply chain problems where the agent controls the supply chain with determinism. Where this condition does not apply, the optimisation method is of limited application, as its result may not be possible to execute in practice (Rzevski & Skobelev 2014). However, even though the method may have limited application to supply chain management, its explicit focus on solving problems which involved recursive functions inspired the development of the optimisation technique *memoization* which stores result of previous function calls, so that the results may be accessed using hash tables (Michie 1968; Cormen et al. 2009). This gave the benefit of a trade-off between run-time and memory, which later became essential for implementations of fly-weight patterns in agent based systems[12].

Conway (1963) speculated on the usage of co-routines which laid the foundations for message exchange – a method that later became fundamental for distributed optimisation methods. With message parsing, multi-agent negotiation, as described by Neumann et al. (1944), became possible, as iterative state-updates of the agents could be calculated asynchronously and thereby a negotiated compromise could be calculated using auctioning principles. A theoretical proof of this was provided by Nash (1950), but the absence of computing power, economic data and very outdated legacy assumptions about the physics of the financial market left the research in slow progress until Axtell et al. (1996) presented the first example of a supply network as a transactional economy in SugarScape. The method was inspired by John Horton Conway's cellular automata "Game of Life" and provided a milestone for growing acceptance of simulation as a research method.

The translation of centralised algorithms into distributed algorithms is not trivial, as race conditions in concurrent calculating threads need to be thoroughly considered to prevent indeterminacy race, which may produce erroneous results which are both hard to replicate and debug (Cormen et al. 2009). Burckhardt et al. (2011) only recently provided a novel revision of the principles of parallel computation, using self-adjusting concurrent revisions supported by memoization which outperforms centralized algorithms by more than the number of computing cores. An alternative approach to translation of existing algorithms into parallel methods, is to decouple the computation using auctions. Bertsekas (1979) provided an early example where auction principles are used for solving the assignment problem, which over the following three decades turned out to be the most efficient method for conflict resolution in agent-based systems (Rzevski & Skobelev 2014; Tesfatsion 2006)[13]. Experiments with other iterative methods for extrema discovery in solution landscapes, such as genetic algorithms, particle swarm optimisation and simulated annealing have also been informative. The table below provides a guiding overview over the past century's development:

[12] (Iba et al. 2002) deploy a range of object oriented pattern covering: Factory, Factory Method, Singleton, Adapter, Composite, Flyweight, Command, Iterator, Observer, State, Strategy, Template Method and Visitor pattern. See §4.2 in (Iba et al. 2002, p.66)

[13] The uninitiated reader may find a suitable guide in Parsons & Klein (2011)

Decade	Method	Characteristics
≤1950s	Linear Programming Integer Programming Mixed Integer Linear Programming	Batch-processing methods
1960s	Monte Carlo Simulation	Probabilistic methods
1970s	Genetic Algorithms	Probabilistic tournament
	Computer Auctions	Iterative convergence
1980s	Particle Swam Optimisers	Concurrent probabilistic tournaments
1990s	Multi-Agent Systems	Negotiations with iterative convergence
2000s	Genetic Programming	Probabilistic algorithmic tournament
2010s	Networks of Multi-Agent System	Distributed negotiations with iterative convergence

Table 8 Optimisation approaches through the decades

These methods are based on the same assumptions associated with how optimality is defined. First, most implementations of the algorithms are batch based, meaning that each computation cannot start unless the dataset required is complete. For the implementations of algorithms which do allow for updates during the computation, practice in industry rarely deploys them with this advantage in mind. One could argue that this is because computer science education only recently has increased focus on asynchronous computation and that there therefore is a certain delay in adoption of these practices.

The critique of SC-optimisation models presented in the previous chapter was that researchers have pursued to reduce the problem to suit the optimisation library. The critique was directed towards assumptions concerned with supply chain optimisation models where the agent had dictatorial control over the supply chain as a whole. The criticised group typically represented the supply chain optimisation problem as ***a centralized multi-echelon time-invariant knapsack problem*** (Kogan & Tapiero 2007) in which the customer side of the network stated demands, which were propagated towards the supplying side based on rational market conditions (Shapiro 2007; Gattorna 2006; Christopher 2005). This line of thought dominates the design of industrial enterprise resource planning (ERP) systems which apply naïve propagation of demand, which is imposed upon the supplier (Snapp 2009; Dickersbach 2009; Stadtler & Kilger 2005). However, practice has revealed that suppliers cannot always fulfil the ordered quantity which adds a feedback loop. This feedback loop is rarely made explicit in supply chain models, but must be considered in order for the optimisation methods to be operational. Two problems thereby needs additional focus:

- How can the centralized METVKP can be constructed as a distributed METKVP?
- How can the feedback loop be included in the optimisation process?

3.2 Multi-echelon time-invariant knapsack problem

Transformation of any centralised Multi-Echelon Time-Invariant Knapsack Problem (METVKP) to a distributed METVKP is equivalent to Johnson (1954) and Bellman (1986)'s divide and conquer method under the assumption that any local problem in the distributed METVKP considers any remote problem as a constraint. A local optimum in a distributed METVKP will thereby be the same if and only that local optimum also is a global optimum. Likewise any update to the constraints of a local problem will thereby lead to subsequent propagation to the connected problems. A simple logical evaluation is that if the connected problems are of insignificant influence from a global optimisation perspective – in comparison to the focal problem – then solving the local problem is the primary task. When comparing the models a set of trends emerge: The divide and conquer methods first cuts the parent problem (1), divides the problem into child problems (2) and subsequently evaluates the solution of these (3) before aggregating. By concept this is not much different than if one considers to solve the centralised METVKP using a genetic algorithm, as described by Coello (2000), where the algorithm mutates (1) using a cut in the genetic string and performs a cross-over of the settings, creates child solutions (2) as sub-solutions and evaluates the child solutions to the environment (3). The pattern of "cut-solve-evaluate" is repeated at a higher level of abstraction. The agent-based method embeds this approach in the message exchange, with cuts (1) defined by each swarm of agents, sub problems as the inner state of the swarm (2) and evaluation of the result (3) as the message is exchanged with the swarms environment (Rzevski & Skobelev 2014, pp.35–48).

Step	Divide & Conquer	Genetic Algorithm	Agent Based
Cut	Cut problem	Cut Gene-sequence to mutate	Cut to limit swarm
Solve	Solve child problem	Generate child solutions	Solve within swarm
Evaluate	Evaluate child solution	Evaluate child solutions	Send message to environment to evaluate

Table 9 (Very) high level comparison of optimisation ideas

The notable differences are that the divide and conquer method is strictly progressive chronological from the top of the recursive tree to the lowest sub problems; whilst the genetic algorithm is initiated randomly in the solution landscape and converge through evolution, and whilst the agent based method is initiated at the particular disruptive event. Any change to constraints in the divide & conquer method thereby requires either renewed top-down batch calculation or backtracking within the tree. Any change to constraints in the genetic algorithm could be compensated for using repeated mutation, which logically is more efficient that complete reset/restart of the calculation, but contains the risk of genetic drift towards a local optimum, where the method may get stuck unless the gene pool is actively disrupted. Any change to the agent based method will propagate from the update (disruptive event) and only perform changes where conditions are identified which are not Pareto efficient. As each agent can report this individually, Pareto efficiency may – in popular terms – be characterised by whether the agent is satisfied with the solution or not. The table below summarises these characteristics:

	Divide & Conquer	Genetic Algorithm	Agent Based
Respond to changes	Restart or backtrack	Continue mutation	Propagate update
Worst case elapse	Backtrack to top of tree twice as long as restart	Drift into local optima	Complete re-computation twice as long as restart

Table 10 Comparison of effect of changes to a computed solution (Time variance)

Evidence of methods constructing divide and conquer trees using genetic algorithms instead of backtracking have shown some interesting properties in terms of scalability, but does otherwise not appear more effective than the ontology guided message parsing applied by Rzevski & Skobelev (2014).

A particularly interesting method is the iterative auction developed by Bertsekas over the period from 1979 to 1992 which uses message parsing in a resource-demand network (RDNs). The main challenge in was not the distributed nature of the problem, but rather the time-variance. Bertsekas (1979) presentation as ***iterative calculation of maximum flow in a bipartite graph*** can handle this problem elegantly when implemented with modern implementations of Conway's (1963) co-routines, as message about updates may be included in every sub problem. Most notably was Bertsekas & Castañon (1991) and Bertsekas (1992) augmentation of their original approach using alternating auctions:

```
For each iteration
  alternate auction direction
    if resource:
      bidding price = resource reservation price
      for each demand:
        if bid ≥ offer:
          bidding price = bid
        else:
          pass
    elif demand:
      purchase reservation = demand maximum price
      for each resource:
        if offer ≤ lowest_bid:
          purchase reservation = lowest_bid
        else:
          pass
```

Figure 1 Outline of alternating auctions to solve assignment problem as max-flow in bipartite graphs.

Rzevski & Skobelev (2007) developed a method using alternating bid-requests in each bipartite set, like Bertsekas & Castañon (1991) but with asynchronous parsing of message using a queue. The first message in the queue initiates the alternation sequence, whereby each set in the bi-partite matching generates a set of messages (bids and offers). However to deal with the time-variance any set in the bipartite matching can be updated and simply add the message to the queue. As the only change is the inclusion of the queue, the alternating sequence – as proven by Bertsekas (1979), Bertsekas & Castañon (1991) and Bertsekas (1992) and honoured by the IEEE – will converge towards optimality at most $\theta(mn)$ so the method is theoretically more efficient than divide and conquer with $\theta((m+n)^2)$.

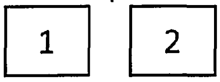
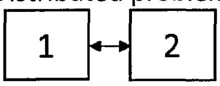
Model	Characteristics
Centralised problem 	The centralised optimisation problem is given by maximizing a multi-objective function f, so that $\max f(x_{global})$
Divided problem 	The divided optimisation problem is given by isolation of optimisation vectors from f so that $f_1, f_2, \dots, f_n$ collectively represent the constraints. The isolation of sub problems dismisses any guarantee of identification of a global optimum unless the sub problems are independent.
Distributed problem 	The distributed optimisation problem is given by the graph G which represents the properties of f, such that nodes $n_1, n_2, \dots, n_n$ represent the functions $f_1, f_2, \dots, f_n$ which collectively represent the constraints. By using message exchange between connected the nodes a compensation can be passed such that the local penalty $p_{local} = max(f_{local}) - f_{local}$ may be offset by a compensation $(f_2(x_2) \rightarrow f_1\ and\ f_1(x_1) \rightarrow f_2)$ if the compensation is less than the total benefit. To avoid that the message are exchanged in an infinitely recursive graph, each message is associated with a transaction cost, whereby the local node forfeits the right to produce new messages until all remote queries have been responded to.

Table 11 Methods for representing the supply chain problem as distributed problem.

A subject discussed mainly in the field of artificial intelligence is *learning*: As the model representing the supply chain problem is based on real world physical constraints, it is reasonable to assume that everything cannot change at any instance. Thereby the model representing the real world should contain the property of gradual change, whereby learning can be an effective strategy (Russell & Norvig 2009). An example hereof is a change in the supply network with a revision of the supply network architecture. The challenge for the divide and conquer method is that any change may require a complete reformulation of the optimisation tree, as some branches may disappear or grow radically. Genetic algorithms will have to restart computation as the solution landscape also may change radically. The agent-based method can add/remove the class representing the change and propagate the changes anew. The agent-based approach is thereby simpler to modify as it only requires a change in topology of the ontology, in comparison to divide and conquer and genetic algorithms which require both abstraction and reformulation of the objective functions.

Effective learning strategies in agent-based systems can thereby be incremental adjustment of gradient of pay-off/fitness within the context of a given problem, and through a reset of the gradient whenever the ontology is updated.

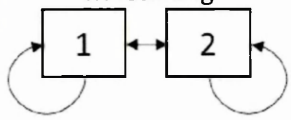
1-state distr. problem w. learning 	The single state distributed optimisation problem is given by distributed problem, though with the notion that each node evaluates the gradient of offsets, so that the highest growth of objective function is pursued. This property reduces the number of messages exchanged.

Figure 2 Single-state distributed representation of the problem with learning

Finally, commitment of resources may be required at any time, whereby it must be possible to delimit commitment of any single decision to a bi-partite subset within the graph given by the topology of the agent-based system with all its relations between resources and demands. This can be done at the cost of double memory and a constant runtime penalty, using memoization where a 2-state distributed representation of the optimisation problem is maintained (even under conditions of learning) such that the memoization always has a suitable solution available (see below).

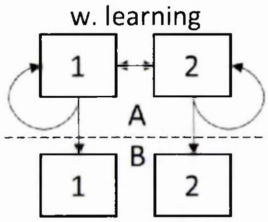
2-state distr. problem w. learning 	The 2-state distributed optimisation problem is given by the 1-state with the additional feature that the best known solution ***B*** is substitute with ***A*** whenever a better solution is identified. Hereby any temporary state that solution A might enter which is worse than the best known solution stored in B, does not compromise the ability to commit resources, as the worst case penalty ***B-A*** is given by excluding the latest event from being considered. If state A no longer can identify incremental improvements, the solution may reset local functions, so that it escapes suboptimal solutions in attempt to explore as much of the solution landscape as possible.

Figure 3 Dual-state distributed representation of the problem with learning and possibility for immediate commitment

To summarize this literature review on optimisation methods, the reader must remember the following contributions:

The divide and conquer model from Johnson (1954), can be presented as a Boolean tree which has been done by several authors in the 1980's. Kauffman developed a framework (1993) for representing the fitness functions as a gene, which Johnson represented earlier as a Boolean tree. Independently from Johnson (1954), Bertsekas (1979) solved the assignment problem using maximum flow in a bi-partite graph with the sets X and Y. Bertsekas showed that this method was guaranteed to converge in $O(mn)$ where in contrast Johnson's method will require $O(m+n)^2$. In 1991-1992 Bertsekas extended his proof. Bertsekas representation of the bi-partite graph is similar to Rzevski & Skobelev (2014)'s "swarm", if – and only if - the swarm has a single resource-demand network(RDN). Under these circumstances, the resource- and demand-side of the RDN are identical to, respectively, Bertsekas bipartite sets Y and X. The consequence hereof is that Rzevski & Skobelev (2014)'s method therefore also should also run at worst in $O(mn)$ as the principles are the same. In 2000 Coello gave a presentation of a genetic algorithms where each retained mutation can be considered as a message from the environment about which genotypes are fit for purpose. However in contrast to Coello's (2000) random mutations, Rzevski & Skobelev (2014)'s ontology acts as routing topology for the fitness response messages whereby the MAS only initiates changes (mutations) to its bi-partite graph (resource demand networks) when it receives feedback. As the mutation runs $O(mn)$ but with a local focus the problem as such is smaller than in generic genetic algorithms which mutate the whole gene pool. The efficiency of a MAS must thereby be higher, as it otherwise has the same algorithmic complexity ($O(mn)$ ref. Bertsekas 1992) but is executed on a smaller problem. For comparison, even if the problem is represented as a gene which is systematically modified using Johnson's method (1954) the fitness evaluation will not happen until the whole gene modification process is completed. Obviously this results in much slower feedback in a distributed system.

Using Kauffman (1993) as reference, Rzevski & Skobelev's (2014) multi-agent system could be classified as an auto-morphic gene network, which adapts through internal reconfiguration – and not in generations. Kauffman argues that the immune system works in similar manner, using proteins as messages. Whilst this is far beyond the scope of this thesis, the analogy appears suitable as a source for further inspiration and research.

3.3 Feedback loop

The second challenge is to consider the feedback loop to determine when the computation is complete. The logic behind this question is simple: What if an agent in a distributed system sends a message to which there is no response before the agent is required to commit resources? Is it legitimate to presume that the request will be fulfilled? There are three options:

(A) Time limit – If the computation is considered complete because the agent has run out of time, then this is an amendment to the halting conditions of the algorithms. The general mathematical definition of allocation of resources to demand as a sequence of events distributed in time is formally a scheduling problem:

> "[s]cheduling is concerned with the allocation of scarce resources to activities with the objective of optimizing one or more performance measures." (Leung 2004, p.19)

where complete information about the scheduling problem is required:

> "in all of the scheduling problems [...,] the number of jobs (n) and machines (m) are assumed to be finite." (Leung 2004, p.24)

Without the feedback at the time limit, the scheduling problem is incomplete and can therefore not be computed according to Leung.

(B) Complete Information – Is the computation complete because the answer must be based on available information, and thereby does not require a response? From a purely mathematical point of view the variables can be treated as a single variable, and – likewise – an agent can act as a proxy for a given organisation of agents, without any reasonable objections can be made. This treatment allows us to construct models which include actual human response instead of assumptions of human behaviour (illustrated below).

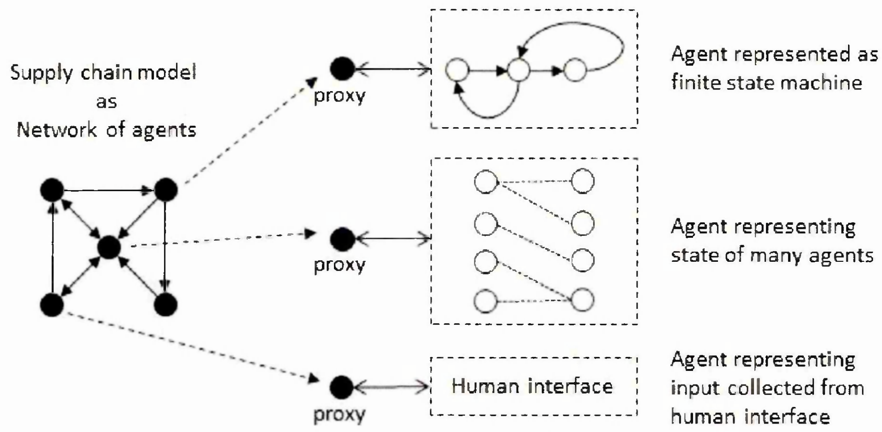


Figure 4 asynchronous communicating agents in a heterogenous agent environment

Whilst this may represent the reality of the information exchange in the supply chain, it is not made explicit in the treatments of assumptions in the literature, what shall be done if responses are pending.

(C) Introspection – Should a game theoretic position be taken which permits assumptions of response until the prediction is substitute with facts? Such a game theoretic perspective would apply expected utility to guide the choice of the agent. In Neumann et al. (1944)'s N-player Robinson-Crusoe economy the optimisation problem is phrased as a strategy choice where all players perform self-interested collaboration:

> Sometimes free competition is assumed, after the introduction of which the participants face fixed conditions and act like a number of Robinson Crusoes-solely bent on maximizing their individual satisfactions, which under these conditions are again independent. In other cases other restricting devices are used, all of which amount to excluding the free play of "coalitions" formed by any or all types of participants. (Neumann et al. 1944, p.15).

The three options make the representation of the supply chain optimisation problem as a mathematical topic difficult: As long as all members of the supply network perform self-interested collaboration, which (i) maximizes revenue and (ii) minimisation of costs-to-serve and tactically neglect order fulfilment on part orders which are not profitable, one could argue that the supply chain is extracting the maximal profit from the served customers. The suppliers are then implicitly required to cost-engineer the value-propositions which are in demand, but not profitable, to increase their competitiveness (Kauffman 1995). This process indicates how innovation in technology may influence the productivity of the supply chain beyond the edge of what is computable: The demand and the sales price is possible to know, but cost-to-serve will depend upon processes which have not yet been invented. The benefit however by having an optimisation model that is easily extendible is that it allows the decision-maker to make projections of any future state of the system without having to commit the investment a physical experiment would require, and only at the risk that the model may be flawed. Several authors have described these which are a suitable for the design of multi agent systems starting with Neumann et al. (1944) and most rigorously treated in Shoham (1993); Shoham & Leyton-Brown (2008) though axioms probably are described most explicitly in Kogan & Tapiero (2007); Kleinberg & Easley (2010);

Chaib-draa & Muller (2010). For a coherent introduction on how to deploy the axioms as "rules of behaviour" see Rzevski & Skobelev (2014). Neumann writes:

> "A choice of axioms is not a purely objective tasks. It is usually expected to achieve some definite aim – some specific theorem or theorems are to be derivable from the axioms – and to this extent the problem is exact and objective. But beyond this there are always other important desiderata of a less exact nature: The axioms should not be too numerous, their system is to be as simple and transparent as possible, and each axiom should have an immediate intuitive meaning by which its appropriateness may be judged directly." ... "To strike a proper balance is a matter of practical – and to some extent even esthetic – judgment." Neumann et al. (1944, p.25, §3.5.2)

Across these three options (a) time constraints, (b) available information and (c) game theoretical predicates, Ockham's Razor creates a divide as it eschews the assumptions of optimisation in both favour and disadvantage to either of the approaches, whereby we are forced to tolerate realism at the compromise of proof of solution with falsification of assumptions of the models which we choose to represent reality (Popper 2002).

Common for all three options the model design which implies *batch processing* of information[14] will return a worse result in comparison to asynchronous updates, as the queue being populated for batch-processing *per design* will delay the information from propagating between the agents in any distributed system. Systems which are driven by clocks (Laplante & Ovaska 2011; Kopetz 2011), and therefore not event driven, will also embed a similar delay in propagation of information and will result in a similar error/penalty. It must therefore be noted that the source of delay in the process of transforming information into decision is a product of the interactions of humans and algorithms which collectively create "a queueing network of batch-processors". Furthermore as the current software development practices the usage of batch processing without memoization and computation at write-time, the computational process will contain more redundant iterations, and will thereby be more likely to run out of time. The supply chain with the fastest computers and algorithms will not resolve this problem, if humans still are engaged directly in the decision making of how resources are allocated to demands. In the absence of facts, a game theoretic proposition based on competitors and partner's observable behaviour, constructed by humans and executed through the usage of software agents as proxy for the human decision maker may be the most feasible solution as it combines the best of all considerations. Unfortunately none of the literature presented a model which takes the game theoretic predicate at its foundation[15]. No rigorous stance towards assumptions for all data analysis is given, whereby the publications remain implicitly naïve about the subject of whether the agents of the supply chain remain loyally collaborative or may have more incentive chose to otherwise. From dialogue with doctoral students, associate professors and professors the general understanding of game-theory as a fundamental evaluation of options appear

[14] Batch processing includes processing of information which queues at least 1 message, in contrast to asynchronous process which processes each message without delay.

[15] From dialogue with doctoral students, associate - and professors, I have quietly developed the impression that the misunderstanding of game theoretical foundations remains my deepest disappointment throughout this research project.

deeply misunderstood. An example hereof is that studies of time series presume that the parties in the system continues to behave the same way. However when everyone behaves the same way the incentive to defect increases, whereby the fundamental assumptions of time-series analysis are no longer valid and only the game-theoretical foundations of the strategic choice remains. As a researcher, this is cause for deep concerns, as it means that Supply Chains are focused on exploitation and risk management in the physical logistics and generally ignorant of the source of origin of the information. An advice for future research in supply chain system design is thereby to include explicit game theoretical predicates before focusing on sub-optimisation of given performance metrics. A very simple example of this is to include the consideration of whether the supplier is unprofitable, as such a case gives incentive to reduce order fulfilment on non-profitable items, whereby the focal supply chain may not be able to fulfil its own customer orders. The consequence of systematic pursuit of myopic performance metrics can lead to temporary benefits, but also leads to monopolistic consolidation in – primarily – supply and purchasing power. Several auction models deals effectively with these cases (Parsons & Klein 2011), but none have been presented in the literature on optimisation methods.

In conclusion, it is implied that a ranking exists in handling the challenges associated with the feedback loop:

1. If no feedback is available at all, the game theoretical predicate (based on observed behaviour and incentive) takes precedence.
2. If information is available but incomplete, a clear distinction of what is factual and what is estimated must be made to take necessary actions to achieve pending objectives without overcommitting to objectives which cannot be validated.
3. Finally the time limit takes precedence to systematically depart from taking actions to which no incentive or estimate indicates a necessity.

Unfortunately none of the literature provides explicit consideration of this hierarchy, nor, in general, considers the feedback loop at all. Badole et al. (2012), for example, raises attention towards the absence of interactively negotiated compromise between optimisation methods. This is an important critique as information exchanged by systems between businesses is a naïve propagation of demand (see Figure 5, below) whereby no feedback loop exists (from purchasing and MRP).

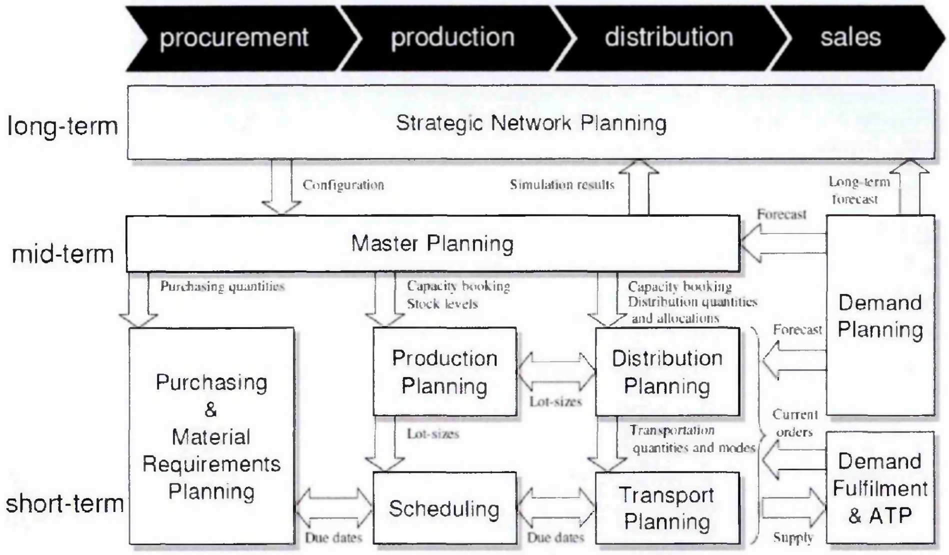


Fig. 13.1. Coordination and data flows of APS modules

Figure 5 From Stadtler & Kilger (2005, p.246) *Illustration of data flow amongst modules.*

The attentive reader will also notice the absence of the human decision maker. The literature implicitly treats the human-computer interactive element as instantaneous, and thereby ignores the delay from information is available with a human decision maker until it is released upstream in the information flow. For practice this means that optimisation methods often are deployed such that each business unit will perform sub optimisation of what it expects its suppliers to do disregarding what the supplier might be capable of delivering (Snapp 2009). Two practices are prevalent for humans (Martin 2013):

a) They wait for a scheduled deadline to pass, then download data from an enterprise system, transform it and either upload the results to the enterprise system or send it to somebody else.
b) They wait for emails with data or links to data, transform it into decisions in spreadsheets, and forward it to the decision maker or upload it to an enterprise system

Two practices are prevalent for enterprise systems (Snapp 2009; Stadtler & Kilger 2005):

a) They await a scheduled deadline to pass, load all data into memory from a database, transform it using predetermined algorithms and send a signal to users and other systems that the scheduled job has been completed/failed.
b) They wait for another user/system to signal that the required data is available, process it, and send a signal to users and other systems that the scheduled job has been completed/failed.

As each decision making entity first awaits access to complete information for a given time horizon – instead of incremental asynchronous information processing – it penalises the ability for upstream decision makers to coordinate and react by the endured delay. In addition the "optimal solution" for the one entity also constraints the solution landscape to upstream entities as the absence of the feedback loop prohibits them from suggesting alternative options for the many cases where the naively propagated demand cannot be fulfilled.

Therefore no matter how fast the local optimisation is – or the human decision maker for that sake – as long as the queueing network of information processors use batch-processing and not asynchronous incremental information processing, each node in the supply network will continue to sub-optimise by attempting to solve the multi-echelon time-variant knapsack problems at the cost of system-wide coordination.

3.4 Summary and conclusion

The review of the assumptions about how optimality is identified was divided into three sections, which elaborated on:

- The origins and evolution of multi-objective optimisation with an outline from "divide and conquer" to "networks of multi-agent systems".
- How the Multi-Echelon Time-Variant Knapsack Problem (METVKP) can be represented as a distributed problem.
- How the information feedback loop is ignored and that the current combination of humans and computer systems create a naive propagating queueing network.

The first section (origins of multi-objective optimisation) summarised the legacy of the theory of optimisation problems and described several strengths and weaknesses of the mathematical approaches from John von Neumann to the present. It outlined the root of the challenges in converting the Multi-Echelon Time-Variant Knapsack Problem (METVKP) to represent the supply chain problems with adequate levels of detail, as the previous literature review highlighted that implementations often are restricted to suit the optimisation library and not the problem at hand. The review included a discussion of how the "divide and conquer" algorithms and Dynamic Programming could be transformed to asynchronous distributed representation of the problem. This was supported in the previous chapter by a rapid increase in number of publications of multi-agent systems.

The review also highlighted the literature occupied with supply chain problems presented no clear assumptions about the feedback loop that exists in the real-world.

This was made evident by the application of optimisation methods in industry, which showed that the practice of exchange of information is naïve propagation in a queueing network of batch-information processors (human and computers alike). The literature review in this chapter gave evidence that perpetual re-evaluation is not possible as the information is delayed in a queueing network that distorts the information through self-interested local sub-optimisation without any indication of up- or down-stream consideration of consequences. The review also showed that delay is completely unnecessary as it is technically possible to convert the METVKP into an asynchronous distributed optimisation problem, where uncertainty caused by absence of feedback is substituted with incentive based forecasts (game theory) and partial information (forecasts) and prevention of premature commitment (time limits).

The previous chapter concluded from a systems perspective that the most productive intervention only can be a meta-intervention which perpetually re-evaluates the information available to the decision maker.

This chapter can conclude that the research occupied with optimisation is preoccupied with "fast identification of optimality" without consideration of whether the "optimality" can be implemented by peers, and it presents little attention towards the observable topology where information propagates in a queueing network.

Chapter 4 – Research Method

Introduction – This chapter returns to the desire to tackle the long-standing problems in supply chain management. It describes the research approach which is based on recasting the problems as an Agent Based Model (citing prior work)) and testing the model with case studies. It discusses the strengths and limitations of that approach, and explains why it is an appropriate choice for this dissertation. It introduces the hypothesis that complexity-based approaches have something to offer.

The past two chapters raised the notion, that the consequence of delay caused by propagation of information in a network of queues is unaddressed by the supply chain literature. However as taking actions based on outdated information seems contradictory to the pursuit of optimality, the key question becomes "does it matter?" The thesis thereby attempts to measure the effect of adding or removing delay in the context of a real supply chain, so that it can be better coordinated by operating on up-to-date information. To be explicit:

The thesis is that reduction of delay in propagation of information, reduces the penalty of being sub-optimised based on the wrong information.

The research method is inspired by the other case studies identified in the literature (Neumann et al. 1944; Simon 1978; Axelrod 1997; Kopetz 2011; Laplante & Ovaska 2011; Rzevski & Skobelev 2014; Borrill & Tesfatsion 2010; Shapiro 2007), and the authors' access to the industry (Chapter 5 – Case Study I: LEGO System). The hypothesis is that the problem could be solved through the usage of Agent Based Modelling (ABM) (ref. Chapter 2, p.30) which is described as:

> *At the highest level of abstraction one can argue that ABM is a goal pursuing system which combines discrete event processing through message exchange amongst agents, with internal rules of achieving state updates.*

The logic being that the distributed nature of the supply chain coordination problem can be represented naturally by the distributed nature of the agents in the agent-based model.

The challenge for this research is therefore to design, sponsor and subsequently create software platform which can be implemented by professional software engineers in Multi Agent Technology.

Prior to the research project, the author was engaged in programming the LEGO Brand Retail (LBR) automation tools for order processing, so data was already available, and that the LBR case was a feasible candidate for the experiment. Hereby the supervision of the software development process, conception of the system-wide tests and the test data was possible.

The second element in the research project is to attempt to clarify whether there is sufficiently strong incentive from the perspective of a set of performance metrics to "make the most productive intervention" by performing the meta-intervention to redirect focus from local optimisation of the physical activities in the supply chain, towards exploiting the information which triggers the physical activities in the supply chain.

Preliminary research using the available dataset gave the evidence that at 9-10% of lost sales was measureable and could be saved, whereby a 6-digit research budget was considered a reasonable investment for mitigation.

The research project could thereby be set up.

Catalyst for research	
- Problem	Lost sales worth several $-US million. Complaints about ability to respond to consumer demand. Queued work processes inhibit quick decisions.
- Opportunity	Research budget Availability of data
Preliminary information	Engagement as SCM-consultant at LEGO, LBR provided consensus of the management concern.
Problem definition	The most productive intervention in the part of the CES which the organisation was a part of, would be to eliminate the delay caused by decision making. Hereby the improvement would be measurable in lost sales, revenue & profitability.
Framework development	
- Conceptual	Deduced from Complexity Theory, Supply Chain Management & Computer Science, that latency/delay of information results in mismatch between correct actions and pursued objective.
- Theoretical	Literature providing a critique of application of optimisation in the field of Supply Chain Management
Research objectives	
- Research questions	Will reduction of the delay caused by decision making have a significant impact on the pursued objectives?
- Hypothesis	Yes – The case study is required to determine "how much".
Research design	Construct an Agent-based modelling platform which permits imitation of the real world processes, including calculation of KPIs. Then use the ABM in three steps: 1. The imitation of the base case 2. The creation of a perfect plan with perfect information 3. The insertion of constraints that cause delay so that the results degrade until they are comparable with the results of the real world.

	If the hypothesis is true – that delay has a significant impact on performance – then provide the ABM to a third party for verification and audit of results.
Data collection	Using financial and transactional information from LEGOs databases. No new information is needed.
Data analysis	Read the KPIs from the ABM.
Interpretation of findings	The KPIs are given prescriptively whereby comparison of the delay and the performance can be read without interpretation.
Report preparation and presentation	--
Management action	Out of scope.

Table 12 Research process (Cavana et al. 2001, p.48)

Due to the central element of the ABM, verification of the ABM is central. The test program therefore has to be extensive enough to assure rigour, replicability, accuracy, objectivity, generalizability and parsimony for all research stakeholders. To assure this, the ABM is handed to a third party specialist's consultancy during a second case study.

The research thereby requires that a queue-free model of the information network is created which permits observation of impact of change. The model must then be populated with data from real-world business, and the consequence of elimination of delay evaluated from the perspective which matters to the organisation hosting the case study.

With this approach the case study will inform the supply chain literature on the consequence of increase/reduction in delay of information and how such effect influences the ability to make the most productive interventions in the supply chain.

The ethical consequences of this research are no different than any other business improvement project: First, the impact of removing delay in decision making processes is assessed. If this is found to be of significant impact, the case study is scrutinised until the management team is convinced of its validity and the consequences hereof. Once the integrity of the case study is established the organisation is prepared for a pilot study which exploits the case study by taking it into everyday operation. This has a foreseeable social impact as employees who currently are engaged in the decision processes which currently are obsolete. However as such consideration is a part of the individual corporate social responsibility program, it is beyond the scope of the thesis.

The thesis only investigates the impact of removing delay in the decision making process, so that effort spent of making decisions which are critical for end-to-end supply chain productivity (profit, order fulfilment) are done based on up-to-date information in a coherent and rigorous manner.

Proposed by the organisation hosting the case study, the first case study investigates the "wisdom of the common" that if everyone in an organisation is working towards a good plan that is made based on perfect forecasts, then the business cannot perform any better. However as complexity science provides evidence that the future is unpredictable and thereby incomputable, the "perfect forecast" is impossible to realise. This is based on the idea that the forecast uses available information only, and that any action taken which impacts the environment will result in a response which in turn will change the situation which was taken for granted when the forecast was made. So by using simulation to evaluate the difference between different forecasting methods (dimension 1) in comparison to injection of delay in the decision making processes (dimension 2), the hypothesis that perfect forecasting can substitute delayed decision making may be dismissed.

Chapter 5 – Case Study I: LEGO System

Introduction – This chapter describes the problems faced by LEGO before the authors research started. It describes LEGOs and the supply chain sectors approaches to supply chain management.

5.1 The problem

LEGOs supply chain is well described in Oliver et al. (2007) where LEGO is a fast moving consumer goods manufacturer, that supplies 58,000 retailers with toys. LEGO prides itself with high product quality and achieves this through a decoupled supply chain with production into a global component storage, from which products are packaged. Achievement of high utilization of the moulding equipment is key to assure economies of scale, and coordination with packaging material providers is key in assuring responsiveness.

In November 2007 the Author participated in a team meeting with an unambiguous dialogue:

> Sales Director: *"The customers are screaming for stock, as their shelves are empty. The consumers want our products and I know that we have 80m EUR of unallocated stock in the warehouse. Why aren't we shipping?"*
>
> Logistics Director: *"I know we have 4 weeks until Christmas and that we have 80m EUR of stock in the warehouse. But don't worry – Sales & Operations Planning will never get a plan ready for how to allocate this stock so it can reach the retail outlets before Christmas."*

The message was astonishing as the supply chain was operating pan European with a total logistics process from warehouse to retail shelf of 4-6 days depending on where in Europe the destination was. The workload in the warehouse operation was at 81% - 83%. The transport companies were responsive to take on new loads. The retailers said that they would prioritise to take the goods in, because the products were amongst the most profitable in its category. However the Directors decided not to take action, because it would take too long to make the required decisions. This appeared odd: Why put all the effort into creating a quick response distribution system, when the main source of delay was decision making?

When mapping the decision process it became clear why the directors' decision was justified. The alignment process of how to allocate resources to demand would simply not permit a rapid response. It was a very efficient process that prevented any costly return flow and erroneous allocations but it was not suitable quick response with 40,000 pallets of stock.

Date	Day	Event
3/12/2007	Monday	Release the available stock to the Markets by Global Inventory Management Team
5/12/2007	Wednesday	Preliminarily allocate the stock to account managers in the markets
7/12/2007	Friday	Get preliminary response from the account managers which customer will take what.
10/12/2007	Monday	Re-negotiate the volume allocation to each market based on the account managers input.
12/12/2007	Wednesday	Perform the final allocation to the market
14/12/2007	Friday	Agreement of commercial terms, ask customers to submit orders for agreed quantities.
17/12/2007	Monday	Receipt of customer orders, release of deliveries to the warehouse
	Monday through Wednesday	Receipt of delivery time window to the customer DC operations. Immediate release of information to the warehouse.
19/12/2007	Wednesday	The first possible loading of goods.
20/12/2007	Thursday	Outer labelling, customisation, packing and preparation of loading.
21/12/2007	Friday	Last loading and dispatch (Get through Germany before Sunday!)
24/12/2007	Monday	Last delivery at customer DC
Christmas	Tuesday	Final dispatch from customer DC to retail outlet

Table 13 Process steps of why delivery in 4 weeks was not possible.

The attentive reader will notice that from first possible receipt of orders until the last delivery day there are 7 working days to dispatch 40,000 pallets (approximately 600 truckloads) which at 60 truckloads a day is impossible for the warehouse. However if the delay in the S&OP process was removed 20 shipping days are available in December, which would reduce the additional workload to 30 truckloads a day – a number well within the constraints of the warehouse operation. The "schedule" of information-processing is clearly a both a mechanism for coordination but also a bottleneck as it is completely synchronised around the employees ability to aggregate spreadsheets with information and upload it to the corporation's Enterprise Resource Planning system. The problem is – in other words – very similar to the problems mentioned in the literature review, with a clear indication of absence of asynchronous communication and interactive planning at the source.

A second element is the role of the ERP system, which performs complete rescheduling every weekend. This causes a change in the supply plan on what is to be produced following week, and how stock is assigned to orders. The problem with this, is that a forecast from a high priority customer – that most likely will change before becoming a confirmed order – will reserve scarce stock in advance of a lower priority customer. The higher priority customer's forecast may thereby be allocated the stock which then cause a backlog to be created with the lower priority customer, even though there in reality are no confirmed reasons for this. The next weekends rescheduling will notice the demand and attempt to create a schedule that fulfils all demand signals (forecasted and confirmed) and releases the prematurely reserved stock, but until then the low priority customer is awaiting stock which sits in the warehouse doing nothing because it has been allocated against a forecast of a higher priority customer.

The changes from the ERP system are also forced upon the suppliers using naïve propagation, whereby a single monthly rescheduling would take 6-8 meetings with a material suppliers to get the supply plan realigned. At LEGO, nearly 400 people influenced the bottom-up planning process, which was met by top-down requirements from the executives. This led to an alignment process with over 79,800 negotiations with self-interested optimisation motivated by the corporation's bonus schemes.

Persons (x)	Negotiations f(x)
2	1
3	3
4	6
5	10
10	45
100	4950
400	79800
x	$f(x) = \frac{1}{2}x^2 - \frac{1}{2}x$

Table 14 Example: How number of connections increase in a growing, fully connected social network

The KPIs of the different departments support the work towards compliance to plan, but when the plan changes people start stressing about sub optimising private interest to achieve the highest possible bonus – even when it is counterproductive for the supply chain as a whole.

During February-March 2008 LEGO accidentally turned the advanced planning and scheduling system off. Nobody noticed and performance of the business unit did not appear to degrade, though employee satisfaction slightly rose through the same period[16].

5.2 Sector wide problems

In lieu of the literature review – Chapter 2 & 3 – it cannot be stressed enough that both humans and computers process information in batches until the ERP system stopped its weekend rescheduling. At that point in time, the ERP system became transactional in real-time. Usage of emails and spreadsheets for decision making left the process delayed only by humans, who now were the only batch processers. However at least it was asynchronous.

As this thesis is providing a contribution to the practical application of modelling techniques the following is worth emphasising three points:

1. The need for interactive decision support systems which operates asynchronous has remained unaddressed by the literature; has left talented, fast humans stuck between batch-processes and left expensive computing systems waiting for human inputs.
2. The Human/Computer coordination problem has not been solved with S&OP and ERP. In fact it has created a queueing network – a supply chain of

[16] Similar cases are described by Snapp (2009) page 36.

information made of database systems, emails and spreadsheets – in which parcels of information are relocated between information processors.

3. It shall be clear that the impact of delay caused by scheduled decision processes (Table 13) is ignored in supply chain management literature and that this problem needs more attention.

The challenges is now to determine a new approach – a new supply chain model – which overcomes these problems.

Chapter 6 – Formulating the New Supply Chain Model

Introduction – The problem concerned with batch-processing of information in a queueing network of decision makers, requires a systematically developed solution. The chosen approach is therefore grounded in a critical analysis of the situation described in the previous chapter. In particular, the following questions will guide the analysis:

1. Where is information produced?
2. Where is it sent?
3. What delays it in getting there?
4. For how long is it up-to-date?
5. Which design strategies minimise the unavoidable delay in information processing?

These conceptual questions are then supported by a more technical discussion:

6. Which software design principles should be suitable? In chapter 2 it was clarified that ABM is the most promising approach. We will extend on this.
7. Which practical principles for agent-based design are required for effective information processing?

The answers to these questions will clarify detailed design choices which are central to transform the information flow in the supply chain from a queuing network to a real-time asynchronous agent-based system. The implementation of this formulation will subsequently be used in case studies which will measure the influence of removing the information delay.

In order to assure that the results from the case studies can be completed with rigour, a comprehensive test suite is presented in summary. The 3658 tests which were used to verify the developed multi-agent system are summarised in appendix "A.2 Test program (extension)". Other discussions which are central to the ABM design, but peripheral to the impact of delay of information, are included in the appendices. Examples are (a) Accounting and the associated challenge of determining how to maximise order fulfilment and profitability and (b) Manufacturing Resource Planning (MRP) including the transformations of bills-of-materials and calculation of materials requirement planning.
In combination this description should contribute to convince the reader that the NSCM has been exposed to sufficient consideration of the required concepts.

The objective with this chapter is thereby to convince the reader that:

1. The design deals effectively with the batch processing problem, so that asynchronous implementation is feasible.
2. That the implementation is capable of considering the technical cases to such an extent that the supply chain optimisation problem can be solved "without reduction to suit the optimisation suite".
3. That the testing provides sufficient evidence to consider the solutions repeatable, consistent, and logically valid.

6.1 Overview of concepts

Where is information produced?

The businesses of the real world have contracts, operate from multiple locations and are constrained by physically limited processes. The agent interactions must reflect these constraints to bear realism. In the figure below (Figure 6, page 60), a contract can be valid for transactions to multiple geographical locations though the contract is a part of 1-to-1 relationship between a seller and a buyer. In the *new supply chain model*, the exchanged messages for supply and demand for objects may therefore be via an agent that represents the contractual relationship and not necessarily the geographical constraint. This would be typical for business units acting as a trading-agent for its business' internal users of the suppliers and customers, as well as shared services provided by business headquarters in multi-national corporations to their respective business units. At a finer level of granularity the logistic processes within a physical location may depend on one another just as trade between businesses. In principle there is no difference in the transactions, as one process within a factory may require a particular subassembly before it can confirm a request to marshal and deliver a final product for dispatch. In this way, the new supply chain model uses the principles of hypersimplices and hypernetworks (Johnson 2012) to maintain its multi-level organisation. This provides a contribution to knowledge, as systems of systems of systems (see Figure 6, below), which is only delimited in granularity by the availability of data to create the model and replicate the real-world's pending, committed and

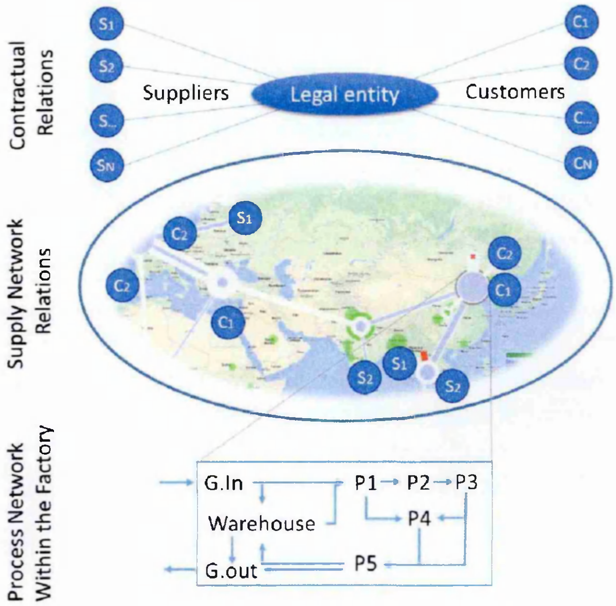


Figure 6 example of agent-to-agent relations

executed transactions. With this model, it is no longer needed to reformulate the supply chain problem into a mathematical model, as the agents in interaction will solve the scheduling problem using messages.

The difficult part is to provide a coherent set of methods and performance indicators to guide the self-interested collaboration of the agents and represent an appropriate level of granularity.

Practical usage of the supply chain simplices in agent-based hypernetworks has revealed that the following methods are useful for representing self-interested collaboration. Methods which consider:

- Commitment time and enable the state-update from "confirmed" to "committed", including how to interpret real-time rescheduling of committed deliveries and stock states
- Volume and weight limits on locations representing physical sites
- Delivery consolidation and collection as result of scheduling
- Dispatch & receipt limitations on sites
- Setup of production lines, with changeovers and bills of materials and quarantine time
- Storage and production sequence restrictions
- Contracts, including minimum order quantity, minimum order increment quantity and minimum order value
- Schedules of availability of physical facilities
- Representation of interfaces and catalogue prices of different companies
- Calculations of "safety stock" of particular items (£,$,€, units, ...)
- Changes in the physical world, for which the model does not have processes. Examples include errors and manual delivery assignment.

Performance indicators which guide agent behaviour:

- How many requests are satisfied/unsatisfied
- How profit is possible if all requests are satisfied
- Current projected profitability (including forecasts)
- How much inventory and work-in-progress is involved at each process

Conclusively information is produced by every activity (physical or transactions of information) by people and machines in each business in the supply chain.

Connecting the virtual world to the real world

A subject not debated in detail is the fact that two different systems which use the *new supply chain model* and which have *different* internal representations, must be capable of negotiating and creating a shared schedule. The system thereby only needs a single interface that is capable of interpreting the schema by which remote requests are made to translate them into a local context. If businesses start to publish their interface schemas on publicly available sites, it would give their supply network the advantage that follows an ease of coordinated collaboration. The only change which would be required for current practices is the requirement to respect mutual commitment horizons, as this enables both parties to perform the most productive allocation of resources to fulfil each other's requests.

As noted in section "3.3 Feedback loop", the system response from the complex economic system (to any action taken by an agent) is beyond the edge of computability (Prokopenko et al. 2009) and therefore not available until the environment responds. However, when the response arrives, it is essential to evaluate the consequence as quickly as possible to clarify if corrective actions are needed. Consequently, the interface to the complex economic system must be adaptable as the economy evolves, so that information may be transferred with minimum delay from any remote source where it is created. It is therefore advantageous to establish as direct connections as possible to the source of information (sensors, mobile devices, users, data-centres) that are used for scheduling.

Where is it sent?

Based on the general outline described so far, there are no constraints which inhibit modelling of the SC as a set of agents representing a decision-making unit, which negotiates using messages, as illustrated in Figure 7, below. The key component is that the agents, representing each of the decision making units

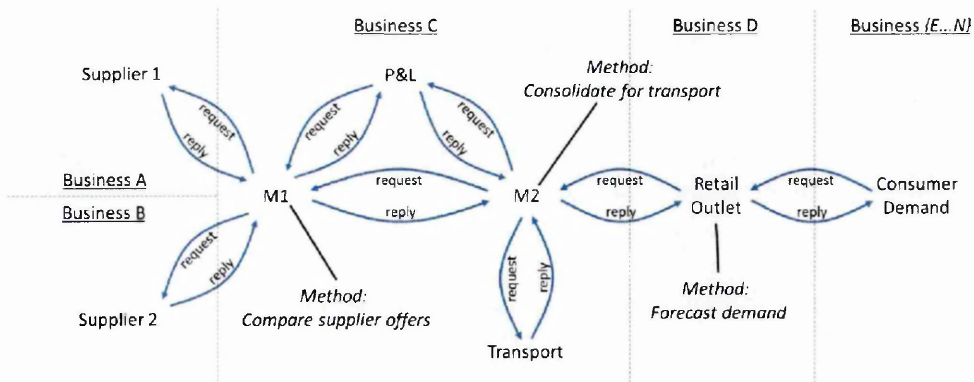


Figure 7 Illustration of a supply chain with multiple businesses using message exchange.

(supplier1, supplier2, M1, M2, P&L, retail outlet and the consumer) do not need to disclose their internal methods. For example, M1 may have a confidential method for comparing the offers from the supplier; M2 may have a special set of rules for calculating transport costs of delivering to the retail outlet. It follows then, that each agent can operate rationally but does not have to. At the same time, each agent does not have to act as if it was a singular monolithic entity, as shown in Business C, where M2, M1 and the P&L are collaborating. M2 can, for example, ask the P&L to finance M1's supplies, so that M1 can purchase and perform the subassembly that M2 needs to fulfil an order for the retail outlet. The collaboration also works between businesses, for example between the C and D, where D may openly share its forecasted orders without committing to them to business C until it has to. In this way the sharing of information contributes to minimize risk of uncoordinated decision-making.

These requirements permit us to determine what agents need to be capable of, at the level of autonomy and message exchange.

<u>Agents</u>

As a large body of knowledge exists on Agent Based Modeling which spends significant effort describing agents, it is worth emphasizing that agents interacts with its environment: The role of the environment is the propagation mechanism for actions made by other agents. In optimisation problems, such as the knapsack-, the assignment- and scheduling-problems, there are two very important groups of agents: Agents governing **resources** and agents governing **demand** for resources. In the rest of the thesis the local environment in which agents exchange messages directly is called a resource-demand network (RDN). For more details on this subject visit Bertsekas (1979) and Rzevski (2014).

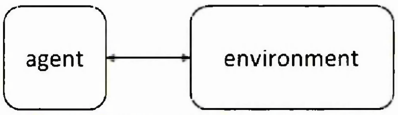


Figure 8 Agent and its environment

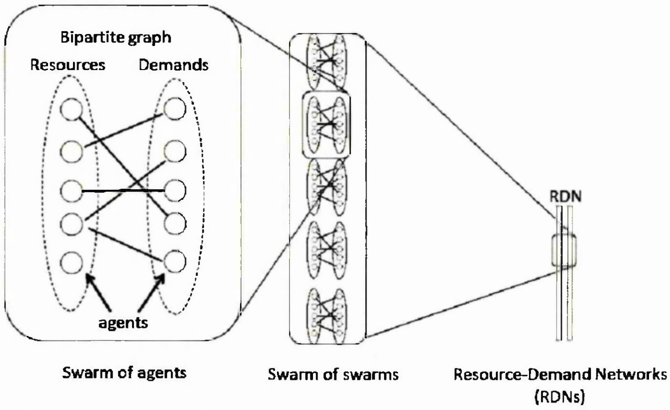


Figure 9 Resource Demand Networks as swarm of swarms of agents

Agents are finite-state-machines which respond autonomously to messages. Though the transition from one state to another are governed by rules, there are no constraints on the evolution of the rules, nor any requirements that all agents obey the same set of rules. This means that real-world behaviour, such as cheating and situational awareness and creative adjustment of behaviour is possible to model.

Agents control resources (inventory) **and can transform objects using processes**. In supply chains only, three fundamental processes are necessary:

- **Storage:** a process that ages an object at a location.
- **Transformation:** a process that converts a set of objects into another set of objects using a *recipe*.
- **Relocation:** A process that updates the location of a set of objects after a duration of isolation, which represent the transit time.

Agents have relationships to other agents (links) and perform self-interested collaboration with other agents using transactions of information based on their internal interpretation of messages exchanged with other agents.

Agents perform the following **transactions**:

1. Request trade (a non-binding request from customer to supplier)
2. Reply to request (from supplier to customer)
3. Propose transaction (from customer to supplier)
4. Confirm transaction (from supplier to customer)
5. Commit to request or abort request (either party)

(1) Agents send *request* to trade to other agents using messages which for example contain the following information:

(2) Agents who receive a request may choose to *reply*, but do not have to. This minimizes the message exchange as agents, who do not have a schema for interpretation of messages, by default will ignore messages they cannot respond to, including corrupted messages. This implements the principle that communication must always be based on the receiver's premises, and that the sender must understand the receiver's interpretation schema.

The second element is that the request is not binding, so that the agents requesting information may gather and evaluate the replies based on their internal rules and performance indicators. This assures that if there are multiple agents who are interested in the trade, the best bid is always found based on what matters to the buyer.

Finally, it is not prescribed how information is transformed into a decision of what the reply must contain. It is therefore fully acceptable that information may be transformed by a human (or human proxy) who replies based on incomplete information and biased irrational gut-feelings. The only thing that is required is that the agent who sends the reply has a method of compiling it in a manner which can be interpreted correctly by the receiver.

(3) Agents may propose a transaction, if they find a *reply* attractive according to their performance indicators.

(4) Agents who receive a proposal for transaction may confirm, but do not have to commit to the proposal until commitment time. Until commitment time, no proposal/confirmation is guaranteed. It is a planned transaction, and as any disruptive event may occur at any time, either of the agents can cancel the proposed transaction completely by referring to the unique message-id. If this happens, the request-reply-propose-confirm-sequence is restarted, typically in attempt by the buyer to fulfil its demand for resources.

Agents can contain multiple agents: who or what the agent is supposed to represent is determined by the system designer. As an agent may have the role to act only as an interface to other agents, it may also represent a swarm of agents, who are not designed as interfacing agents. This reduces the complexity constructively, and can conceal very complex agent behaviour (Rzevski 2011). This allows the real-world complexity to be reflected at multiple levels, as emphasised by Johnson (2012). Beinhocker (2007) emphasises the contractual relationships; Rzevski & Skobelev (2014) emphasise the interfaces of the physical world where transactions happen, such as the nodes within the supply network's where suppliers deliver to customers.

What delays it in getting there?

Chapter 5 – Case Study I: LEGO illustrates how the coordination in the S&OP process was inhibited by delays between scheduled meetings between different parties:

- global inventory management and the account manager,
- the account managers and the customers, and,
- the ERP system and the users.

The framework for message exchange, presented so far, allows agents to react autonomously to events of both local and remote origin. As there are no batch-processes or delays[17] the queueing network of batch-information processors can be imitated by injecting delays. The assumption that remains unaddressed is how to model the delay and the source that causes the delay. The illustration below attempts to put this question into a framework representing any node in the network of agents.

[17] Beyond the computational message parsing

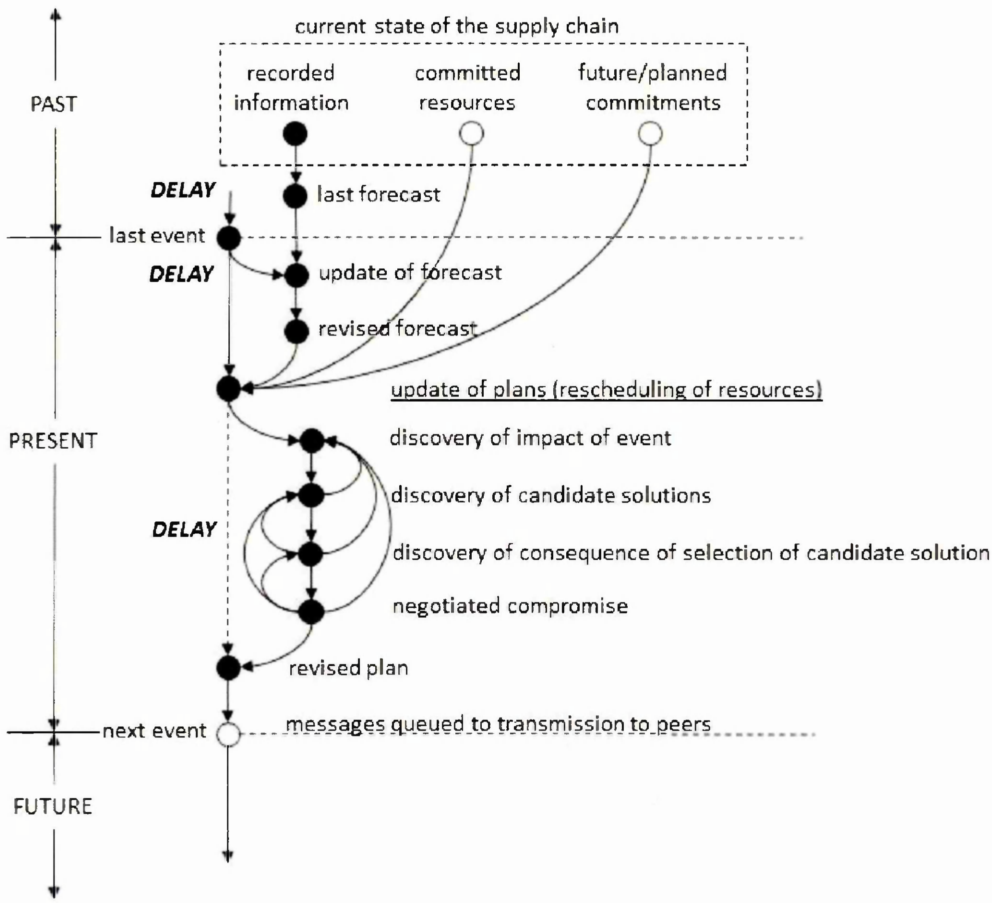


Figure 10 Steps in the translation of available information into agent behaviour (actions)

For completeness it should be noted that niches of the scheduling literature is occupied with the phenomenon of delays as a discussion of online versus offline scheduling. Leung (2004) writes:

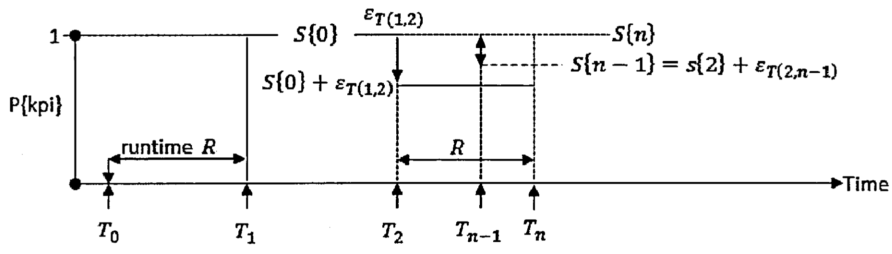
> In online scheduling even less information is known a priori. The information is released gradually to the decision maker. The decision maker does not even know in advance the number of jobs that are going to be released. He knows nothing about release dates or processing times, whether or not they are deterministic or random. He does not even know the distribution of the random variables. There are two types of online scheduling models: in the first type, the decision maker is given a job's exact processing time the moment it has been released. In the second type, the decision maker knows the processing time of a job only when the job has been completed. (Leung 2004, p.847).

For classification purposes this may be a convenient discussion, but for practice the supply chain agent may require responses from remote peers, whereby any proposed schedule may never be confirmed. The scheduling problem is therefore not an "online or offline" problem as Leung discusses, but a hybrid problem in which some parts are confirmed, unconfirmed and unknown (though forecasting may compensate under the assumption of stochastic properties). This must be included in our assumptions when verifying posits of optimality as the retrospective analysis of the committed schedule will differ from the prescriptive by not having to account for delays in information, which gives the presumption of perfect information and perfect coordination. To bring planning and management closer to realism, we will have to put these perspectives in contrast and understand the consequences.

For how long is it up-to-date?

As it is impossible to know for how long the information is up-to-date in an asynchronous system, expecting the worst case conditions is required.

With the assumption that time has a significant influence on performance, responding rapidly to changing conditions is considered essential (Rzevski & Skobelev 2014; Laplante & Ovaska 2011; Kopetz 2011). To challenge existing assumptions on this topic, it is necessary to dive into details of how computation is performed. As most computational processes are described from initial conditions without memoization – such as the simplex algorithm or the Hungarian method – unnecessary overhead is added by restarting the batch computation perpetually with every update. It is therefore necessary to take a systematic departure from this assumption. Figure 11 (below) illustrates the key principle: Complex system thinking requires the consideration that any individual event will change the optimal solution slightly, and, that the emerging result of several events may produce a significant change (Rzevski 1998). The requirement to the algorithms is that they must be capable of producing a result in the time between two events. This implies that the runtime R of the algorithm must be close to zero, as any two events may happen at any time. Without memoization, repeated computation from initial conditions will waste the time on re-computing the parts of the schedule which have not changed. The algorithms which thereby do not apply memoization are thereby not c-competitive in comparison to algorithms which do (Burckhardt et al. 2011).



Assuming only that the dataset D is complete at T_0, the computation of the schedule may be initiated. The algorithm halts at T_1 after detection of a feasible solution $S\{0\}$ based on $D\{0\}$.
The performance of the schedule at T_0 is $P(S_{T=0}\{0\}) = \emptyset$, but after the computation (at T_1) the performance of the schedule is $P(S_{T=1}\{0\}) = 1$ i.e. a perfect schedule based on perfect information.
The solution $S\{0\}$, however is only valid in the time period from T_1 until T_2, where a new event $e(T_2)$, will cause the performance of the solution $S\{0\}$ to drop, as the dataset $D\{0\}$ which was available T_0 is different than the dataset $D\{2\}$ available at T_2. The difference is referred to as the error term ε, whereby the revised scheduling problem requires that one determines whether it is feasible to compute the schedule anew. If a commitment time occurs in the time between T_2 and T_1 and $S\{2\} \neq S\{0\}$ then rescheduling should occur, to reflect the benefit of using up-to-date dataset available at T_2.
However if during the rescheduling $R_{T(2,n)}$ any event $e\,(n-1)\,|\,n > n - 1 > 2$ will occur, the rescheduling will have been initiated with the dataset $D\{2\}$ available at T_2, which will out-dated by the time T_n where the rescheduling completes. This leaves the set $\Delta D = D\{T_n\} - D\{T_2\}$, whereby aggregate error $\sum_{T=2}^{n}(\varepsilon)$ will be the result of $S\{n\} - S\{2\}$.
The error will thereby persist as long as the rescheduling process is longer than the time between disruptive events:
$P(S_{T=2}\{0\}) = 1 - \varepsilon_{T(1,2)}$.
$P(S_{T=n-1}\{2\}) = 1 - \varepsilon_{T(1,2)} - \varepsilon_{T(2,n-1)}$.
$P(S_{T=n}\{2\}) = 1 - \varepsilon_{T(n-1,n)}$.

Figure 11 Mathematical details of how updates during rescheduling establish the error term

A suitable strategy has been devised by Rzevski & Skobelev (2014) at the cost of memory of maintaining two schedules. One schedule is in the state as "latest best known schedule" $\{n-1\}$, to which any requirement to commit is available at any time (as read-only), and the other, is the "next schedule" $S\{n\}$, in which the new event is propagated to determine the consequence of the update. At any time where commitment of specific resources is required, the schedule maybe used where the performance $P(S)$ of the solutions is higher: $max(S\{n-1\}, S\{n\})$. Should the process establish that a solution is feasible in which $P(S\{n\}) > P(S\{n-1\})$, the pointer may be updated in such manner that the "latest best known schedule" is disrupted minimally and represents the "latest best known schedule".

As a side comment, these requirements exclude genetic algorithms and particle swarm optimisers, as these methods initiate randomly and hence will be less effective than the structured propagation used in event triggered ABMs. Another point is that it will intuitively be known whether it will be possible for a supplier to respond to any incremental changes in the supply schedule as the incremental change is faster to evaluate than a complete rescheduling. The focal agent could thereby represent this knowledge using stochastic profiles of response times, and consider to absorb the cost associated with an error in coordination in contrast with the cost of the risk associated with a complete disruption of the suppliers operation.

A suitable approach is given by Bertsekas in 1979, refined in Bertsekas & Castañon (1991) and applied in practice over a time period from 1994 to 2014 by Rzevski & Skobelev (2014), which uses alternating auctions. The algorithms establishes a queue of messages which are processed sequentially for each assignment problem. Each message prompts for either a resource-side or a demand-side auction, which establishes relationship between resources and demand for resources (and inversely) as resource-demand networks (RDNs – ref. Figure 9). Each message from the class' message queue may be processed independent, as long as a relationship check is made at the end of the auction to evaluate whether the auction winner needs to break an existing relationship to another entity as a result of the auction process. The risk of indeterminacy race is therefore prevented as inferior updates are not applied to the winner, but handed over to the 2^{nd},...,k^{th} auction winner instead.

The worst case outcome is therefore that the solutions is improved at the cost of breaking the k^{th} relationship. However as the update improves the solution with $\Delta S = S\{n\} - S\{n-1\}$ where the broken relationship only existed in $S\{n-1\}$, the difference must be $\Delta S > 0$. Hereby the solution is incrementally converging towards optimality.

Even for initial starting conditions, this method scales in worst $O(m+n)$ per assignment problem, i.e. per class of item in a supply chain node as RDN, where m is the number of resources and n the number of demands, in contrast to the simplex method which in worst case exhibits $O(m^n)$. Whilst this may be of formal interest, **this type of runtime evaluation ignores the effect of memoization in the RDN whereby the runtime does <u>not</u> depend on the size of the problem, but on the length of the propagation path of the event causing the update.**

Of the 800 articles and book chapters read during these studies, only Bertsekas (1979) notes that the propagation path may depend on the topology of the problem, and only Burckhardt et al. (2011) attempt to address the problem systematically. Given Burckhardt et al. (2011)'s results, the limited exploration of this area combined with the very novel innovations in memristors and flash memory, this is a very promising area of research. In particular as the root of the ideas that memoization takes up, allows revision of ideas back to the birth of computability: Turing (1936) raises the conclusion (p.231) that the Hilbertian Entscheidungsproblem can have no solution (ref. §11) under the assumption that a "solution" may be effectively verified as belonging to the solution set. This chapter extends the definition used by Turing (1936) by including time as a variable, whereby all problems must be transformed into pursuits of multi-objective optimisation. This view denotes that the solution may be optimal without being the mathematical optimum under conditions of exhaustive search, without such time constraint (Coello 2006). The problem, in other words, is not about being mathematically correct, but rather about being *<u>judiciously correct given the time which is available</u>* (Aho et al. 1974, pp.69–70). The reader should thereby expect a slightly different usage of the Hilbertian Entscheidungsproblem as the definition also must include the constraint of the number of steps permitted for the computational process. Exhaustive search of a large solution landscape – for example – may thereby return an invalid answer, if it exceeds the permitted threshold of the parameter for computational steps. This extends discussions by Arora & Barak (2009) beyond the scope of this thesis. In principle this is not more novel than playing a game of football in 90 minutes. The game will end with a winner & looser or undecided. The requirement in either case is that time is central to the strategy which is deployed to identify optimality.

Based on this direction we may construct a strategy which includes the additional time constraint in the evaluation of the Hibertian Entscheidungsproblem, such that, a solution is *only* considered to belong to the solution set, if it can be efficiently verified to be identifiable within the time steps available. As the supply chain problem is a distributed problem, this implies that the time steps depend on both the internal propagation path for the decision making agent and any external peers which need to include the changes, to verify that the solution is feasible. The definition of the strategy is thereby as follows:

Definition of strategy: Maintain a solution such that the error term of responding to changes is minimised.

The strategy is thereby *c-competitive* at any time, if the error term caused by the individual event is less than the aggregate error caused by its runtime. This implies that the disruption to existing schedules is minimized by propagating events locally before communicating externally.

Figure 12 Minimize error of disruptive events by propagating changes in real-time

An assumption which so far has not been discussed is whether the plan needs to be communicated or committed to with immediate effect? As illustrated in Figure 14 (below, p.71) the progressive chronological commitment of a "Yes/No" or "Do/Don't" decision is expected to have future consequences as the decision is irreversible. The literature on advanced planning and optimisation in supply chain – including Leung (2004), Stadtler & Kilger (2005), Shapiro (2007) and Oliveira & Gimeno (2014) – do not present any assumptions about this question. The argument could be that it is assumed that information is shared in real-time, and hence is negligible? However as it is well-documented that rescheduling is performed by ERP systems in batches which typically are computed overnight or on a weekly basis (Snapp 2009; Dickersbach 2009; Shapiro 2007), this assumption cannot hold. The earliest record of the assumption of what to do in future actions, was identified in Neumann et al. (1944, p.19) where:

> [However,] it would be an unnecessary complication, as far as our present objectives are concerned, to get entangled with the problems of the preferences between events ***in different periods of the future***. [Footnote: It is well known that this presents very interesting, but as yet extremely obscure, connections with the theory of saving and interests, etc.]

Whilst not having the required information, one must also consider the possibility of having the wrong/outdated information. In Harrington (2008) a discussion of how strategies which include misinformation may be (ab)used to obtain a temporary advantage. Harrington's discussion, however, is limited to the concept alone, and does not clarify to whom what information/misinformation is more valuable at what time, nor how to consider the dilemma of alliance building which may occur between different agents in the supply chain. Rzevski & Skobelev (2014) use the concept of delayed commitment, by maintaining a distinction between "current state" and "next state" – and – only communicate once it is clear that the "next state" is favourable in comparison to doing nothing in the "current state". This approach is *c-competitive* (Ajtai et al. 1994; Aspnes 1998; Leung 2004, p.327) and may be extended by the following definition:

Definition of strategy: Maximum delayed commitment

A strategy is guaranteed to be *c-competitive* if an agent communicates the requirements needed by self and peers – on which it depends – at the time required for the peers to take action. It may therefore provide none or complete transparency of plans, as long as it assures that plans are aligned for the time horizon for which commitments must be made and makes it explicit to which parts of the plans it commits resources. Commitments beyond the required horizon lead to no advantage. Commitments beyond the required time horizon lead to premature exhaustion of resource reserves, which make the strategy *non-c-competitive.*

Figure 13 Definition of the resource commitment strategy "Maximum Delayed Commitment"

As no research was identified during the literature review which presented application of the strategy, this theoretical contribution is assumed novel for the domain of scheduling, though it is well debated in game-theory concerned with choice of strategy given what is known about the opponents options and outcome (utility).

The classical method of evaluation of a strategy is discrete event simulation with perfect information (Harrington 2008; Kogan & Tapiero 2007). The simulation is then compared to a known optimal result (Van De Ven 2005; Harrington 2008) which is referred to as the perfect play. When the solution landscape cannot be exhausted a tournament amongst different strategies is still considered suitable for theory development. This idea is hard to justify under realistic business conditions, as information is incomplete at any time (see Figure 43, p.127). Proponents of game theory typically suggest to compensate for this through a systematic approach, where the evaluation is performed reflecting the discrete steps of progress through the simulation, but uses known conditions only. This involves introspection which includes a forecast of what the opponents will do (Harrington 2008). This would conceptually work perfectly if the system is permitted to assure coordination and is capable of executing its play without errors at each discrete time step. But is this assumption realistic? In the supply chain there are inherent delays in all decision processes and this will inhibit the ability to coordinate even if the supply chain would be capable of executing its plan perfectly. An axiom must therefore be present to reflect the error term caused by delay in decision making, so that a choice – that retrospectively is wrong – will have a negative chronological impact in the longer term as it will prevent the solution s_{max} from being reached. Figure 14, below, attempts to illustrate the progressive chronological elapse where the solution set grows as a product of time, and whereby it may be presumed that the error of commitment based on incomplete information will aggregate as time elapses from $T = 0$ towards $T = n$, as commitments are irreversible.

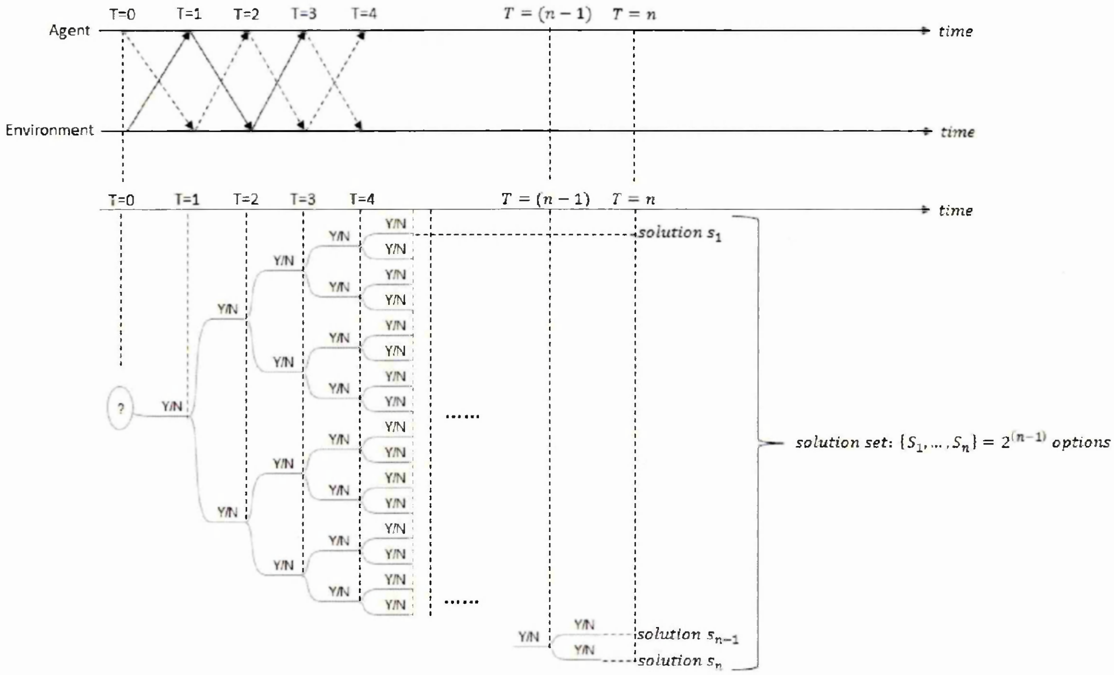


Figure 14 (Top) Emergence of Boolean options over time. (Bottom) Size of solution set over time.

While a network of a few agents may succeed in finalising the message exchange before another piece of information is received to which commitment may be made, it is highly plausible in a network of realistic size of interacting agents that – at the time which a decision needs to be communicated – the message exchange will not be finalised. It should therefore be emphasised that game theory's general assumption of flawless execution of moves, becomes less probable. This is very important, as alignment of plans reduces the risk of flawed execution – in the supply chain for example due to synchronised arrival of supplies – and that delay designed into the alignment process will increase the error term.

The most productive intervention through commitment of resources in the supply chain, must be evaluated with respect to the constraints that optimality is bound to a chronologically progressive elapse in which information is made available.

The 2x2 matrix below gives an indication of the consequences.

Delay?	**Retrospective analysis (Future information available)**	**Prescriptive analysis (information limited to present & past)**
Assumes No delay	No Error – Perfect strategy possible to compute and execute	Error term product of imperfect knowledge of future options, including forecasts
Assumes Delay in interaction	Error caused by imperfect coordination. Perfect strategy conceivable but not executable	Error caused by imperfect coordination in addition to imperfect knowledge

Table 15 Summary of error caused by delay when comparing prescriptive and retrospective analysis.

With much appreciation of investigations in supply chain models – which contributed in their own way – it may here be concluded that this evaluation provides the foundation for a departure from other publications, as the conclusions based on assumptions of either perfect information or perfect execution simply bear lack of realism.

Whilst the Operations Research community quietly frowns upon the reductionist approach that most models take, the retrospective analysis should be frowned upon – even for benchmarking purpose as it does not recognise this form of errors. Strategies, however, which bear the realism of the conditions of decision making which the agent is exposed to in real-time – between last known and the next event – provides much more promising venues for research.

So far the assumptions of "how optimality is defined" has been evaluated from the perspective of the individual agent insofar as how delay in transformation of raw information into decisions, influences the outcome in combination with a resource commitment strategy "maximum delayed commitment" that keeps options open for as long as possible to prevent premature commitment. The subject matter of how and where information comes from has been assumed at some boundary which is yet to be defined. However as any node that will relay information will be a source for distortion and delay, the architecture of the information network which the focal agent is a part of, must be considered. To put this into contrast of supply network optimisation which has been studied widely to determine the number of factories and warehouses and their location to minimize the production and distribution costs, there are no studies of application of such or similar methods to determine the information network on which the supply network depends, which guarantees that the cost of the error caused by delay in information is minimized.

It is well known from queuing networks (Leung 2004) and studies in information propagation (Kleinberg & Easley 2010), that the delay associated with getting information from the source to point of exploitation, is thereby correlated with the length of the chain. But its impact has so far not been considered in literature on optimisation methods with application to supply chain management[18]. In the literature review the only sources identified which treats this problem from an analytical perspective is concerned with epidemic modelling (Daley & Gani 2001, pp.133–150) and information permeation in social networks (Dezső et al. 2002; Barabási & Bonabeau 2003; Menezes & Barabasi 2008; Albert et al. 2001; Albert et al. 2000; Kleinberg & Easley 2010).

To remain pragmatic in the approach, relay of information using the internet is – for the problems at hand – considered near instantaneously (200 millisecond for a world round trip for a data package), in comparison to relay through ERP systems (with relay after rescheduling on a weekly basis)[19]. Human intervention, considered as an information processing node in this network, where, for example information is received as email, transformed through updates in spreadsheets and communicated either to the supply chain model or 3rd party applications, is also a source of significant delay (Özkarabacaka et al. 2014): As delay in propagation of information is associated with an error in decision making, the argument – that humans should maintain the ability to override and intervene directly with information in supply chain models – is outdated. Human interests,

[18] References to studies in biological warfare available for policy making (tertiary sources) indicate that studies exists in which maximisation of effect of biological weapons exist, but the primary sources are not publicly available.
[19] This is common knowledge in industry – hence no references.

as decision-makers, are much better represented through software agents as proxies which are present to make decisions 24/7/365. This observation aside, the infrequent updates to plans caused APS configurations, where information may remain outdated for a week or more, is a far greater source of disruptive updates than the asynchronous information processing performed by people (Stadtler 2005). In addition, when APS updates are transmitted, the aggregated changes will be more extensive and require complete, rather than incremental rescheduling by the peers.

Delay may thereby be observed as follows:

- Transmission, in milliseconds – though this latency is enough to be critical for high frequency stock trading.
- Queued for processing:
  - Ranging from milliseconds to weeks in systems, and
  - With unpredictable delay when waiting - for example - in someone's email box.
- Being processed:
  - Ranging from milliseconds to hours in systems, and
  - With some stochastic duration profile if processed by a human.
- Queued for transmission after processing, typically in the range of milliseconds.

The ability to shorten the chain through which the information must propagate is thereby equal to a relative reduction in the error term caused by delay.

Definition of strategy: Maximize external connectivity

An information network strategy which seeks to create new connections is guaranteed to minimize the delay if it increases connectivity towards nodes from which signals propagate.

Figure 15 Definition of strategy to maximize external connectivity

Whilst this strategy may be appear as a statement of the obvious to the practitioner, the influence of changing connectivity is not discussed in the supply chain literature. Based on the applications presented in the sources of epidemic modelling, supply chain management appears to be able to augment existing models using knowledge from this promising area for research.

Figure 16 (below) attempts to summarize the overall problem illustrated as the line "limits to realism".

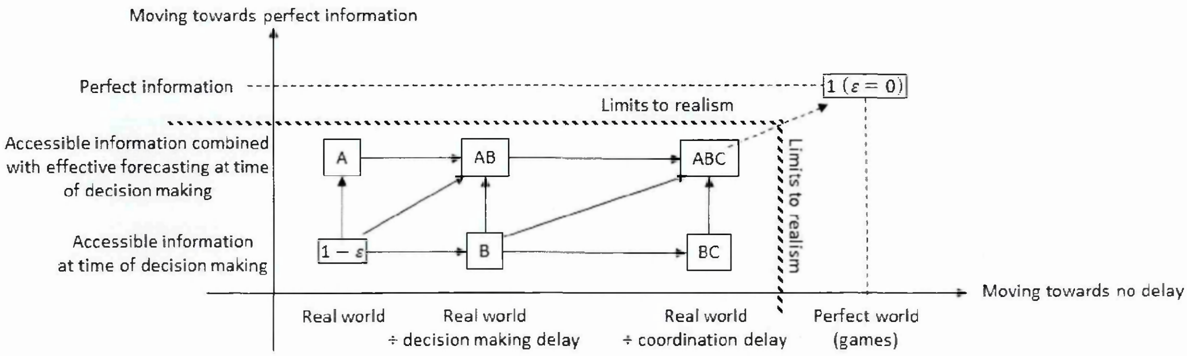


Figure 16 Reduced error caused by minimizing delay of information and alignment of actions in the SC.

Which design strategies minimise the unavoidable delay?

Collectively three design strategies thereby reduce the error under realistic conditions:

- Operate with **maximum delayed commitment** by sharing plans openly with relevant agents, but make it explicit what is committed and what is not committed.
- Maintain the solution through memoization so that the error term of responding to changes is minimized.
- Maximise connectivity to sources of information to minimize delay cause by the structure of information channels.

Chapter 5 – Case Study I: LEGO still leaves the question open of how to represent the human preferences and subtle information in the agent based model. Transfer of human preferences to a system as a set of rules or preferences has through experience proven to be difficult as the translation process from intention or idea into rules of behaviour is far from trivial. However as the purpose of the business is to satisfy customer demand and maximize the profitability for its owners, the task might be simplified significantly: Preferences express by people can be performed as changes to a schedule that is computed by first maximising order fulfilment and secondly maximizing profitability. Any change made by any human to such a schedule can then be illustrated through simulation. It may then be decided whether the optimal solution identified by the system based on explicit criteria, shall be overruled (by people) in favour of subtle or ethical criteria which are more difficult to model explicitly. This design guideline should also result in a higher rate of adoption as the system does the tedious work, whilst people can overrule the proposed schedule. For now, this requirement means that simulation and online transactions must be able to coexist as a part of the system design.

A few notes should be made with a view to critique of previous models: In a review of supply chain models by Melo et al. (2009) it was discovered that 75% of the literature mainly was focused on costs, compared to 9% multiple objectives and 16% on profit (Melo et al. 2009, p.408):

> "In addition to these findings, we note that the large majority of location models within SCM is mostly cost-oriented. This somewhat contradicts the fact that SCND[20] decisions involve large monetary sums and investments are usually evaluated based on their return rate."... "...Moreover, substantial investments lead to a period of time without profit. Companies may wish to invest under the constraint that a minimum return will be gradually achieved." ... "By considering profit-oriented objective functions, it also makes sense to understand, anticipate and react to customer behaviour in order to maximize profit or revenue. This means bringing revenue management ideas into strategic supply chain planning." Melo et al. (2009, p.410)

The last statement cannot be emphasised enough: Revenue management has been left out of consideration of supply chain design for most of its history, disregarding the fact that cost reduction is a question of minimizing the cost-driving activities even though some cost-driving activities also may be highly profitable. Combined with the observation that 75% of the articles are associated with facility location problems, this observation should raise alerts with the critical reader, as facility location problems are the most prominent business investments and influence many jobs. But if they only are evaluated from a cost-perspective, the models will favour facilities, which combines economies of scale, forcing a centralization into the planning approach. This is a serious problem when the models are used to inform management decisions. The chosen approach with focus on maximisation of order fulfilment and profitability thereby seems very reasonable.

In research by Shen et al. (2006) six requirements were outlined for what they call the "next generation manufacturing systems":

> *R1. Full integration of heterogeneous software and hardware systems within an enterprise, a virtual enterprise, or across a supply chain;*
>
> *R2. Open system architecture to accommodate new subsystems (software or hardware) or dismantle existing subsystems "on the fly";*
>
> *R3. Efficient and effective communication and cooperation among departments within an enterprise and among enterprises;*
>
> *R4. Embodiment of human factors into manufacturing systems;*
>
> *R5. Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments;*
>
> *R6. Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the working environment.* (Shen et al. 2006, p.416)

The design strategies enable these requirements to be fulfilled, as the message exchange platform combined with the strategies do neither inhibit:

- Message exchange to other systems (R1),

[20] Supply Chain Network Design

- Change of agent behaviour (R2) disregarding implementation form,
- Real-time information exchange (R3)
- Inclusion of human factors (R4) including updates to information anywhere in the ABM.
- Quick response (R5), nor,
- Recovery from errors or other deviation from plans (R6).

6.2 The optimisation process

Based on the design choices in the previous section, the example that follows will elaborate the made design choices as follows:

First a departure is made from the conventional approach of modelling supply chain and logistic systems by example using a description from a real world supply chain. This intends to allow the reader to refocus on the information flow in the supply chain and not the physical activities. Once the consequence of this change of focus is presented, the aggregation of problem-classes is described to illustrate how the agent based model deals effectively with redundancy of problem classes through specialisation and message exchange. This detailed description is essential to assure the reader that delays in propagation of information is minimised and that there are no needs for batch-processing anywhere in the architecture.

With this foundation, a detailed description is given of how scheduling can be performed through message parsing in a distributed system. This is supported by examples of how the scheduling process deals effectively with disruptive events and incomplete information.

To help the reader, this section closes with a summarizing overview of the architecture which includes where the human user will interact with the system.

A large scale supply network

Below is Soft Drink Ltd.'s supply chain illustrated as a logistical network of business units. The production facilities purchase raw materials from suppliers – to the far left – and fill the bottles with soft drinks, add glossy product labels and distribute through a network of retail distribution centres, which in turn deliver to retail outlets that sell to consumers (far right).

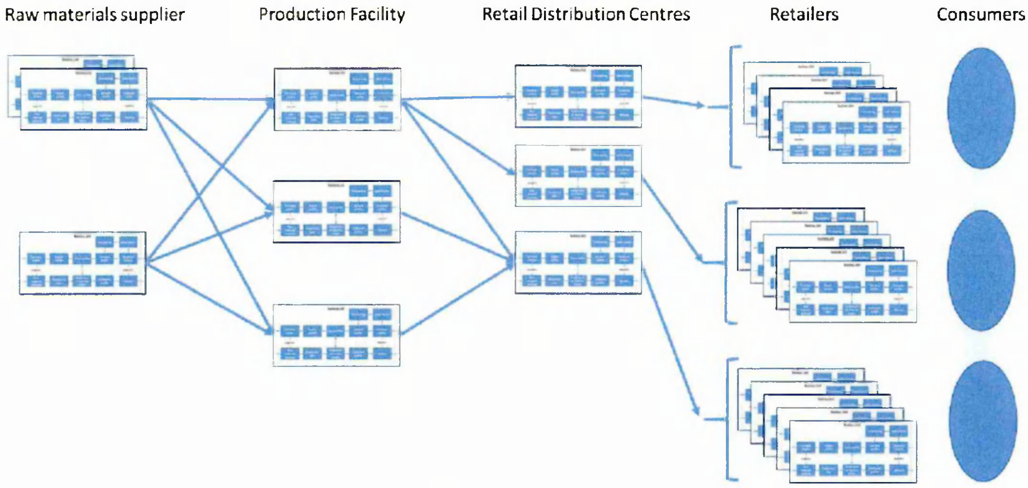


Figure 17 Logistics network of a large FMCG

This is a common way of viewing the problem before attempting to solve the many assignment and allocation problems within this network. This perspective induces a certain bias, in favour of discovering the costs associated with serving the individual consumer, and provides transparency of where the transactions of capital occur. Unfortunately this perspective also conceals the information which triggers the activities which cause the costs.

The illustration below is made to show the flow of information and not the logistics. Here the dominance of one department stands out: The national order management (see below) who has the power to reallocate stock from any plant to any customer.

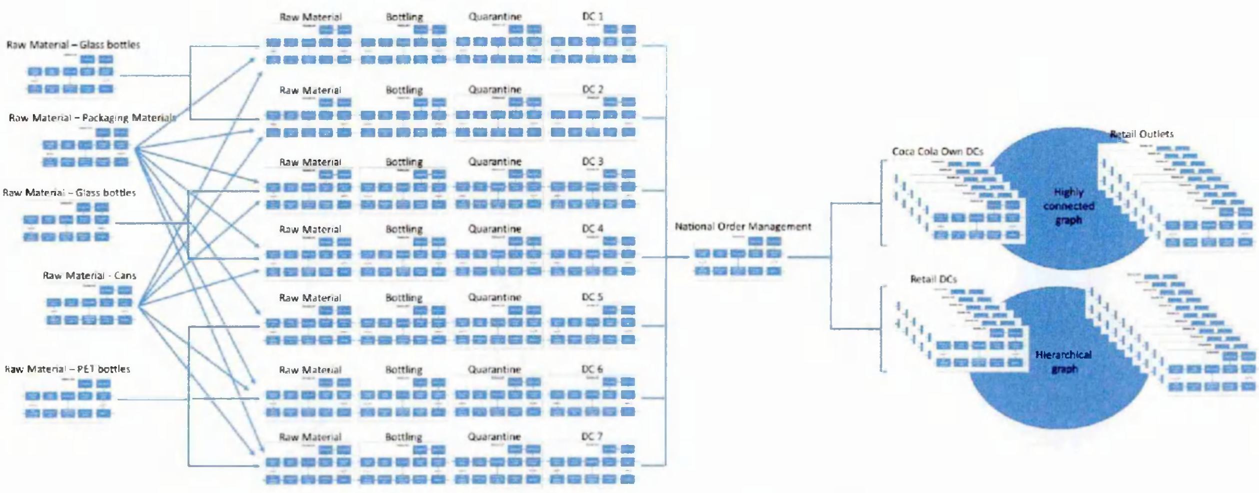


Figure 18 Information of a large scale supply network (zoom in to view details)

The influence that the national order management may exert on the system as a whole is clear as it is a central node – a bottleneck – which determines the flow of information. However, the inverse situation is also valid as any other business unit may influence the decisions that the national order management are making.

This perspective also provides visibility of another thing: That the ability to match supply to the order may be determined by the constraints of every single process.

Even in internal processes within each production plant, where supply and demand is assumed to be deterministic, there is spillage and other disruptive events which result in variation in supply compared to demand. When we therefore zoom in on each node in the supply network, a set of processes repeat themselves at each stage of information processing.

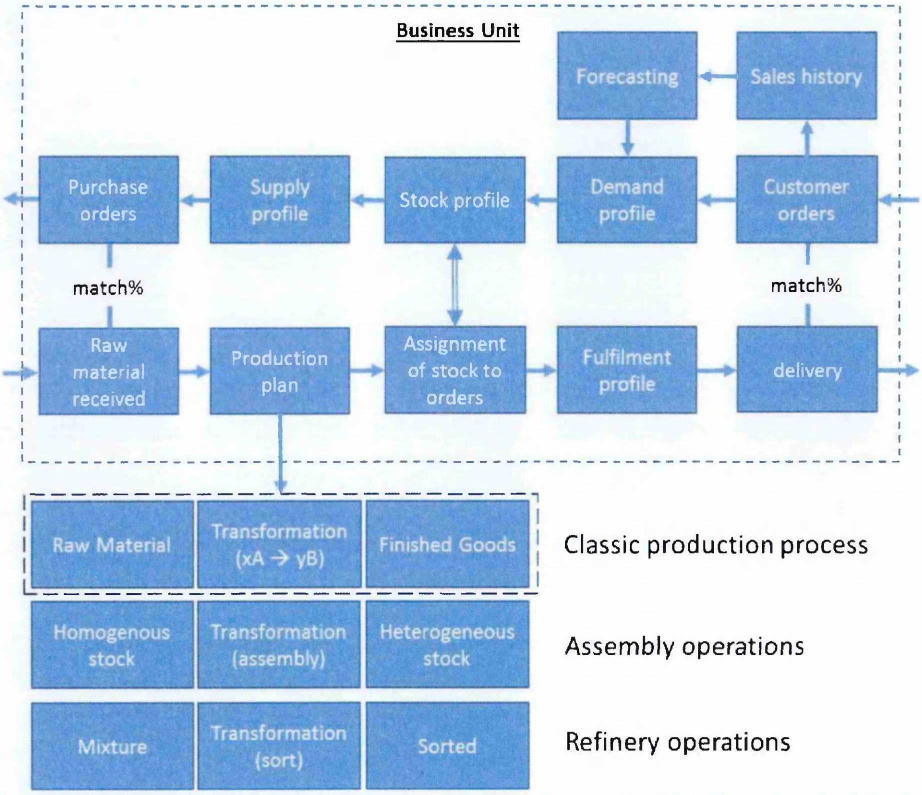


Figure 19 Internal construction of processes in each business unit within the network of the large scale supply network

This model is made to illustrate the repeated nature of each supply node's properties to give a clear presentation that each business unit:

- Records customer orders (internal or external doesn't matter)
- use sales history for forecasting
- Combine forecasts and actual sales to create a schedule for demand
- Maintain a stock profile, which is the schedule of supply and demand
- Calculate a supply schedule for preventing negative stock
- Generate purchase orders to suppliers (internal or external doesn't matter)
- Compare received raw materials from suppliers with purchase orders
- Compare customer orders with actual deliveries as a match % which may be forecasted as a fulfilment profile (schedule).

While these points are simple calculations of sequences of events which may be computed in parallel in more or less trivial fashion, the model illustrates the redundancy of the method which is required to deal with updates: A chain calculation in an array.

In addition, when there is a gap between the fulfilment profile and the demand schedule, which cannot be covered by the existing stock, the assignment of stock to orders becomes an important decision process, which in turn may influence parts of the production plan which is not committed which, again, in turn may influence the purchase orders. These are by category scheduling, consolidation and assignment problems.

The attentive reader will quickly conclude that within a single business unit, there is not much redundancy of computation. However in the supply chain, where the same process is repeated several times, the same services occur:

- Forecasting occurs in all process steps.
- Computation of the demand-, stock- and supply-profile occur in every unit
- Assignment of available stock to orders is also repeated in multiple places.
- Production planning also occurs, both as:
  - *Classic production process* where objects type A are transformed in type B.
  - *Assembly operation* on assembly lines and in distribution centres where large volume homogenous stock is repackaged into quantities as ordered by customers
  - *Refinery operations*, such as in the retail outlets, where the process of goods receipt requires that the heterogeneous – though effectively consolidated load – is unpacked to retail shelves, so that the stock is sorted correctly.

When the colour scheme is changed to highlight the unique computational processes it become easier to see how often the whole system reuses the same type of services across:

- 5 suppliers
- 21 production units
- 7 distribution centres
- 1 national order management
- 21 own DCs
- 81 retail DCs
- 100,000 retail addresses

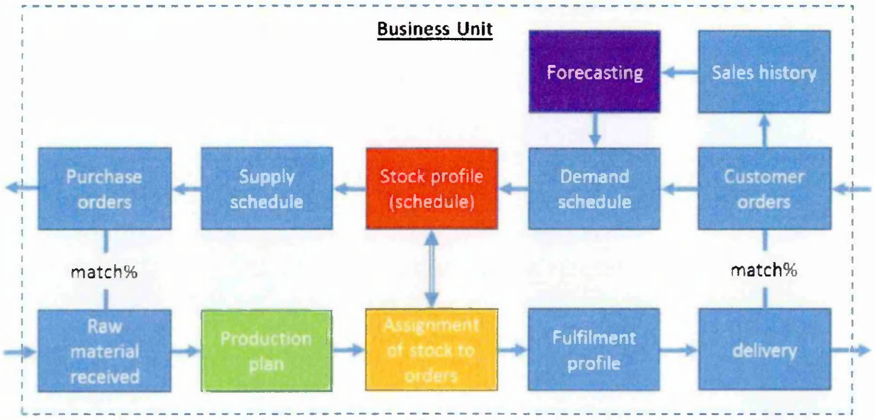


Figure 20 Colour-coding of the processes within the business unit to highlight similar operations

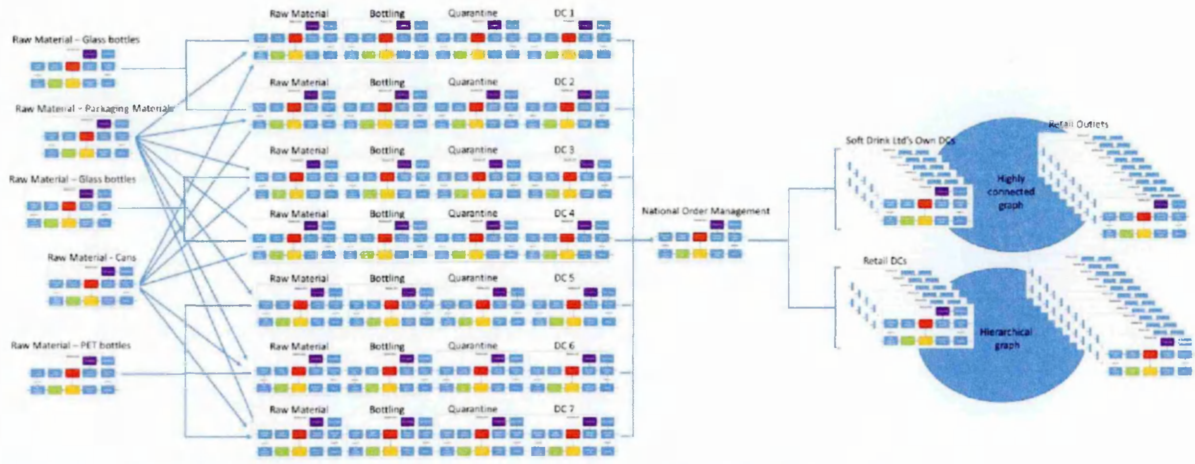


Figure 21 Colour coded overview of the processes which are repeated throughout the supply network.

Ignoring the 100,000 retail addresses, which update only every 24 hours, the example of Soft Drinks Ltd. includes:

Forecasting	...**136 cases of forecasting**. In its simplest form, this is a mere propagation function, but it could also be a tournament of different forecasting models which are quicker to compute concurrently and compare asynchronously as the computation of the concurrent processes complete.
Stock profile (schedule)	...**136 cases of maintaining the stock profile**, which in its simplest form are fast array of operations. However if the stock profile does not fit into computer memory, the operation can effectively be distributed to multiple processors as delta updates. These computations can also be performed concurrently.
Production plan	...136 sites with production planning for 3-5 production lines each results in **544 production plans**. These may be solved as asynchronously as alternating auctions with a message queue. As production planning and routing conceptually is the same class of problems, a vehicle scheduling system could reuse the methods deployed here.
Assignment of stock to orders	...**136** sites where the production output needs to interact with the stock profile and assure fulfilment of orders as **assignment problems**, but with the objective function that pursues profit maximisation and not order fulfilment as the production plan.
Purchase orders	...**136 sites with 8 cases of propagation of changes to quantities**. These operations can also be distributed effectively to multiple processors as delta updates and computed concurrently.

Table 16 Overview of services required in the New Supply Chain Model using the colour coding scheme from earlier.

The system thereby governs collectively 2040 asynchronous computations, which may be tuned effectively using a distributed service model.

To provide an example of how scheduling can be occur as a distributed process the following example will illustrate a scheduling process which is distributed and can handle disruptive events asynchronously.

Scheduling

A message requesting an order to be fulfilled is sent to agent *M2* for the delivery of a set of components {a,b,c,d,e,f,g} as soon as possible. Agent *M2* is dependent on preproduction of the components by Agent *M1*, which in sequel has unconstrained access to raw materials. The message parsing would require 14 steps (illustrated on the next page):

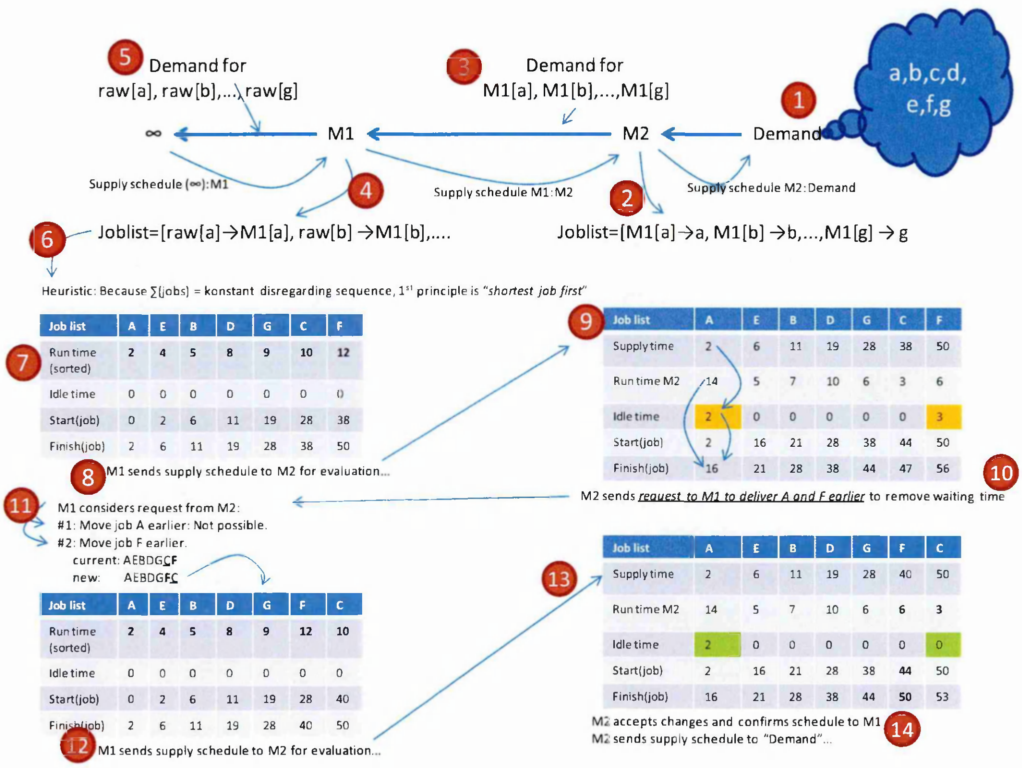


Figure 22 Illustration of message parsing process for a multi agent system with internal state updates

1. The request to fulfil the order {abcdefg} as soon as possible is sent to M2 from the customer.
2. M2 can make the transformation of {abcdefg} using the material {M1a, M1b, ... M1f} so it preliminarily accepts the order, but will respond with a supply schedule to the customer for confirmation.
3. M2 requests M1 to produce {M1a, M1b ... M1f}.
4. As M1 is capable of making the transformation of {M1a, M1b ... M1f} it preliminarily accepts, but will respond with a supply schedule to M2 for confirmation.
5. M1 orders raw material, which in this simple example is confirmed immediately and available to M1's transformation process.
6. M1 transform the orders into a job list using a simple rule: shortest job first. This is done using the job-list used to calculate the supply schedule requested by M2.

7. The supply schedule is sent to M2 for evaluation.
8. M2 takes the supply schedule, provided by M1, and computes the supply schedule to the customer.
9. As there is idle time for jobs {a, f}, M2 requests M1 to move the two jobs associated with {a} and {f} earlier in the supply schedule.
10. M1 considers M2's request, and concludes that {a} cannot be moved earlier, but concludes that {f} can be moved one step up.
11. M1 sends the revised supply schedule to M2.
12. M2 accepts M1's response that moving {a} is not possible, and only recalculates the changes part schedule, which is affected by the change {c, f}.
13. M2 concludes that this is the fastest possible supply schedule as it has the minimum possible idle time, and confirms to M1 and the customer.
14. Done.

To fully appreciate the power of message exchange, this method should be compared to other methods. The solution of a scheduling problem is according to computational complexity theory classified as NP-hard (Cormen *et al.* 2009; Arora & Barak 2009). By their definition, this means that guarantee of optimum requires exhaustive search.

Assuming this is true, the principle of exhaustive search, the permutations of an r-length tuple with all possible orderings is calculated as n = {abcdefgABCDEFG} → $\frac{n!}{(n-k)!} = \frac{14!}{(14-14)!} = 87{,}178{,}291{,}200$. This is a small problem, but if the network grows, the runtime of the batch-processing system grows from exponential (heuristics) to factorial (search). These assumptions are traceable back to Richard Bellman's method for dynamic programming by breaking problems into sub-problems traceable in literature back to Johnson's scheduling method in 1954 (Bellman 1986; Johnson 1954). By comparison, defining the problem as in Rzevski & Skobelev's message parsing approach, the computational complexity grows linearly (sub-linear for parallel processes) as the problem is treated as a distributed problem instead of as a centralized. The attentive reader would notice that "idle time" (in the example above) gives evidence that the most productive sequence has been found, as no further reduction is possible, and may therefore conclude that no further search is needed.

Dealing with disruptive events and incomplete information during scheduling

Previous attempts to solve the problem of allocating resources to achieve objectives have been based on the idea that information often is complete and time to reach the decision is infinite. This line of thought contains some empirical flaws which we have to deal with. Firstly, time to respond is not infinite, so to make the most productive intervention we must be capable to commit to making the intervention at any instance and with incomplete information. Second, time is not reversible, so transactions to which we commit resources cannot be reversed.

If one did not think further about this, it would produce a schedule which commits resources in a random order in which information about events arrive, and though it might happen, in general this will not be the most productive schedule. The assumption is therefore that *at any time during the scheduling process*, the scheduler may be required to commit resources to some part of the schedule.

Using the example above, it is clear that the schedule increases in quality with each message: in Step 2, M2 only accepts the customer order; it is not confirmed. In Step 9, the first schedule is in place, and if no more time was available before the work should be started, the first job for this initial schedule *could* be committed by M1. This is on the assumption that M1 is initiating the work on job A based on incomplete information somewhere between Steps 9 and 10: the fact that the message from M2 in Step 10 arrives after M1 has initiated the work on job A is not a problem, as M1 still may communicate back that it cannot move job A as it is already committed, but jobs F and C may still shift position in the sequence. Hereby, M1 copes with the disruptive event committed by M2 by only changing the part of the schedule that is not committed.

M2 can therefore decide whether it can or cannot fulfil the customer order from the moment where its supply schedule from M1 is available, but it - M2 - may defer to respond to questions about, for example, costs or delivery time, until it concludes that its schedule can no longer be improved. In this way *maximum delayed commitment* allows a long chain of activities to collaboratively determine the most competitive offer to the customer's interface before communicating.

Even in the case where the customer might ask two suppliers – M2 and M2's competitor, M2C – to obtain a price comparison as a part of the planning process, all that the customer needs to tell M2 is *when* it wants a committed answer in terms of time and cost. M2 may then defer answering until the deadline given by the customer. This is quite normal for public processes such as bidding for funding, public tenders and even auctions on eBay. However only (Rzevski 2011; Rzevski & Skobelev 2014) included the component of *maximum delayed commitment* in their design of multi-agent systems.

Architectural summary

As hinted in Table 16 *Overview of services required in the New Supply Chain Model* the *virtual world* which describes the known supply chain optimisation problem of Soft Drinks Ltd. will be divided into specialised service. The benefit hereof is that the MAS does not need to be a single memory block as typical for openly available agent based frameworks. The link that connects the service is a high performance message broker (see below) which assures communication between Resource-Demand Networks (RDN's). Thereby there are no requirements of where (physically) each service runs. However, it would be wise to assure that message exchange between sub-problems that are tightly linked travels shortest possible physical distance, as changes to information in highly dependent problems also will require more messages to be exchanged when the revised optimum needs to be determined.

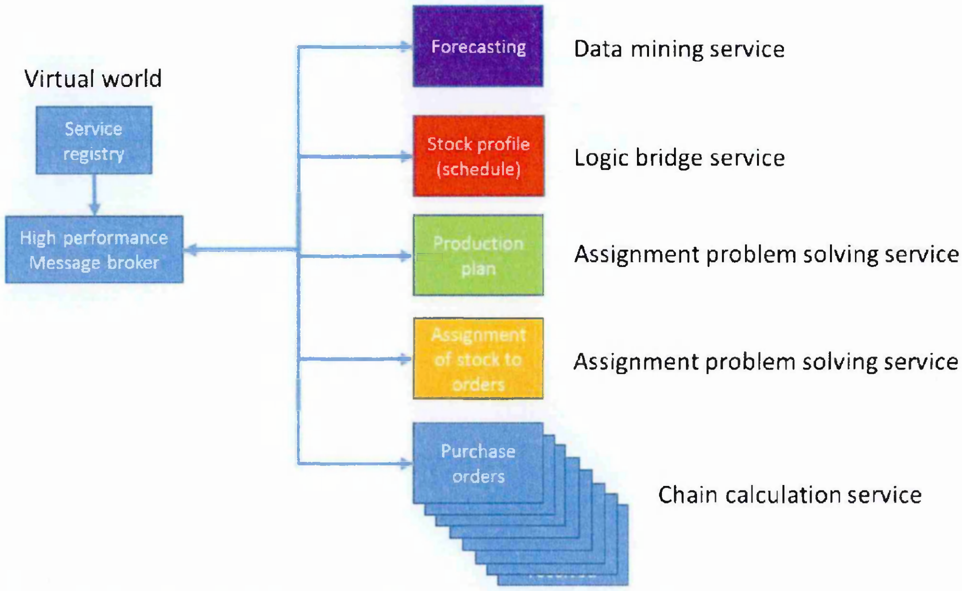


Figure 23 The internal grouping of micro-services which is used to construct the New Supply Chain Model.

In addition to the description above, three practical design features need to be added to the New Supply Chain Model.

1. First the user which need to interact with information held by a service.
2. Second questions of authentication
3. Thirdly the role of persistency of data, knowing that machines fail.

The illustration (Figure 24) provides an overview the network of services which provide a functionally sufficient model for deployment in industrial context.

The user's entry point is referred to as a reporting service. To prevent the reader from misunderstanding this concept, an analogy is convenient: When shopping online on for example amazon.com the user is viewing a report of what is in store. Literally Amazon.com is not a shop. It is a website which reports what services the supply chain of amazon plc can do for the web user. The user may buy products, update delivery schedule and performed other transactions of information through web based forms. Fundamentally the website remains a report with interactive features. The same applies for the New Supply Chain Models simulation of, for example, the consequence of change of transportation rates: The user interacts with a report that is rendered for the web-browser. For consistent usage of terms, the point of interaction for the user is a reporting service and not a UI. This decoupling of concepts should make it clear to the reader that the New Supply Chain Model allows any user interface to interpret the information made available through the reporting service. Other operations, such as peer-to-peer communication to external system are handled through a system-to-system communications interface.

The next component is authentication, which must be embedded in any transaction of information through a hierarchy of permissions. This is trivial but important as the system will be connected to untrusted systems.

Finally comes the data storage. The usage of a storage service allows the specific storage system to be chosen without intervening with the New Supply Chain

Model. This is done by letting storage service subscribe to messages transmitted over the high performance message broker. Depending on purpose and usage, the storage device can filter relevant and irrelevant information.

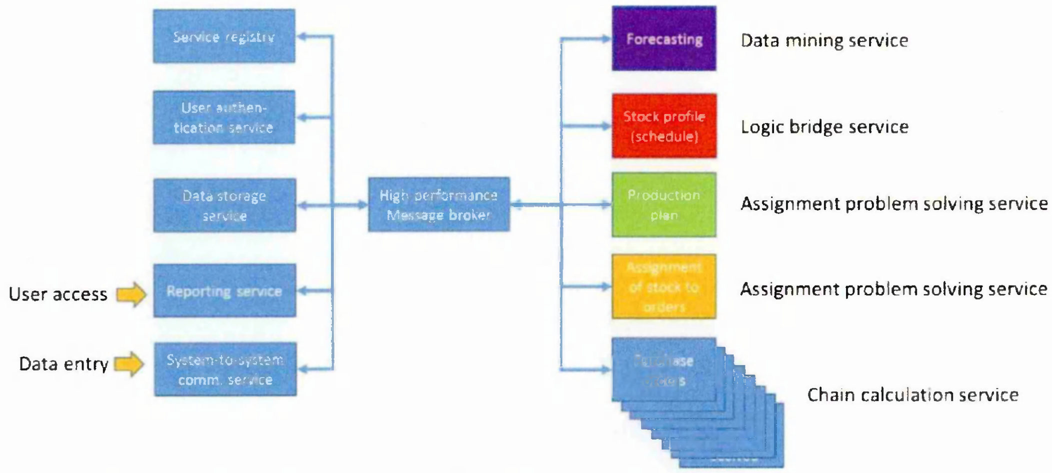


Figure 24 The complete ABM including user and system interfaces (to databases)

A system administrator may provide a new "solution" by creating a script that connects between different types of services. The Soft Drinks Ltd. example is just a message parsing network which:

- Receives data in the system-to-system communication service (source)
- Populates the services with information in agents (optimisation) and stores in received information using the storage service (for persistency)
- Delivers the computed results in the reporting service which the user may access (sink)

In the book ***managing complexity*** (Rzevski & Skobelev 2014) the virtual world is illustrated as an abstract environment in which the lifecycle of the agents elapse. With this architecture, the virtual world is a network of services, with the benefit of this is that it scales, reuses the code base and remains transparent.

The New Supply Chain Model is thereby characterised by:

- Distributed decision making
- Real time decision making
- Maximum delayed commitment
- Only updates resource allocations affected by changes
- Solves optimisation problems by exchange of messages rather than by computation
- Is batch-free
- Minimises delay amongst people by transferring human preferences (customer satisfaction & profitability) to agents which vigilantly maintain optimality.
- Permits human override of system allocations
- Permits simulation as decision support in parallel with handling live transactions.

With these design criteria for software development, the most productive method of intervention in a complex economic system is demonstrated using the following example.

6.3 Testing

A major challenge for researchers is to verify models of complex system behaviour. This is well known from issues with debugging multithreaded programs where indeterminacy races occur (Herlihy & Shavit 2012). This is caused by the operating systems switching between threads which causes small variances in the sequence of messages exchanged amongst swarms of agents leading to different paths of convergence. However in contrast to sequential processing of information, the distributed asynchronous information processing that is used in an agent based system requires that attempts to evaluate intermediate state of the computation is disregarded. Attempts to trace the convergence of the state of the system, is much more productive. This leads to a set of tests (see Figure 25, below) which assure consistency of transactions, predictability of run-time, rigorous treatment of micro economics and a consistent quality of schedule.

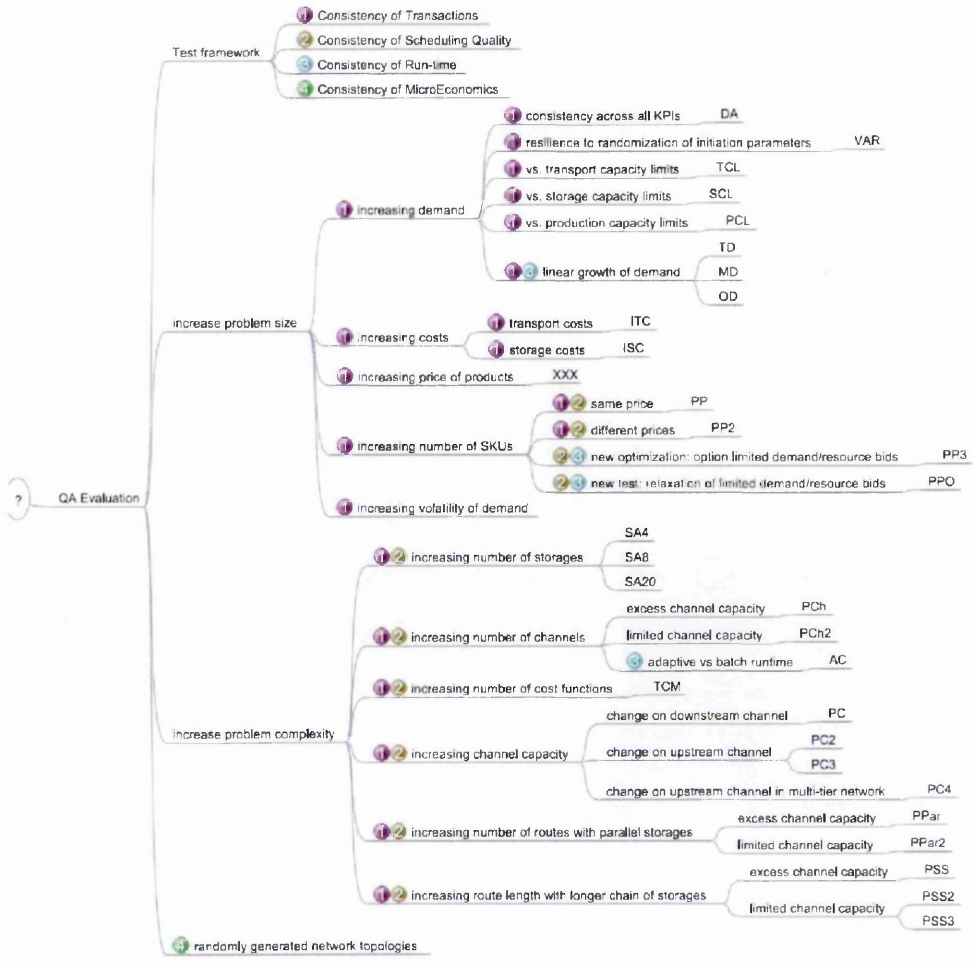


Figure 25 Test framework used to assure the quality of the implementation of the New Supply Chain Model

The examples where a part of a wider quality assurance program that was created to validate the implementation of the New Supply Chain Model. As the test suite contained more than 3658 tests (A.2 Test program (extension)), each varying a single variable, a detailed explanation of each test is senseless. That being said, it shall also be mentioned that the automatic test-suite contributed to the discovery of more than 100 implementation errors which each influenced consistency of results, variation in runtime or correctness of results.

The challenge is therefore to assure with consistency of transaction and convergence, and to a lesser degree worry about corrupted messages, byte order, and other low level error corrections as any modern operating systems deals with this.

6.4 Considerations for implementation

Designing distributed systems is not in the mainstream education. Technical challenges persist which often deceives developers of ABM. Wooldridge & Jennings (1998) have performed an analysis of a series of agents based models which the author finds only far too familiar and should appear as a notion warning to other researchers:

Perspective taken in this thesis	(Wooldridge & Jennings 1998)
Understanding of ABM - The intuitive understanding of ABM often guide people to analyse and discover new ways of solving their target problem. However as Polya (1945) describes: The main fallacy [of mathematicians] is to believe that they know how to solve a problem just because they recognise features of the problem from another domain. To their surprise, failures often appear as they prepare the proof of their logic. The same applies to adopters ABM: Their initial excitement with distributed methods for solving problems, deceives them into belief that they understand it, until they test their logic in practice. Woolridge & Jennings highlight a particular dangerous symptom which is when designers believe that all they have to do is to "put agents in a melting pot" from which solutions magically will appear.	2.1 You oversell agents 2.2 Getting religious or dogmatic about agents 3.1 You don't know why you want agents 3.2 You don't know what your agents are good for 4.2 You confuse buzzwords with concepts 7.1 You see agents everywhere 7.5 Your system is anarchic
Understanding of the problems domain - Any solution written as software must be highly specialised to a particular category of problems. When developers' approach the problem without in-depth understanding and hope to achieve understanding of the domain through iterative modelling using ABM, the many iterations (and many consequent layers of code) will result in a an ABM that is unmaintainable and thereby unsuitable for industrial usage. To overcome this problem a team is required, as the skills needed extend beyond what can be expected of any individual.	3.3 You want to build generic solutions to one-off problems 3.4 You confuse prototypes with systems 6.1 You decide you want your own agent architecture 6.2 You think your architecture is generic

Understanding of the software development process - At the other end of the range of spectrum is development performed by domain experts who think they know how to write code. The leads to a long list of software engineering challenges which again produces unmaintainable codebase (Goldstone et al. 1985; Brooks 1995). The response once more is that only a team with a range of skills can therefore overcome this problem.	4.3 You forget you are developing software 4.4 You forget you are developing distributed software 5.1 You don't exploit related technology 5.2 Your design doesn't exploit concurrency 7.6 You confuse simulated with real parallelism
Understanding of transformation of information using agents - Amongst the teams who develop agent based models, the most problematic is those who develop large complicated agents. The flow of information becomes disrupted, particular agents become bottlenecks and team becomes disillusioned with the performance gain that was pursued with parallelism and message parsing. Piping data in functional style has proven itself as the approach which assures the highest throughput. Decoupling and grouping of sub-problems in the agent based model is thereby the most powerful method to assure high throughput and minimise latency. Complimentary to Jennings & Woolridge, I would argue that intelligence should be pursued at the level of the swarm and not the agent. Having many agents is not a problem, as long as the swarm is managed effectively.	6.3 Your agents use too much AI 6.4 Your agents have no intelligence 7.2 You have too many agents 7.3 You have too few agents
Understanding of the deployment environment - Since 1998 a lot has happened in the IT sector. In particular computing platforms have emerged which allow users to launch virtual high performance clusters with 20-25 minutes of education[21]. To attempt to build a distributed high performance agent based system from scratch would effectively be a waste of resources, as standards exist and provide a suitable canvas for automated management of large numbers of heterogeneous servers.	8.1 The tabula rasa 8.2 You ignore de facto standards

Table 17 Comparison of the authors experiences with Wooldridge & Jennings (1998)

In addition to these notes, a particular focus should be directed towards (Wooldridge & Jennings 1998): "4.4 You forget you are developing distributed software" which seems to repeat itself in all projects which are led by software developers and not domain experts:

[21] http://star.mit.edu/cluster/

> Distributed systems have long been recognised as one of the most complex classes of computer system to design and implement. A great deal of research effort has been devoted to understanding this complexity, and to developing formalisms and tools that enable a developer to manage it [2]. Despite this research effort, the problems inherent in developing distributed systems can in no way be regarded as solved. Multi-agent systems tend, by their very nature, to be distributed — the idea of a centralised multi-agent system is an oxymoron. So, in building a multi-agent system, it is vital not to ignore the lessons learned from the distributed systems community — the problems of distribution do not go away, just because a system is agent-based. (Wooldridge & Jennings 1998, p.4)

To give provide an annum 2014 extension to the warnings in the 1998 paper, a set of detailed discussions follows as guidelines for the development of distributed systems. At the end of each discussion one or more "rules" are given as advice for development.

Communication amongst applications

There are plenty of options for communication amongst applications. Within the each swarm the most effective method is simply to update an object (message queue) for each swarm. Across swarms, sockets and cores, but within the same box, the most effective method is to add the message to the message queue of the swarm class. Between boxes the two most widely deployed options are ZeroMQ. Some argue that implementations of MPI, such as Beowulf clusters, MPICH and LAM MPI, are suitable too, but MPI is designed for "parallel computing" on a fast, reliable networks and not "distributed computing". MPI thereby make good sense on a cluster, but not for a distributed application. Given the current GPGPU development it could even be argued that MPI is about to be substituted with CUDA or openGL as the graphics cards provide more bandwidth (14Gb/s vs. 1 Gbit/s) than MPI and better economy (flop/joule). ZeroMQ – developed by Pieter Hintjens and maintained by iMatix – was made for distributed systems. A pseudo MPI example using ZeroMQ which handles 1.1 million messages per second is given below:

```
import sys, zmq, time
from  multiprocessing import Process

def worker(n):
    context = zmq.Context()
    work_receiver = context.socket(zmq.PULL)
    work_receiver.connect("tcp://127.0.0.1:5557")
    for task_nbr in range(n):
        message = work_receiver.recv()
    sys.exit(1)

def ventilator(n):
    Process(Target = worker, args=(n)).start()
    context = zmq.Context()
    ventilator_send = context.socket(zmq.PUSH)
    ventilator_send.bind("tcp://127.0.0.1:5557")
    for num in range(n):
        ventilator_send.send("MESSAGE")
```

```
if __name__ == "__main__":
    testsize = 10**6
    start_time = time.time()
    ventilator(testsize)
    end_time = time.time()
    duration = end_time - start_time
    msg_per_sec = testsize / duration

    print "Messages Per Second: %s" % msg_per_sec
```

```
$ python test.py
Messages Per Second: 1081782.78293
```

Table 18 Sample Python code (test.py) illustrating the throughput of ZeroMQ as a high performance message broker

A research team headed up by Andrzej Dworak evaluated ZeroMQ for usage at the LHC at CERN and concluded, that it was the only message queue that scaled reliably[22]. ZeroMQ was chosen based on:

- Easy to trace peer-to-peer communication with reliable request/reply and publish/subscribe messaging patterns.
- Synchronous and asynchronous/non-blocking communication.
- Quality of Service (QoS): timeout management, message queues and priorities, various thread management policies.
- Small library size, low memory and resource usage.

With ZeroMQ being suitable for distributed systems (rather than parallel), a design challenge is how to organise the communication? Pieter Hintjens research clarified a set of dilemmas illustrated below[23]:

Option (A) represents a typical messaging system with a messaging server ("broker") in the middle. This results in a set of basic advantages – such as transparency and decoupling – which are useful when re-engineering a monolithic application into services, and, the broker gives a point of reference if the application is prone to failure as the messages will be retained. However the communication footprint is excessive and the broker may become the bottleneck of the system. Option (B) reflects a pipelined alternative that departs from the SOA model. To get more effective than (B), the broker will need to be removed (option (C)). Whilst option (C) achieves the lowest latency and permits the highest transaction rate, a management system needs to be in place as each application needs to know where the applications are that it must connect to.

[22] International Conference on Computing in High Energy and Nuclear Physics 2012 (CHEP2012) IOP Publishing
Journal of Physics: Conference Series 396 (2012) 012017 doi:10.1088/1742-6596/396/1/012017 [link]

[23] http://zeromq.org/whitepapers:brokerless

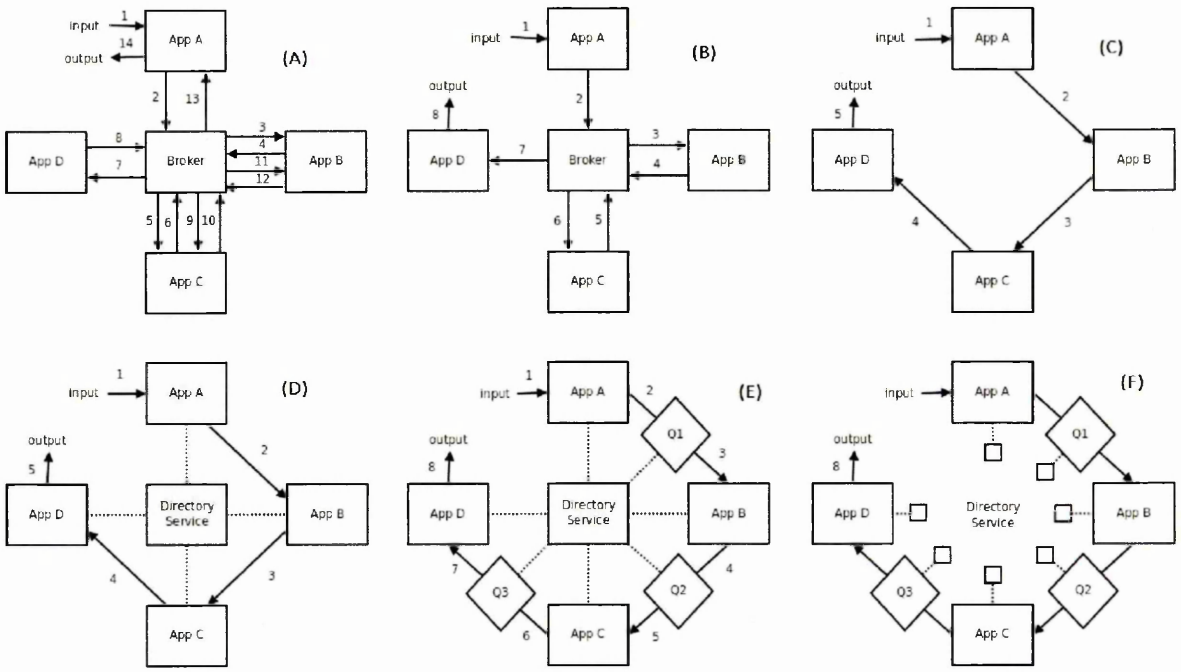


Figure 26 Communication steps under different layouts for a distributed system supported by zeromq.org

Option (D) uses the broker's repository of applications on the network and divides the functionality into two parts: The broker acts as a directory service for the applications, and the applications query the broker to learn where the other applications are, so that the communication can happen directly. Option (D) is thereby suitable as long as no messages are ever lost. Option (E) avoids to have the broker as a bottleneck, by providing a distributed broker, where each message queue is implemented as a separate application that is registered with the directory service (broker). To avoid a single point of failure, option (F) suggests a distributed directory service, where the configuration is copied to all nodes in the network. As the networking topology changes, the configuration may be updated.

Though it may appear trivial the first rule for the design process is:

Rule #1: Expect a distributed design from the beginning.

Key distributed optimisation principles

Literature on Agent Based Modeling typically distinguish ABM from Object Oriented by claiming that agents are autonomous entities. Unfortunately a computer cannot thread or multi-process with millions of agents each having their own thread. Therefore the Multi-Agent System (MAS) needs to assign agents to threads. To avoid cross-socket access to memory objects, the MAS is assigned to memory objects assign to threads by problem type: Hereby agents who need to communicate a lot are close to one-another. For inter-thread communication queues are used. For inter-box communication messages are exchanged over TCP.

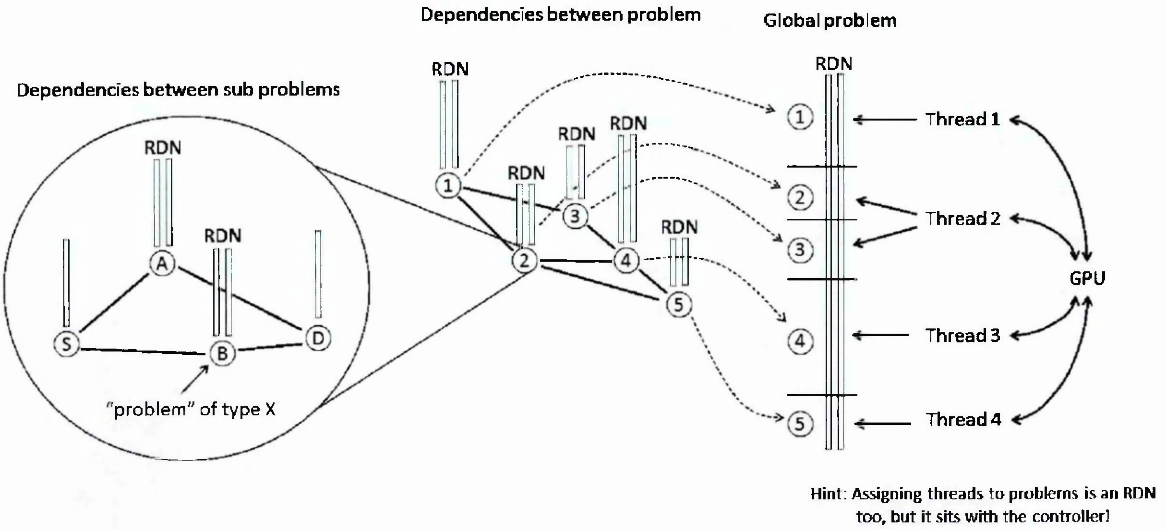


Figure 27 Dependencies between RDNs as network and assignment of RDNs to threads

In the RDNs requests for resources or offers to demands are implemented using Coroutines. This avoids context switching and requires a lot of evaluations (Single instruction, multiple data - SIMD), whereby GPUs are well suited. As the memory that is evaluated is thread local, there is no need to protect the register using mutual thread exclusion (mutex[24]). The operation is bound to messages dispatched by the coroutines. Hereby the MAS exploits the CPUs ability for memory management, whilst exploiting the GPUs ability for numerical operations.

Rule #2: Avoid cross socket queries by assigning memory to be thread local.

Rule #3: Expect by design that two agents never will be in the same threads memory pool.

Rule #4: Expect swarms will need load-balancing to prevent that any keep hyper threads becomes a bottleneck and that as a result agents will be moving around.

Key asynchronous update principles

Events are imported from the message queue managed by each thread. In the first instance the event is added to the RDN as either an offer of a resource or demand for a resource in the RDNs "current state" as an unsatisfied agent, and a copy of its pointer in the RDNs "next state". The RDNs thread then propagates the new events offer as a request evaluation in a directed acyclic graph which is maintained in the RDNs "next state". As this chain reaction elapses, thousands of messages may be sent to the GPU. As responses are returned, the coroutines, complete and the new solution emerge in the "next state". The thread then patches the "current state" (previous solution) with the completed solution from the "next state". The thread then deletes the content from the "next state" memory object to prevent that time needs to be spend on garbage collection.

Simulations of schedules for supply chains several years will contain 10^6 - 10^9 objects and as RAM unfortunately is not infinite a mitigation method is needed.

[24] Mutex is defined by Herlihy & Shavit 2012.

Fortunately information in a supply chain has a limited planning lifecycle: Resources can only be planned for as long as they have not been committed to demands. In the optimisation process it is therefore distinguished whether a set in the RDN are historic or that external events now claims the resource irreversibly. When this happens the thread, sends a signal to a database that the transaction is irreversible and upon confirmation, it removes the committed RDN-set from memory. Hereby "planned" data is in memory, whilst "committed" is dropped to a database.

Rule #5: Expect that computation at write time is faster than read-time

Rule #6: Expect that thread local patching must be possible without overhead of locks and mutex.

The benefit of the swarms

Systems that are suitable internet scale applications that are based on MAS, have many (thousands) of simple agents (bytes). Systems with a few, relatively large agents (several kilobytes) do commonly not scale efficiently: The reason is that developers loose overview of the agents as the code base grows, together with a reduction in transparency of how the information is used within the agent. A better approach is to create an agent as a proxy for the agent that exhibited a growing memory footprint, and start to decouple the agent internally to become a swarm of multiple agents. This brings us to agents which are stateless functional programs:

CPUs are good at memory management. GPGPUs are excellent for stateless or functional transformation of information and can run thousands of threads at low overhead. NVIDIA has produced some excellent results with JIT-compilers for GPU for particular problem classes and illustrated reduction in costs from $ 259.00 per core to $ 1.00 per core[25]. The research is in its infancy on how to load-balance computation between CPU and GPU computation in large scale connected systems, however a logical step would be to perform the dispatch to GPU at the level of swarms, whereby the shared memory on the Graphics Card can perform concurrent operations at far greater efficiency (20 GFLOPS/watt) than the CPUs. By designing each Swarm may have its own dictionary of sub-swarms, the system may, as a while scale with the same efficiency as for example the DNS-system, utilise the GPUs effectively whilst the CPU only performs memory management.

---

[25] Based on http://www.hpcwire.com/off-the-wire/high-performance-computing-modernization-program-adds-capability/ where the US DOD has spent $150million for 577,000 compute cores in 2014 vs. Amazon's off-the-shelf price of $4,999.00 for 4992 CUDA cores for Nvidia Tesla K80 24GB GPU Accelerator

1. Launch the control service to all available hyper threads (locally and remotely) to await tasks
2. Calculate assignment of agent services to threads, then send message to threads to import the specific agent services
3. Send messages to each agent services to request data to populate agents
4. Initiate message broker on each agent service
5. While running:
    a. Let agents communicate asynchronous using Coroutines.
    b. Call for reports by sending messages directly to agent services
    c. If new event: Instruct the agent service to import event.

Figure 28 Outline of the start-up of an multi-agent system

Some projects have attempted to develop learning capabilities within the agents. A reoccurring example is seen in agents which use divide and conquer methods on a frozen dataset. But this is naïve because at internet scale the programmer should not expect that the dataset can be frozen, in addition there are limits to how much the individual agent can sense and thereby learn. Systems which have collective learning capabilities, at the level of the swarm – such as ant colony models for example – outperform systems with agent specific learning by orders of magnitude. The evidence that the ability to share knowledge within the swarm and exploit convergence of behaviour of small – not so clever agents – is abundant, though not adopted widely. The reasons why is unclear at this moment in time, though it may be due to questions of message exchange which we will return to.

Rule #7: Break large agents into functional-program type agents to assure optimal performance

Use Co-routines

In distributed computation, agent based systems should always use Coroutines[26], which generalize subroutines for non-pre-emptive multi-tasking, as multiple entry points are created to permit suspension and resume of execution. Coroutines have proven excellent for cooperative task-handling, events loops, iterators, infinite lists and pipes. Scientific computing has already adopted this method but industry seems to be lagging behind. By designing the MAS so that multi-tasking happens at the level of the agent with suspension and resume of execution using Coroutines, the agents within the swarm can communication asynchronously as well as between swarms – even if the swarms are on different socket. Context switching is also completely avoided.

[26] https://en.wikipedia.org/wiki/Coroutine

```
class agent(object):
    class_knowledge = {} # Knowledge shared within the
class

    def __init__(self, id):
        self.id
        self.knowledge = {} # Knowledge held by the
individual agent
        self.messages = []

    def send(self, self.id, to_agent, message):
        to_agent.messages.append(tuple(self.id, message))

    def respond(self, message):
        try:
            message[0].send(knowledge[message[1]])
        except KeyError:
            message[0].send(class_knowledge[message[1]])
```

Figure 29 A sample class swarm template for an asynchronous message based multi-agent system.

A legacy assumption is use a single message queue where threads obtain lock, read and remove a message, and release the lock, before searching through a context and whilst applying principles of functional programming for updating the context based on the processed message content. This is a slow process, which can only be made worse by forcing the kernel to perform context switching if the number of messages exceed the number of hyper threads. A much better method is to keep the agents registered in a dictionary with pointers to their objects. The dictionary access is hashed which means constant-time access ($O(k)$) and the objects can be based on an empty shell which can be updated at run-time. This allows for maximum flexibility at a minimal overhead. Even if such a dictionary turns out to be extremely large the usage of an agent to manage the local and remote memory pointers can become an effective solution.

Rule #8: Use coroutines to suspending and resume execution within the program.

Use scripting

Some academics argue that system written in scripted languages are slow. However a critical distinction should be made: General purpose programming languages such as for example Python or Julia is not a runtime[27]. All runtime have their own performance characteristics, and none of them are slow. A more categorical error is to believe performance assessments are assigned to a programming language. Always assess an application runtime, most preferably against a particular use case.

For those who insist that certain runtimes still are slow, several translation packages allow compilation of scripted languages to optimised c.

Given the developer is very skilled, systems written in c or c++ may be faster, but that does not help when the system needs to evolve, as the lack of access to skilled

[27] https://www.paypal-engineering.com/2014/12/10/10-myths-of-enterprise-python/

developers may be limited. Systems which are scripted or written in high level languages also tend to go through more iterations of development and testing and thereby evolve faster.

Tools – like coverage and cprofile – that help to determine (1) how fast is the runtime, (2) where the bottlenecks are, (3) how much memory it is using, and, (4) where is memory leaking – increase developer productivity by reducing the time spend on extending the system. It should be added that in a properly designed MAS, the cprofile will point out exactly which agent functions that are the bottlenecks. An excellent case of API design based on profiling is provided by Jack Diederich[28] at PyCon-2012, where he presented the process of rewriting 120 classes to a single Python function.

Another component which developers often forget, is not just to performance profile the test cases but also the deployment. The bandwidth and runtime of self-replication of a full VM is a lot longer than moving a Python or Julia script. The lesson: Tailor the runtime selection to a minimum of libraries, and use code coverage profiling to assure that unnecessary accessories are not consuming bandwidth and RAM.

Rule #9: Use scripting as long as much as possible.

Rule #10: If a certain part is too slow use runtime profiling on real user data to determine the bottleneck. Avoid synthetic cases.

Design on small systems

40 years ago there was no high-end hardware by today's standards. The system understanding and creativity of developers was tested every day just to make the most basic applications run effectively. For example programmers in the current era rarely worry about garbage collection, and hence make heavy use of garbage collection, which will slow large systems down as around 30% of the time computational time will be spent on garbage collection. It is therefore poor coding practice to design applications which repeatedly creates and destroys memory objects. A much better approach is to reuse objects instead of creating new ones using object pools which may act as general memory containers. Attention to real-world hardware constraints which occur when working on low-end hardware thereby forces the developer to consider exception-handling, such as memory overflow, during the design phase. Whilst the philosophy of developing on low end hardware is still to be appreciated, the focus has been on developing applications which present the right logic, passes tests and scales on the available hardware. Given this narrow scope on locality and less on computation as interaction, it is possible that the absence of understanding of communication models in complex systems is what inhibits the emergence of systems that are scalable to millions of users.

Rule #11: Update objects so that dependency on the garbage collector is minimised.

---

[28] www.youtube.com/watch?v=o9pEzgHorH0

Rule #12: Develop software to run on low-end hardware as this will reveal problems earlier in the development process.

Complementary to Wooldridge & Jennings (1998) the rules for software development may be summarised as guidelines for agents based modelling.

Rule #1: Start with a distributed design from the beginning. Rule #2: Avoid cross socket queries by assigning memory objects to be thread-local. Rule #3: Expect by design that two agents never will be in the same threads memory pool. Rule #4: Expect swarms will need load-balancing to prevent that any keep hyper threads becomes a bottleneck and that as a result agents will be moving around. Rule #5: Expect that computation at write time is faster than read-time Rule #6: Expect that thread local patching must be possible without overhead of locks and mutex. Rule #7: Break large agents into functional-program type agents to assure optimal performance Rule #8: Use coroutines to suspending and resume execution within the program. Rule #9: Use scripting as long as much as possible. Rule #10: If a certain part is too slow use runtime profiling on real-use data to determine the bottleneck. Avoid synthetic cases. Rule #11: Update objects so that dependency on the garbage collector is minimised. Rule #12: Develop software to run on low-end hardware as this will reveal problems earlier in the development process.

Table 19 list of Guidelines for Agent Based Development

6.5 Summary

This chapter formulated the New Supply Chain Model as a synthesis of *Chapter 5 – Case Study I: LEGO* and a critical review of the thinking in supply chain modelling and agent based system design.

The system description considered where information is produced and what the requirements to the interfaces between real and virtual world should consider. Particular attention was given to the elements which account for any delay in propagation of information and a systematic evaluation of what the consequence is to prevent any need for batch-processing. This included a critical analysis of the expectations of for how long information can be expected to remain up-to-date and which mitigation methods minimises any unavoidable delay.

With this conceptual background the optimisation process was analysed to develop the New Supply Chain Model in a manner that is scalable to large systems from the beginning. This meant that an explicit departure was taken from following the logistics processes and instead model the information flow which determines which activities should be committed to the physical processes. Particular attention was paid to assure that the process did not require any batch-processing of information and detailed example of a distributed scheduling process was given with a view to deal effectively with disruptive events and incomplete information. This was supported by an architecture summary which illustrated how storage services, authentication, and people & remote systems would interact with the designed system.

The tests used during development (test-driven development) were briefly described followed by detailed considerations for implementation, which highlight twelve "rules" for the development process.

The next two chapters will provide detailed examples of how the New Supply Chain Model was used to evaluate the consequence of delay of information in the network batch-information processors.

Chapter 7 – Case Study II: Real-time Retail (LBR)

Introduction – This case study is based on work done by the author as an employee of LEGO with access to the necessary confidential data. The analysis used the software designed by the author who managed the data and its analysis.

The case study including the design and development of the multi agent solution for LEGOs Branded Retail Outlets (LBR) was published in (Madsen et al. 2012) as "Real-time Multi-Agent Forecasting & Replenishment Solution for LEGOs Branded Retail Outlets" which was accepted by LEGO and selected as best paper at the IEEE's SNPD Conference in Kyoto. The papers 1st author is the author of this thesis.

This chapter is focused on the details of the case design & implementation and will take a more detailed view of how the results are produced.

1. First, the case study will describe the original focus and methods used by the team responsible for the inventory at LBR.
2. Second, the case study will describe how the manual methods where automated and how that provide the required evidence to create the *LBR base case*.
3. Third, the processes of the *LBR base case* are analysed in lieu of the previous chapters to provide an outline of "how to make the most productive intervention" and how this relates to the design of the experiments.
4. Fourth, the New Supply Chain Model is deployed to simulate the impact of making the chosen intervention and to evaluate the results of the "improved situation".
5. Fifth and finally, the feedback from the organisation is summarised to highlight human/organisational aspects of taking the New Supply Chain Model into operation.

7.1 LBR using Excel manually

Before the case study was initiated LEGOs Branded Retail Outlets (LBR) were supported by an inventory management team. This team's job was to assure that the shelves of the LBR outlets were filled with the right product in the right quantity and never ran out of stock. Logistics cost were ignored and costs of operating the stores was out of scope. Methods for collaborating with the carrier or warehouse were considered beyond the teams' responsibility.

The urgency for change was established as a financial review concluded that LBR was performing below the standards for its industry. First, it was concluded to involve senior logistic consultant with experience in retail collaboration: The author. Second it was concluded to establish a quick best-practice amongst the inventory management team, to reduce the variation in the management of replenishment orders.

The "current practice" was based on downloading data from LBRs ERP system: SAP ECC 6.0. Then manipulate the data in a spreadsheet to create replenishment orders followed by copy-paste entry of orders in SAPs order entry form. This time consuming process was augmented as one of the employees created a template in Excel which was used by all members of the inventory team. The spreadsheet grew quickly to consider all the many "quirks" that each employee applied in good faith, but ultimately made the decision making methods in-transparent. As the spreadsheet furthermore was formula based, the "best practice template" had to contain more formulas than needed in order to cope with the variation in order lengths. This excess of formulas put the spreadsheet software to its limits causing it to crash at random times, which caused rework.

The "supply chain model" used by the inventory management team was thereby constrained to the following features:

1. Between Monday and Thursday the inventory management team created replenishment orders as follows:
   1.1. Pick the outlet with highest revenue, for which no replenishment order has been created.
   1.2. Download point of sales data from SAP (POS data)
   1.3. Download inventory position from SAP (inventory)
   1.4. Calculate a linear trend line of projected demand based on POS data
   1.5. Create replenishment orders so that inventory + replenishment orders would cover the current and 3 following weeks of projected sales.
   1.6. Revisit emails from LBR store managers complaining about shortage/excess stock and adjustments the replenishment order.
   1.7. Apply heuristics wherever the trend line "seems wrong"
   1.8. Upload the replenishment orders by copying from spreadsheet to SAP.
   1.9. Return to 1.1 until all stores are served.
2. Thursday night a scheduled operation in SAP assured allocation stock released from LEGO system to the LBR orders in same sequence as the orders had been submitted to SAP by the inventory team.
   2.1. Replenishment orders for which there was stock, are fulfilled and a virtual delivery document was created (deliveries).
   2.2. Replenishment orders for which there is no stock, stay on SAP as "backorders".
3. After stock allocation SAP sent notification to the warehouse that there are new deliveries.
4. Friday morning the warehouse downloaded the documents for the deliveries and calculates the number of trucks required to deliver to the outlets.
5. Monday morning the LBR products were picked, packed and loaded for dispatch to the outlets.
6. Tuesday and Wednesday the physical deliveries arrived at the LBR outlets.
7. The LBR outlets took the most urgently needed goods in and told the carrier to hold the rest for later call off. When the deliveries were large, the outlets would have daily replenishment deliveries from the carrier without interaction with the inventory management team.

Table 20 LBR manual process

This process revealed a set of immediate opportunities for improvement, such as saving 10% additional costs caused by redeliveries (7.), but in order for the inventory management team to be able to execute effectively the most important element was to repair the spreadsheet they used every day.

7.2 LBR with VBA automation

To repair the spreadsheet the choice at the time was to use visual basic (VBA) to perform the simple operations as a set of loops. The spreadsheet was thereby changed to:

1. Two inputs sheet: POS data and Inventory.
2. A set loops with output to temporary sheets for validation during development.
3. Out output sheet: replenishment orders.

The work process below illustrates the weekly work process (step 1. Above) supported by the spreadsheet with VBA (VBA-template):

- 1.1. Pick the outlet with highest revenue, for which no replenishment order has been created.
- 1.2. Download point of sales data from SAP (POS data) and copy into VBA-template
- 1.3. Download inventory position from SAP (inventory) and copy into VBA-template
- **1.4. Press "run button", which automatically:**
  - 1.4.1. Calculate a linear trend line of projected demand based on POS data
  - 1.4.2. Create replenishment orders so that inventory + replenishment orders would cover the current and 3 following weeks of projected sales.
- 1.5. Revisit emails from LBR store managers complaining about shortage/excess stock and adjustments the replenishment order.
- 1.6. Apply heuristics to adjust the orders VBA-template "seems wrong"
- 1.7. Upload the replenishment orders by copying from spreadsheet to SAP.
- 1.8. Return to 1.1 until all stores are served.

Table 21 Step 1 after automation from LBR manual process (table above).

The automation (step 14) reduced the order creation from about 1 hour to 4 minutes for each outlet and the manual adjustments (step 1.6) were reduced to less than 5% of the orderlines.

As the time saved allowed for further analysis of POS data, a set of logical flaws were discovered:

- New products were not ordered as they did not have past POS data.
- Outlets which were out of stock, showed zero sales even through the products might be popular and in demand. However with zero POS data, the trend line is skewed negatively.
- Back orders in SAP for unfilled replenishment orders extended almost 10 months which obstructed the queue for stock when new orders were added.
- The size of the outlets were unaccounted for, resulting in inventory of volumes equivalent to 2.5 weeks of sales, being stored at the carrier with daily redelivery attempts.
- Products were ordered in loose units, and not in full case pack quantities which are less time-consuming to pick and pack.

The work process (below) illustrates the weekly work process with the revised VBA-template "v2":

- 1.1. Pick the outlet with highest revenue, for which no replenishment order has been created.
- 1.2. Download point of sales data from SAP (POS data) and copy into VBA-template
- 1.3. Download inventory position from SAP (inventory) and copy into VBA-template
- 1.4. Press "run button"
  - **1.4.1. If no POS data, use nearest price point and product category as substitute.**
  - **1.4.2.** Calculate a linear trend line of projected demand based on POS data**, but ignore records with days with zero sales if stock is zero.**
  - 1.4.3.Create replenishment orders so that inventory + replenishment orders would cover the current and 3 following weeks of projected sales.
  - **1.4.4. Replenishment orders are rounded to full case pack volumes.**
  - 1.4.5.If replenishment order > 25% of store capacity, alert the user with message box, so that multiple dispatches can be arranged.
- 1.5. Revisit emails from LBR store managers complaining about shortage/excess stock and adjustments the replenishment order.
- 1.6. Apply heuristics to adjust the orders VBA-template "seems wrong"
- 1.7. Upload the replenishment orders by copying from spreadsheet to SAP.
- 1.8. Return to 1.1 until all stores are served.

Figure 30 The revised VBA-template with product substitution, error correction and rounding to case packs

The unfilled replenishment orders were removed manually and as per routine cleaned up every Friday after the allocation performed by the SAP ECC 6.0 Thursday-Friday night was completed.

This initial review of processes and tools helped to increase productivity, but only provided the foundations for the base case.

Two main problem in the process persist:

A. That the orders from the "most important outlets" - the ones having the largest revenue - are processed first in the week, and the less important ones toward the end of the week. This leads to a queue of orders in the ERP System by which stock is assigned, but with the side-effect that the most important stores use the most obsolete information. In addition, as only limited stock may be available the method sustains the self-fulfilling prophecy that well-performing stores, i.e. stores with higher revenue appear earlier in the queue and hence achieve a higher order-fulfilment than the stores which appear later in the queue.

B. That the impact of supplying a given outlet with stock is unknown for 2 weeks, due to delays in the network of information processors. The figure below illustrates the problem, by highlighting that the impact of replenishment decision (week 1) only have effect for 3 days until a week

3's new decision is made. The effect of the replenishment decision from week 2 remains unknown.

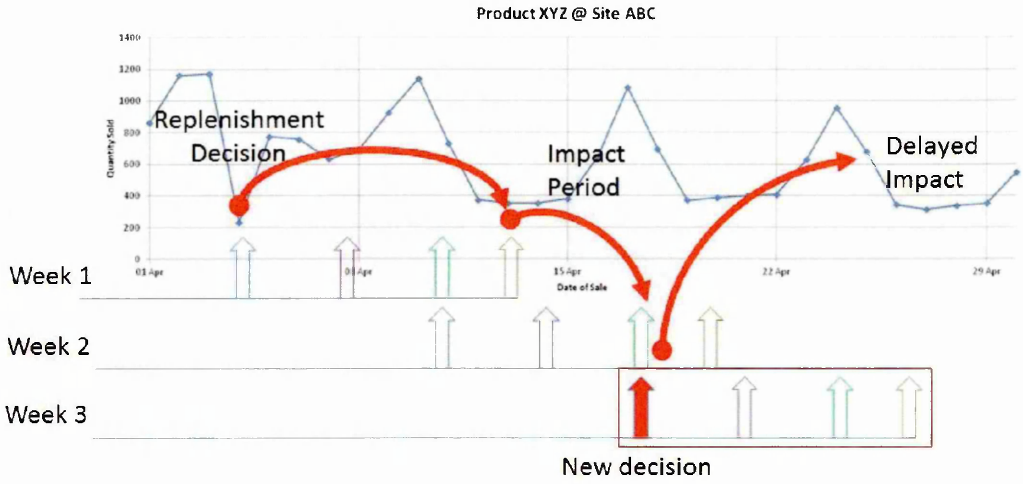


Figure 31 The impact of the decision to replenish remains unknown for 14 days.

The observed delay is the result of the process which is a network of queues:

1. First the POS data is captured, but not used for 2-5 days.
2. Next the out-dated POS data is used to commit resources.
3. Thirdly commitment of resources remains without re-evaluation for 2-5 days.
4. Fourth, the warehouse operation and transportation executes the delivery on a just-in-time basis.

In summary, out of 14 days from order to delivery, the process uses the information for 2 hours (administration) and 2 days (logistics). The rest of the time the information is used for nothing. That is 14.88% value added time drawn from the available information. All process steps are effective, but the system is unquestionably based on a network of queues.

The chart below illustrates a simple count of number of transactions per day from the database with demand events. It shows that number of updates range from 4,000 to 16,000 updates per day, or on average and update every 21.6 to 5.4 seconds, which provides evidence that information arrives all the time, but isn't used for revised decision making.

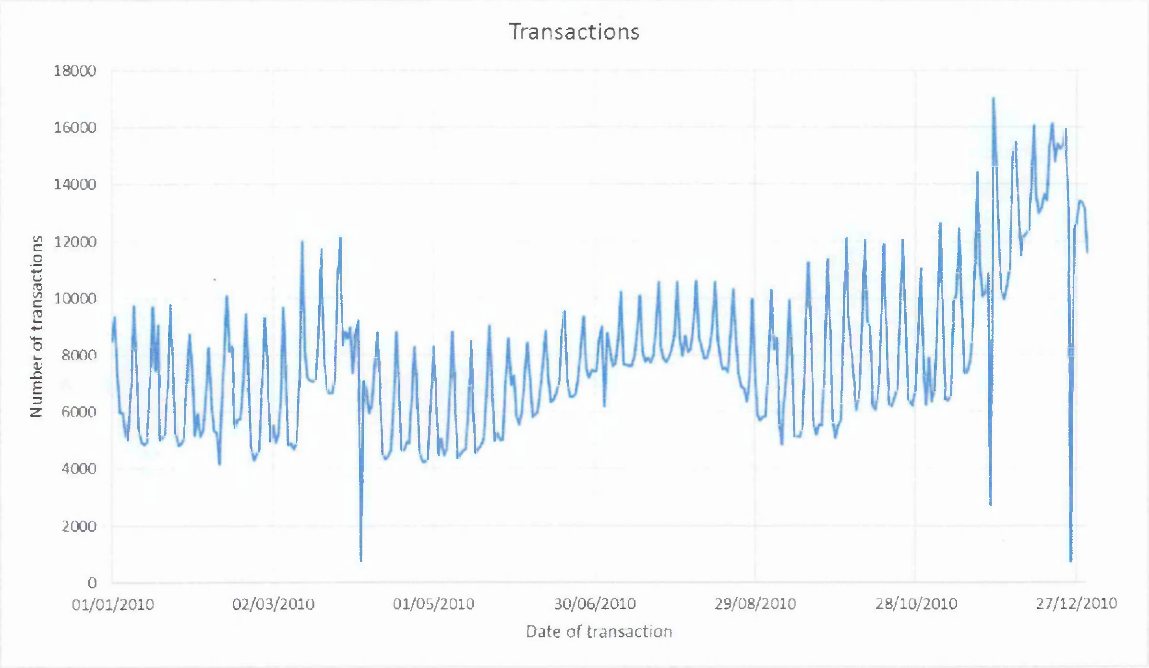


Figure 32 Count of demand events per day from the LBR dataset

As these problems are focal to the thesis, a set of options should be outlined for the business:

A. Simplest is to delay the decision making until Thursday evening, so that information processing is performed based on the most up-to-date POS data before delivery creation. This would require the spreadsheet to be extended to 1 extra loop only, at the cost of forfeiting the opportunity to apply heuristics (1.6). Ad-hoc orders based on complaints for outlet managers (1.5) could still be performed incrementally. The next step would be to deliver in alignment with the outlets, so that stock is dispatched when:
   a. The outlet is capable of receiving the replenishment order.
   b. The vehicle can be utilised effectively.

   This "optimisation" simply assures that excess costs from letting the carrier redeliver are avoided.
B. The third step would be to allocate stock in SAP every night, so that the outlets can be assigned stock as demand "pulls" (Womack 2008) for replenishment. This would also require a slightly modified business process in the warehouse which picks and packs the goods and rounds the volume that is to be dispatched to efficient shipping quantities. Using the VBA-template to assure rounding of ordered quantities to full case-pack quantities and thereby eliminates the otherwise time consuming single-item picking, is trivial. So is the splitting of deliveries. With these changes the warehouse would be capable of coping with the changed workload.
C. A final option is to improve the forecasting method. As LBR records every sale electronically the 4,000- 16,000 transactions which it logs every day provide a suitable dataset for time series analysis. The experiment would thereby purchase a commercial forecasting package in the range of \$ 600 - \$ 800.

Based on these three options, the "most productive intervention" can be determined experimentally through simulations 1.1 through 3.3 below:

1. Do nothing about the interpretation of information.
   1.1. Do nothing about delay
   1.2. Reduce delay locally (i.e. administrative delay – option A)
   1.3. Reduce delay in the whole chain (option B)
2. Improve interpretation of information with better forecasting (option C)!
   2.1. Do nothing about delay
   2.2. Reduce delay locally
   2.3. Reduce delay in the whole chain
3. Have perfect information by loading future demand as forecast!
   3.1. Do nothing about delay
   3.2. Reduce delay locally
   3.3. Reduce delay in the whole chain

Using Figure 16 from chapter 6.1 Overview of concepts, the experiment's listed above can be shown as how they lead towards the "most productive intervention" as the error term is minimised.

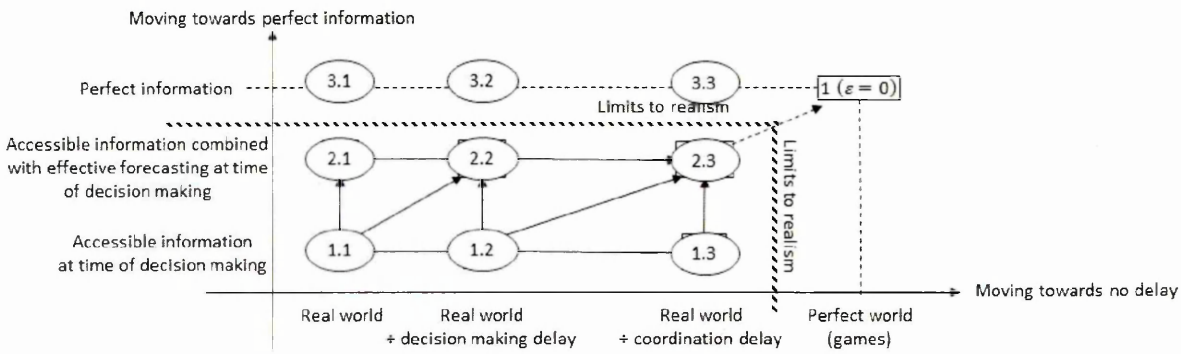


Figure 33 The list of experiments from the case study represented as bubbles on top of Figure 16.

In the experiment a set of sources for errors need to be considered:

One error which consultants do when they perform analysis – such as outlined above – is that it is done retrospectively (See also Table 15). To avoid this error, the experiment needs to be performed in chronologically progressive steps over a significant time period. Hereby POS data, inventory position by a given start date and supply of inventory must available for, say, a full year in a database. The chronologically progressive method thereby requires that the "new supply chain model" only is given data for one day at a time and required to commit resources incrementally. This means that for a full year, with commitment of resources on a daily basis (cases 1.3, 2.3 & 3.3) there will be 365 individual commitment calculations. In contrast the weekly basis (the rest of the cases) will be performed with 52 calculations. Of pedagogical reasons it cannot be stressed enough that the consequence of a chronologically progressive method "locks" committed resources at the end of each simulation, just as the decision maker will be required to commit to his/her decision when submitting orders to SAP. The 52nd calculation in the simulation does thereby not revise the whole year, but is only permitted to revise the allocation for the uncommitted horizon and must take starting point in the consequences from the 51st calculation.

By "re-running" the POS-data as demand with chronologically progressive decision making, the problem of adjusting the base-case data to account for changes made in the organisation during the data capture horizon also become irrelevant. The replay of the decision making method on a realistic data set is what the investigation is about: By using the POS data as demand and using each of the methods (1.1 – 3.3) to determine the replenishment orders, the theory indicates that a loss should be visible. To be explicit:

The only observation which can be made is that the each of the methods cause a loss. The question is "how much"?

The "ideal" scenario ("perfect information" & "perfect world" in Figure 33) will provide a reference point of the 100% ideal schedule of how goods could have been consolidated to maximise utilisation of the supply chain. This reference point will use the POS data as forecast (hence perfect information) and immediate transfer of stock (zero logistics lead-time) to maximise order fulfilment and profitability. This scenario should thereby has zero loss. All the other scenarios have some loss through the delay and imperfect information.

A final point of critique is, drawn from the SCM Literature review, is the modellers ability to create a reasonable model for the experiments. However as the author was the programmer who developed the VBA-template v1 and v2 it can only be concluded that there was more than sufficient insight.

7.3 LBR with NSCM as MAS

The LBR network[29] is thereby based on the New Supply Chain Model with 3 main entities:

1. Supplier – a site with weekly output from the production which is invisible to the network.
2. LBR inventory management team – a site with a full business unit and two processes for warehouse and transport costs.
3. 20 LBR Outlets – which each are sites with real-time stock profile, forecasting, etc., but no processes).

The events and activities in the simulation are:

- The supplier releases stock into the system as permitted by supply data from the database. There are 52 supply events where the supplier updates the fulfilment profile.
- The supplied stock is stored in the warehouses is available for allocation, from which deliveries can be created.
- Dispatches from LEGO system (supplier) to LBR inventory are instantaneous as they are physically in the same shared building.
- Dispatches from LBR inventory to Outlets are determined by the real geography, ranging from 12 to 24 hour transit time plus waiting until the outlets opening hours permit unloading.
- The demand agents are created based on POS-data, consumes available stock if it is present in the outlet, or registers a lost sale and ends its lifecycle. With one agent per transaction, this constitutes 2,857,414 demand agents.

[29] This description differs from the one presented in the IEEE paper to provide a more coherent overview with the NSCM.

- The outlets maintain the forecast based on POS data and forwards it to the LBR inventory team. Whenever a new demand agent appears, the outlets forecasting agent recalculates the demand profile for the particular product. In the implementation each product (SKU, not item) has its own agent to exploit parallelism and can learn from other outlets if it doesn't have sufficient data for the calculation.
- LBR inventory team calculates the replenishment orders by first attempting to fulfil all outlet purchase orders received from the outlets. Next it attempts to maximise profitability wherever there may be a shortage of stock. LBR inventory management team do not forecast, as this would result in the bullwhip effect[30].
- The transportation channel "stores" stock for the duration of transit until it is made available to the outlets.

For details on the accounting principles applied in the model, please visit the appendices.

The below illustrates the case as the New Supply Chain Model:

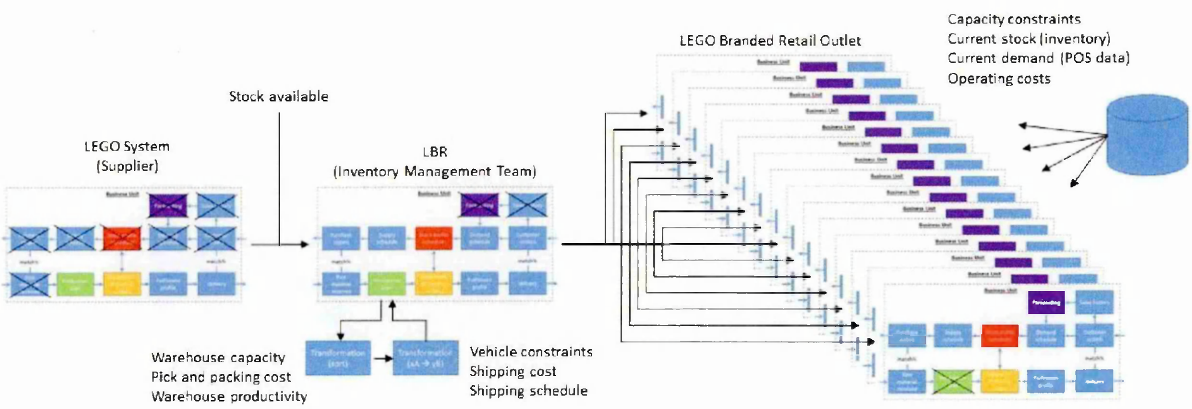


Table 22 LEGO Brand Retail illustrated using the New Supply Chain Model

Four performance metrics are used for evaluation of results:

- *Lost revenue* is calculated as $\sum_{demand=1}^{n} units\ unfulfilled * value$
- *Costs* are calculated as cost-to-serve, using simulated microeconomics which is described in the appendix under accounting practice.
- *Profit* is calculated as total revenue minus total costs, where revenue is calculated as "retail value of units sold".
- *Service level* is calculated as "count of demand agents" which are not satisfied divided by total agents. Example: 1 – (285,741 / 2,857,414) = 90% service level.

In addition to the performance metrics, the runtime was monitored. The implementation of the New Supply Chain Model in with .NET on a HP i7 vPro with 32-bit windows7 completed the asynchronous calculation with 2,857,414 demand events distributed over 365 days, in 189 minutes and 21 seconds. This is equivalent to 52 seconds per day, leaving the schedule up-to-date 99,964% of the time. This measurement is not exactly a scientific measurement of runtime as

[30] Considered common knowledge for M.Sc. in SCM & logistics. Hence no reference.

approximately 60% of the time is spent on querying the Microsoft SQL server's database for the supply and demand events in addition to time wasted by the operating system switching between tasks. However when compared to runtimes experienced in by Enterprise Resource Planning systems, which typically range from 3-28 hours, this is a significant improvement of the availability of a valid schedule.

The performance metrics are summarised as illustrated in the table below where the percentages are indexed as percentage the "ideal" scenario:

Scenario			Lost	Service		
Forecasting	Delay	Scenario	revenue	level	Cost	Profit
Do nothing about the interpretation of information	Do nothing about delay	1.1	40%	66%	95%	56%
	Reduce delay locally (option A)	1.2	35%	71%	96%	61%
	Reduce delay in the whole chain (option B)	1.3	20%	86%	105%	76%
Improve interpretation of information with better forecasting (option C)	Do nothing about delay	2.1	31%	69%	95%	66%
	Reduce delay locally (option A)	2.2	22%	79%	96%	76%
	Reduce delay in the whole chain (option B)	2.3	16%	86%	105%	81%
Have perfect information by loading future demand as forecast!	Do nothing about delay	3.1	17%	82%	96%	81%
	Reduce delay locally (option A)	3.2	17%	83%	96%	82%
	Reduce delay in the whole chain (option B)	3.3	10%	90%	102%	88%
Perfect information	Perfect world	"ideal"	0%	100%	100%	100%

Table 23 Results

The experiment was designed to reveal the loss to the supply chain caused by delay which is illustrated in the figure below as "damage to profitability caused by delay in decision making".

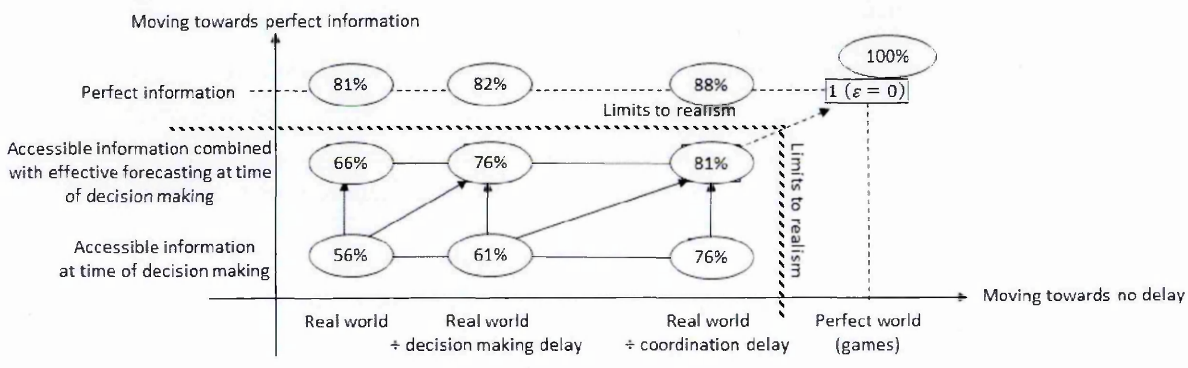


Figure 34 LBR results - percentage of ideal profit achieved using the methods 1.1 through 3.3

A few observations are appropriate for interpretation of the results:

Forecasting 1 - First the notion that improvement of forecasting method yields a higher profitability (56% -> 66%) than removing the delay cause by LBR decision making (56% -> 61%). The logic required to explain this observation is simple: If a poor decision is made faster, it will by default not be a better decision.

Forecasting 2 - The second notion is that even with a perfect forecast (scenario 3.1) the supply chain is only capable of achieving 81% of the potential profit.

Delay 1 – The scenarios 1.1 and 1.2 illustrate that delay in the current system where LBR uses a visual basic template for managing the business, results in a relative loss of 61% - 56% ≈ 5%-point of profitability, which at 56% of potential is close to 10% in damages. If the employees automated the process as option B implies (and thereby made themselves redundant) the performance of the business would increase equivalently.

Delay 2 – The increase in profitability from scenario 1.2 (61%) to 1.3 (76%) of 15%-point justifies the extra costs associated with more frequent, smaller, pallet shipments instead of attempting to fil the trucks. Pallet shipments are still very efficient in the supply chain as consolidation of cargo going towards the same region cost-effectively come at a lower price than sending a partly loaded vehicle. Supply chain managers struggle to provide evidence for this case as the consolidation needs to be coordinated with the receiving customer, which often is beyond their reach of authority. This case illustrates that when retail-logistics is coordinated such that both parties can operate efficiently, the cost increase from 95%/96% to 105% of the "ideal" are outweighed by far by the increase in profitability (->76%) as the revenue increases towards 80% (20% lost sales) due to more timely allocation of goods to the outlets who need them.

Forecasting & Delay 1 – The combination of improved forecasting and reduction in internal delay (scenario 2.2) is as influential on profitability (56% -> 76%) as changing business process without improved forecasting (scenario 1.3) (56% -> 76%). However as the lost revenue is reduce from 22% to 20% and service level increase from 79% to 86% the "right choice for the business" is in theory to pursue the service level. However as the forecasting package used only would cost the business $600 - $800 in licensing fees the only sensible option is to strive for scenario 2.3 which gives the best possible result to the business. See Table 24 below.

Improvement direction	Scenario	Lost revenue	Service level	Cost	Profit
Reduce delay in the whole chain (option B)	1.3	20%	86%	105%	76%
Reduce delay locally (option A)	2.2	22%	79%	96%	76%
Reduce delay in the whole chain (option B) + forecasting package (option C)	2.3	16%	86%	105%	81%

Table 24 Excerpt from Table 23 Results.

7.4 Subsequent implementation challenges

The results of the case study were convincing enough for the financial controller of LBR to study the implementation in detail and recommend that the business case was prepared for implementation of the system. The pilot study would implement the following features:

- The MAS based on the New Supply Chain Model would be implemented.
- The Inventory Management Team would the MAS output instead of the VBA-template, so that they could make corrections if needed.
- The MAS would update SAP using SAPs .Net connector (an API).

According to the results achieved in the simulation the usage of the MAS would reduce the losses caused by delay and inappropriate forecasting with an amount that would give the project a return on investment time of 2.1 days.

When the business case in the LBR MAS project was presented to the executive management with a return on investment time of 2.1 days, it was considered impressive, yet the executive decision maker of LBR considered the project "too advanced for the organisation to implement".

A year after the study, the management team initiated a request for quotations for an automated system for inventory management which performed the same operations as the MAS which LBR sponsored the development of. The NSCM is now pending decision for implementation.

7.5 Conclusion

The observation is, that decisions made based on outdated information, will lead to a mismatch between action and effect. In hindsight this is not a surprise. What is a surprise is size of the accumulated error. LBR operates with 2 weeks of delays and premature commitment in its decision processes explain the gap from 56% to 76% of the achievable profitability. A number which – when scaled from 2010 data to 2014 revenue ($-US 438m) is nearly to $-US 100 million on top of the existing results.

On may therefore argue that the chart below (Figure 35) is the most important chart of the thesis, as it illustrates how the aggregate error grows when outdated information is used. The grey area which represents the gap from missed opportunities to adjust the schedule to updates, results in the total loss of profit from 56% - 81%. Hereof three factors contribute: Poor forecasting techniques, premature commitment of decisions and delay in execution.

As generalised conclusion on can argue that when the frequency of updates to information repositories increase, the aggregate error of interventions increase unless the decisions may be revised at the same pace.

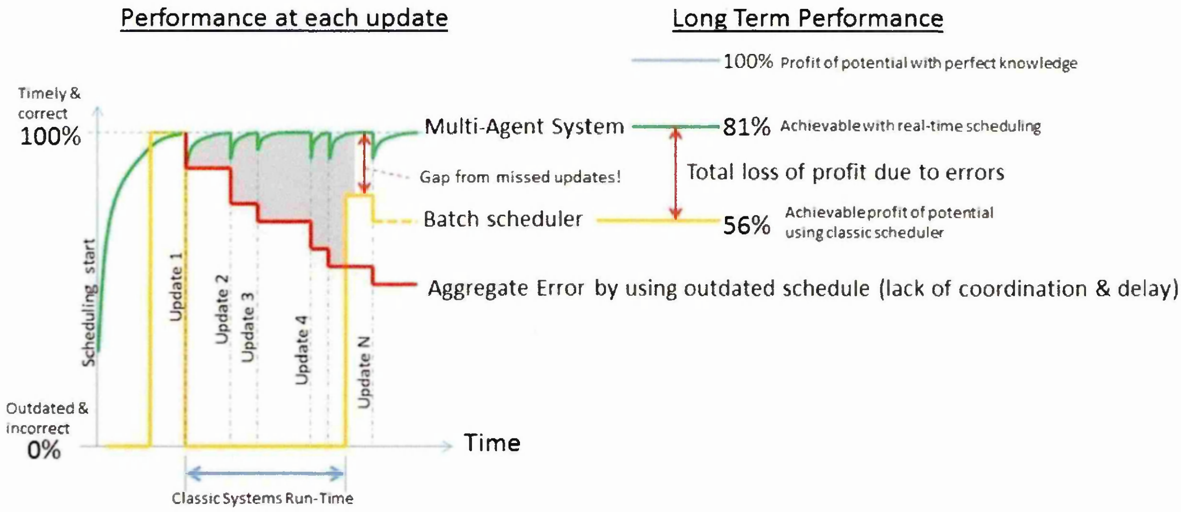


Figure 35 A generalised model of the results (lost profit)

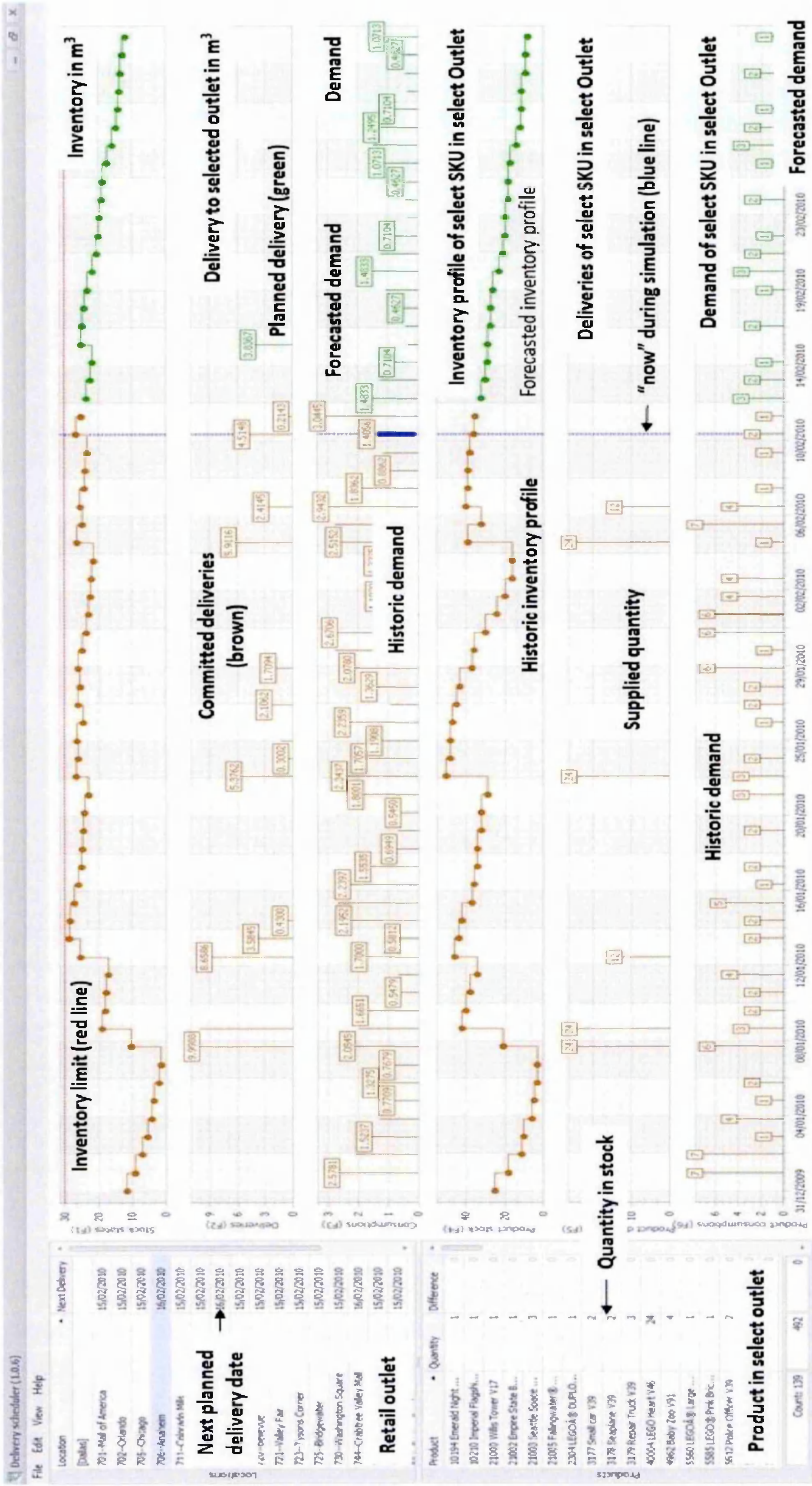


Figure 36 Screenshot of the user interface used for the LBR case study. The blue line illustrates the transition between uncommitted and committed time horizon

Chapter 8 – Case Study III: Real-time Manufacturing (FMCG)

Introduction – This case is concerned with a top 100-megabrand who wishes to remain confidential. The study is structured similar to the previous case (LBR) and pursues similar objectives:

1. First it presents the background to why the FMCGs problem exists at all.
2. Second it provides an overview of how the FMCG attempted to cope with the problem.
3. Third, the coping strategy is criticised and the outline for a solution using the NSCM is given.
4. Fourth, the NSCM is deployed to simulate the effect on order fulfilment and costs.
5. Fifth (and finally) the challenges for the implementation is given by the hosting organisation.

The chapter concludes with reflections on the results and observations made during the case study.

8.1 FMCG using Excel manually

The case focuses on an order management team (OMT) who have been given the responsibility to assure optimal assignment of finished consumer goods to customer orders. The orders are received from retailers, ranging from global retail chains to local merchants. The supplies are produced by the FMCGs supply chain. The FMCG's management team pursued to increase the range of offers to its customers, which was done through a series of acquisitions over the past decade, which also resulted in a highly fragmented information platform. This result in absence of coordination with the subsequent consequence of high logistic costs. To resolve this problem the OMT was established. The OMT was staffed with 6 full time employees (FTE) who were assigned to one of 6 plants and DCs each. The OMT tasks were:

1. Assignment the customers to a default dispatch location (annual review):
   1.1. One of 40 local depots, if the typical volume ordered was less than full truck loads (FTL).
   1.2. One of 7 distribution centres (DCs), if the typical volume, was divisible in FTL.
2. Monitor that volumes ordered from DCs were FTL and contact with Customer Service in case of deviation, which typically was order-entry mistakes.
3. Assure that sufficient stock was available to fulfil the product range sold from the depots through forecasting.
4. Assure that forecasted volumes less than full truckloads (LTL) orders were consolidated into full truckloads (FTL) and dispatch from the nearest DC.
5. Resolve conflicts caused by mismatch between production output (supply) and customer orders (demand).
6. Keep customer service informed so that they could relay any deviations in time and quantity to the customers.

Microsoft® Excel® was used as the primary data analysis and manipulation tool, which the OMT used to receive data from 6 different production systems in attempt to increase the coordination. However as the employees in OMT struggled to synchronise their activities through verbal communication the OMT management became aware that there were limits to how much information humans effectively can exchange.

8.2 FMCG using MS Access for automation

The manager of OMT arranged that a Microsoft® Access® database was constructed to merge the information from the 6 production and inventory systems into one. The database system allowed the OMT to make changes and perform a simple auction in the same room. An employee who needed a particular item that was not available in his/her assigned plant/DC could call out to the other employees: "Does anyone have a FTL with item X2567 available? I need at least 52 pallets?" Another employee would respond "yes" or remain quiet. In this manner coordination happened informally around the table. As the MS® Access® database (Access DB) was updated throughout the working day, a solution emerged and the result was as good as the employees could negotiate. As the 6 employees each could handle one update every 2-5 minutes (total 500-1200 updates a day) and the system received 40,000 order lines on average) the influence of the employees was limited (1.2% -3.3%). With the MS® Access® database the workflow also changed:

1. User Logs in through the User Interface of the database at 07:00 am local time.
   - 1.1. The user filters the view of orders on unfulfilled orders
   - 1.2. The user searches for excess stock in DCs of the colleagues
   - 1.3. The user select the lowest cost delivery option the (nearest DC which has idle capacity)
   - 1.4. The user evaluates time to delivery:
     - 1.4.1. If order equals FTL move order to be fulfilled from colleagues DC in database and inform colleague.
     - 1.4.2. If order does not equal FTL, but there is enough time: Ask colleague for FTL transhipment to own DC.
     - 1.4.3. Else contact customer service that no stock is available at requested delivery date (customer service might move the delivery date one day later, but OMT did not influence this)
   - 1.5. (From 1.4.1 and 1.4.2) The colleague reviews the updates in database and accepts/rejects request verbally.
   - 1.6. Every two hours the update in the database is uploaded to the legacy system (return to 1).
2. At 14:00 pm local time the reassignment stops and the database content is loaded back into the legacy systems.

The Access-DB helped to develop a uniform approach to transfer FTL shipments from one DC to another to increase order fulfilment, but the method had several shortcomings:

- The Access®-database's default assignment of orders to the closest DC contributed to a reduction in logistics costs, but left the employee to deal with many exceptions.

- Though the customer orders by default were assigned to the closest DC, but the inter-DC shipments were of any distance, ignoring the option to create chain propagations, which might cause more movements of stock, but kept the total costs lower.
- If an OMT employee didn't want to give up unused stock to another employee, the order would remain unfulfilled. The incentive program contributed to this behaviour, as the OMT employee was benchmarked on order fulfilment, but not inventory- and inventory-holding costs whereby the volume of inventory in stock could be up to 2 weeks of sale.
- When one of the 20 production lines had a different output than expect – for example due to a mechanical fault on the packaging line – the planned quantities had to be reassigned. This process involved assignment of stock to priority orders, such as where the account managers had ongoing product campaigns, and thereby reducing the stock assigned to others. That meant, for example, that a retailer chain would always get stock, whilst hundreds of local merchants would not – disregarding the fact that the local merchants may be more profitable.
- Production runs were made in batches that matched the calculated economic order quantity. If an order required 1.2 batches, the 0.2 had to wait until the demand was divisible in whole batches, instead of extending the batch-run to match the confirmed demand.
- The forecast of packaging material was revised only once a week.
- Workload calculation at the warehouse was not coordinated as the same staff looked after inbound and outbound volumes. Thereby a large batch delivered from production could inhibit dispatches as the staff was relocated to get the stock off the production line before continuing.
- With commitment made every 2 hours, using the batch-based optimisation approach, there would be no guarantee that decisions made earlier would not be revised. For example if a call was made to a customer made at 08:57 am requesting a booking slot for unloading of a FTL the next day, the updates performed 3 minutes later (at 09:00 am) could render the appointment invalid if the stock was assigned to another customer – unless of course the customer service employee was quick enough to register it as a locked agreement. With a dominance of asynchronous email correspondence to the customers, this type of problems would render all communication difficult.
- Communicate with customer service whenever the customer order only could be partially fulfilled, as business practice required a 1:1:1 relation between order, delivery and invoice.

The observed problem is thereby the results of the interactive process is a network of queues with premature commitment:

1. First, the delay (in minutes and not days as in the LBR case) is in the interactive process between OMT employees and OMT and the customer service.
2. Second, that premature commitments are made without getting actual information about the state of supply.
3. Third, that workload is committed to the warehouse operations by production and customer service without negotiation or revision of the capacity requirements.
4. The lack of ability of OMT employees to exchange information about available supply and demand amongst one another.

These problems could be solved by:

A. Enabling an asynchronous workflow, with chronological progressive commitment. Not batch-based override as this may disrupt agreements made with customers.
B. Delay of commitment of resources that do not need to be committed, such as for example, limiting booking of delivery slots to stock that has been produced to complete the order.
C. Exploitation of all available stock across all DCs, with automatic consolidation of inter-DC loads.
D. Visualisation of the data to provide overview of conflicts of interest, so that OMTs attention is directed towards the most impactful problems. A large order of one SKU that is a scarce resource, can, for example, be cancelled with a single call to customer service. If that single call allows a wide number of multi-product-order which only are missing a 1- or 2 pallets of the constrained SKU, then the cancellation of the one FTL order can result in 26-52 fulfilled orders instead. Such decisions increase revenue without increasing logistic costs.

Options A. through D. leads to an experiment similar to the LBR case. However only the cases 1.1 and 2.3 are required:

1. Do nothing about the interpretation of information 1.1. Do nothing about delay. 2. Improve the interpretation of information 2.3. Reduce delay in the whole chain/

Using Figure 16 from 6.1 Overview of concepts, the experiment can be shown as an intervention which should lead towards minimisation of the error term caused by delay, which in this case is considered the most productive intervention.

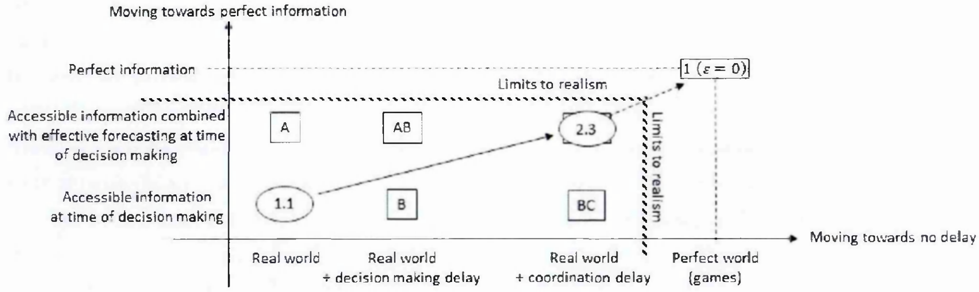


Figure 37 The two experiments in the FMCG case study overlaid on Figure 16

8.3 FMCG with NSCM as MAS

The NSCM model of the FMCG supply network has to account for:

- Random arrival of customer orders 48-96 hour from order receipt to delivery.
- 100,000 customer locations in a fully connected graph ($7! * 10^5 = 5.04 * 10^8$ links) with the DCs
- Requirement for 1:1:1 relation between order, delivery and invoice.
- Accept priority amongst customers.
- Shared Inbound/outbound capacity limit of the warehouse (120 FTL total in and out) including repacking.

- Packaging material lead-time >7 days, whereby forecasting is required.
- Production plan with change over times in the production (resulting in dynamic EOQ)
- Allow updates to production output due to incidents in the production (variation in output)
- 1- & 2-step cross docking via DCs (DC to DC cross flow)
- Stock updates are delayed, so that stock arriving before 23:59:59 is not released to the warehouse for picking until 00:00:01 +1 day.
- Quarantine of stock until batch is released from the laboratory. Quarantine groups:

Quarantine time in hours	% of Total quantity	Product Group
0	74.3%	A,B,C
72	1.0%	D,E,F
96	0.1%	G
120	9.6%	H,I
144	14.5%	J,K,L,M
168	0.3%	N
192	0.1%	O
-/-	100.0%	-/-

Table 25 Overview of quarantine hours and % of quantity of units in QA before release for sale

The quarantine operation is added to the production, so that each batch is uniquely identifiable. The quarantine agent is assigned a quarantine zone from production to DC, but is not permitted allow quarantined stock to move beyond the DCs.

- Reach to retrospective updates of the inventory caused by warehouse data entry errors.
- Include overlapping transport routes with multiple freight rates.
- Use BOM for production and for repacking

The events and activities during the experiment are:

Demand from 7^{th}-9^{th} of July 2013 with 129,000 order lines, bundled in 14,798 orders for 208 ship-to's across 211 active SKUs (of 270 listed). Supply is given with 14,396 production orders for 179 SKUs distributed across the 20 production lines with varying capacity. The warehouses can handled 398 dispatches a day through 1,456 active customer channels with unique freight rates. Storage limit is 201,000 pallet positions. Load calculation for vehicles is done with 19 unique SKU sizes. Production plans is taken for given if provided, otherwise a forecast of need is made.

Hereby 3 new properties need to be implemented in the NSCM:

- Quarantine is added to the product as product property.
- Delay added to the stock booking.
- Orders are assembled in the marshalling area.

The performance metrics used for evaluation of results are:

- Lost sales (units) is the count of demand agents which are satisfied. SCM literature uses the measure of "on-time in-full" (OTIF)
- Transport cost as other costs are fixed during the simulation.
- Cost per unit transported.

In this experiment a set of sources for errors need to be considered:

With the small sample size of orders – less than 7 days – the increased utilisation will only come from unutilised stock and otherwise poorly coordinated usage of the production output which is with less than 48 hours of quarantine.

The production plan can – if needed – be changed in sequence, but the content of the production can not be changed, as this would propagated to a new forecasting horizon for which the order data will not suffice.

The influence of the employees is also limited as this is a simulation. It is known that the employees manipulate the assignment of stock to orders in the range from 1.2% - 3.3%. The NSCM will also manipulate this assignment, but be respectful towards the OMT employees "override" of the NSCM's proposal. This means that "tacit" decisions for which the NSCM does not have information is augmented by the OMT employee's actions. However in the simulation such events cannot be incorporated as it makes the tacit information explicit. It can thereby only be concluded that the simulation will pursue the assignment which leads to the optimal order fulfilment and maximal profitability, where after the OMT intervention will reduce the result of the two performance vectors. The chosen approach is that the NSCM is used in ignorance of the tacit information.

A third error is the influence of random events, such as updates to production plans, DC capacity and production output. When loading the data to the NSCM it is not possible to know how disruptions may appear. To test the NSCM implementations ability to deal with the disruptive events, all information is provided to the NSCM in a randomised sequence. If the signal to noise ratio of the solution computed by the NSCM is less than 0.1% the performance metrics, then solution is considered stable as it handles the disruptive events caused by updates effectively.

In summary, **the only thing the experiment will reveal, is the result (performance metrics) of the ability to coordinate the supply chain actions across production and DCs when delay is eliminated and the ability to exchange information is done with an asynchronous operating agent based model.**

The figure below illustrates the principles applied in the model used for the case study:

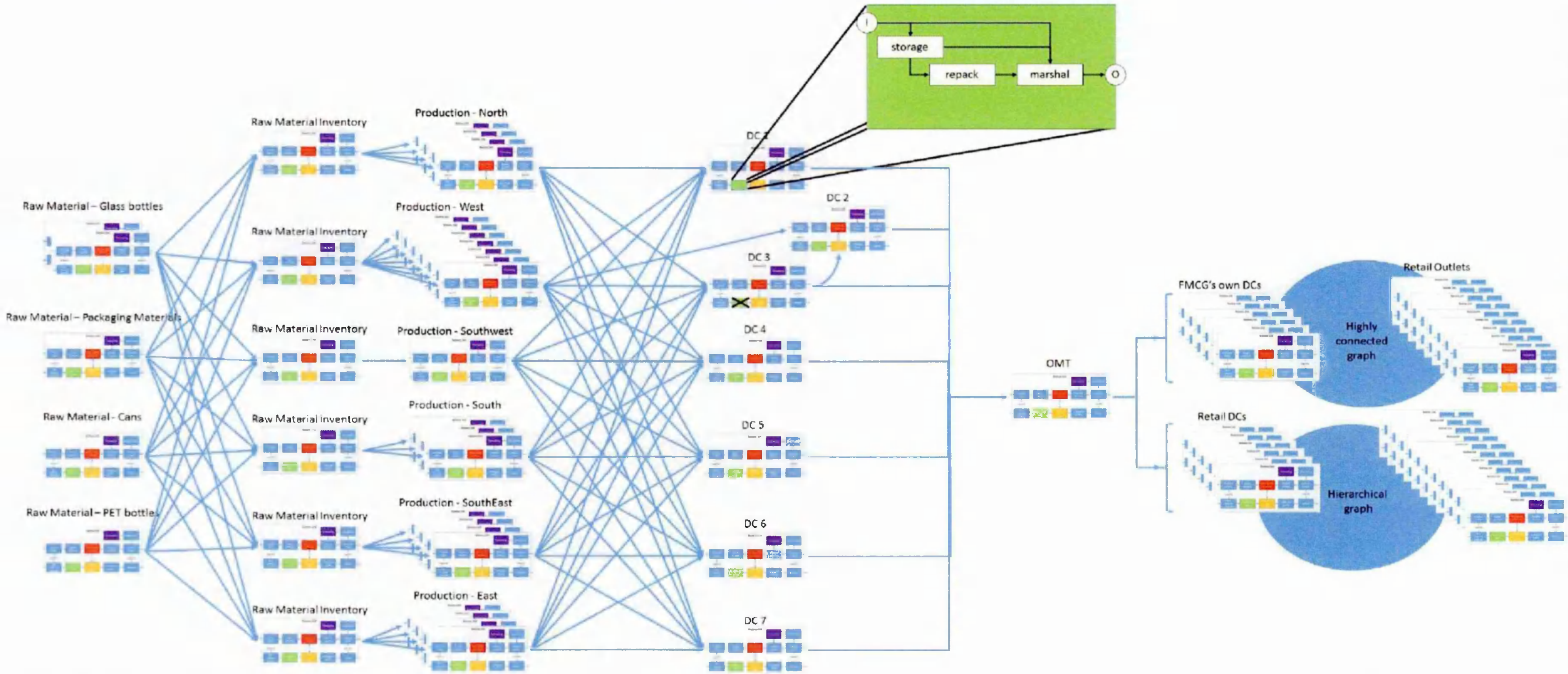


Figure 38 FMCG Supply Network illustrated as New Supply Chain Model

Overview of the supply network of the FMCG. From left to right:

Suppliers of raw materials for packaging. These are present as the main production constraint. Other raw materials were considered non-bottleneck and was requested to be ignored by the manager of OMT.

Raw material inventory storing materials for the production lines.

Production lines, 20 in total located in 6 production centres.

Distribution Centres, 7 in total in 5 are co-located 1:1 with production centres, 2 DCs are co-located with production centre “west”. All DCs have repacking lines, except DC3 which is a bulk pallet storage location.

Order Management Team, which assigns stock to orders.

Customer DC and FMCGs own DC, in the same echelon: The FMCGs own DC supply local merchants, whilst the retail chains are supplied via their (Customer) DCs. Retail outlets, selling to consumers.

Scenario		Scenario	Lost Sales	Total Cost €	Cost €/unit
Forecasting	Delay				
Do nothing about the interpretation of information	Do nothing about delay	1.1	77%	186k	0.47
Improve the interpretation of available information	Reduce delay in the whole chain	2.3	96%	220k	0.45

Table 26 Results

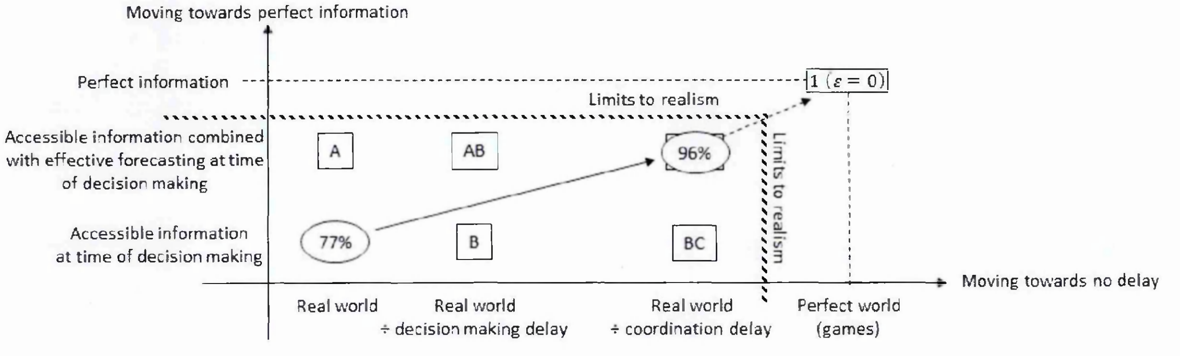


Figure 39 FMCG results on improvement of order fulfilment.

A few observations are appropriate for interpretation of the results.

The dataset is small, so there is no impact of improved forecasting to the production. There major influence comes from the significant amount of unutilised inventory, which each OMT employee govern.

3rd Party verification

Jim Wilson from Barloworld Supply Chain Software participated in the review and performed a 1:1 comparison of the implementation of the *new supply chain model*.

The report denotes:

To enable a comparison between the existing and optimised order allocations, the current Order Fulfilment KPI has been calculated based on the following assumptions as this cannot be provided by the business.

The comparison between existing and optimised order allocations shows a significant increase in order fulfilment as well as a reduction in transportation costs.

The **increase in OTIF[31] of 19 percentage points** represents a significant opportunity to increase both customer service as well as the resulting increase in revenue with associated profit margins.

This increase in **OTIF represents an increase in fulfilled orders of 25%.**

The transportation cost difference of €36K represents a 20% increase in transportation costs, although the cost per unit delivered actually decreased from €0.47 to €0.45.

8.4 Subsequent implementation challenges.

The immediate challenge with results of such significance as achieved is to go through the calculation with the OMT manager to develop a detailed understanding of the results. This dialogue produces a significant quantity of additional tests which attempt to falsify the model. Once basic confidence in the NSCM is achieved, the next implementation challenge is to attempt to "break the software", before it is taken into production. Examples are attempts to achieve memory overflow, concurrent tasks which overload the kernel and injection of corrupted messages in the communication system. Once the error handling is leaves a satisfactory level of resilience (takes about 2-3 months), the system is declared ready for production usage. From this point the FMCG needs provide information to the system. This is done via encrypted (TLS 1.2) hypertext transfer protocol (https) and a web-service API.

Whilst the NSCM completed its project elapse on time. The FMCGs IT department couldn't get the data from their newly implemented ERP system. This delayed the process of taking the system into usage, though it simplified the IT operations by removing the legacy systems.

The FMCGs ERP systems SCM module schedules once a week, but receives incremental "suggestions" from the NSCM. This leads to a redefinition of the role of the OMT employees, whereby they are preparing for "continuous improvement" (meta-intervention) across the business, so that disruptive events may systematically be eliminated. Deciding what to do operationally is left to the NSCM, which can be monitored from any browser using the UIs, which were design for the OMT to provide overview (Figure 40).

[31] OTIF means orders that can be delivered "on-time and in-full" (OTIF). See also (Christopher 2005)

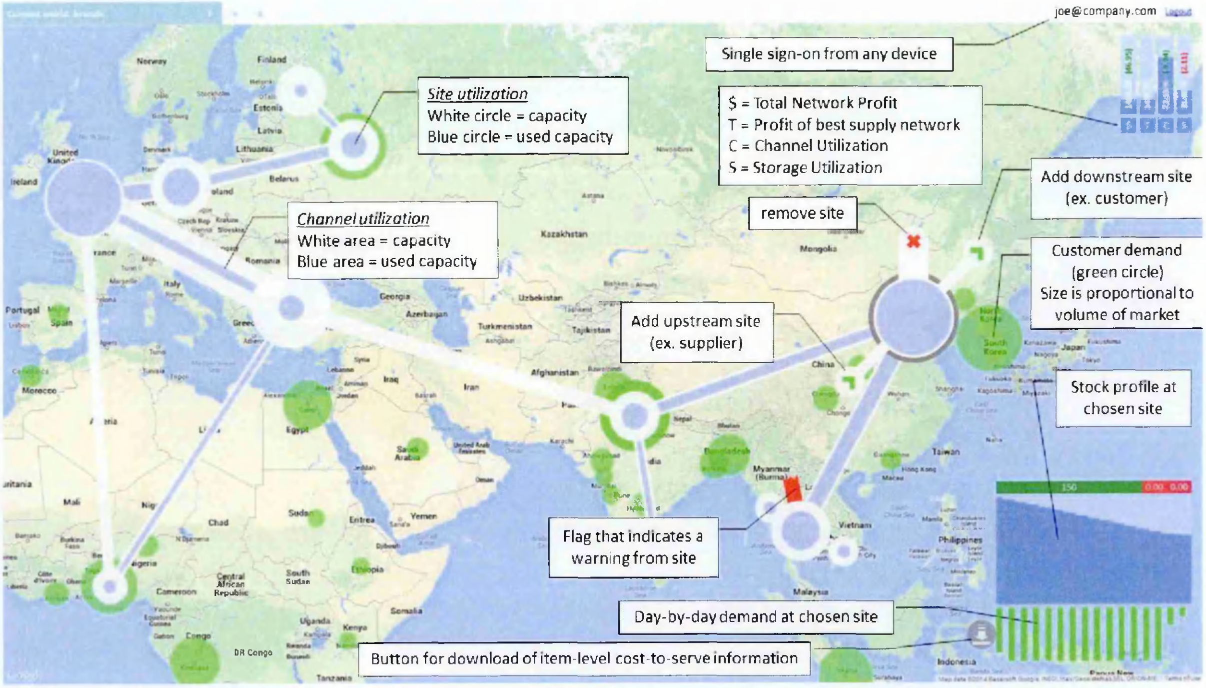


Figure 40 An anonymised model of the user interface created for OMT

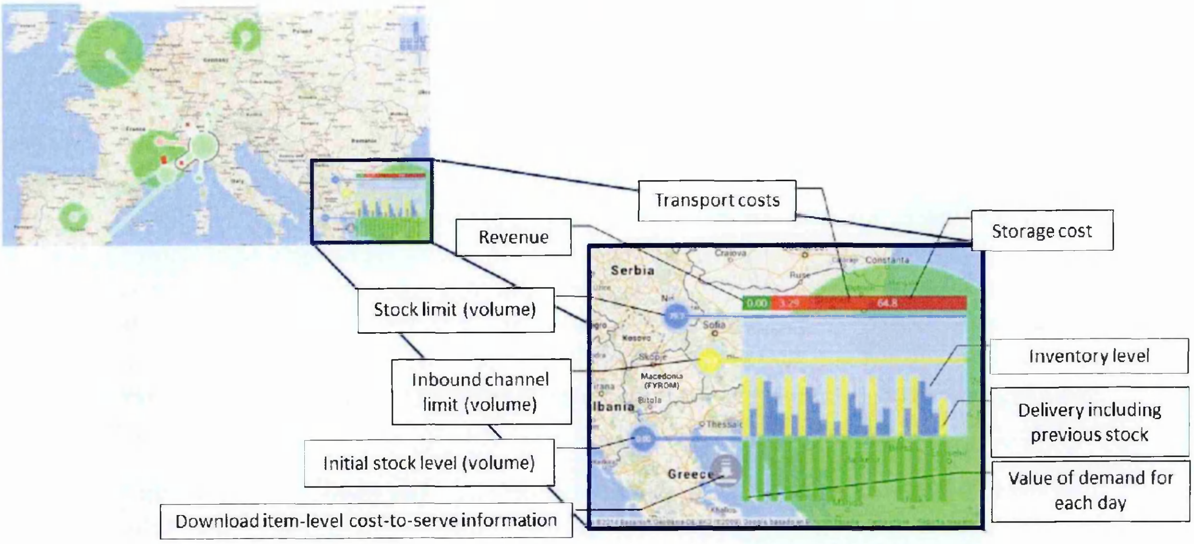


Figure 41 Detailed view of the user interface created for a selected location for OMT.

8.5 Conclusion

The lack of ability to coordinate in real-time leaves the optimisation of the activities in the SC without limited impact. The FMCG received up to 40,000 order lines a day plus a range of information regarding disruptive events. 6 FTEs had no chance of coping with the amount of information nor, had a living chance of optimising the flow of goods to minimize costs. In fact they didn't have an overview – which is why the new UIs were required.

The chart below (Figure 42) attempts to visualise the problem once more: The total loss of sales (77% vs 96%) occurs because of the lack of ability to use the information received from the environment.

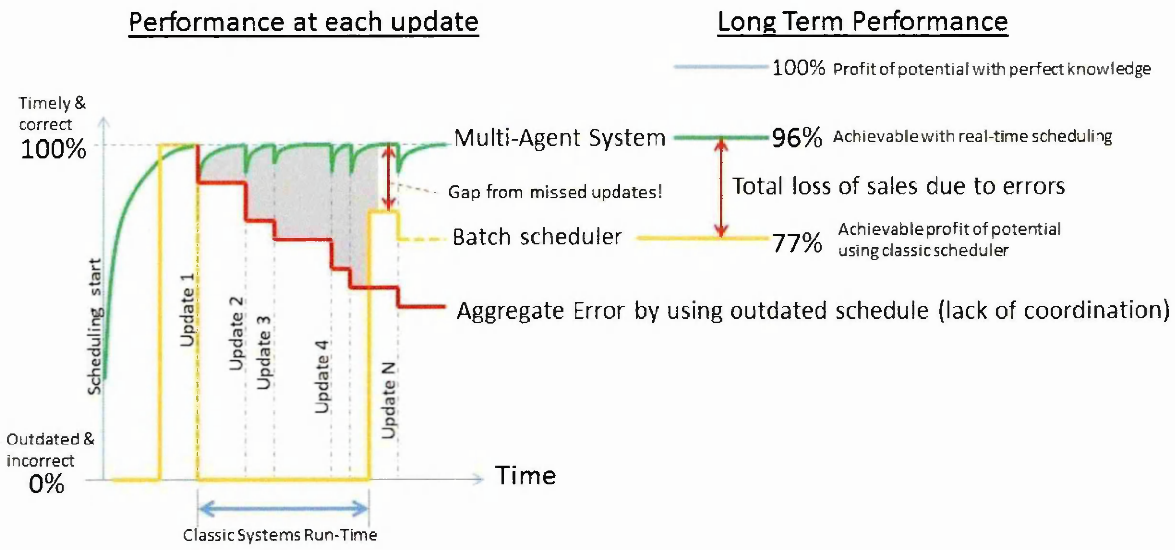


Figure 42 A generalized model of the results (reduction of lost sales)

Once it should be argued that Figure 35 and Figure 42 are the most important charts in this thesis, as they illustrate how the aggregate error grows when outdated information is used. The grey area which represents the gap from missed opportunities to adjust the schedule to updates, results in the total loss of sales from 77% - 96%. Hereof three factors contribute again: Lack of coordination, premature commitment of decisions and delay in execution.

As generalised conclusion on can argue that when the frequency of updates to information repositories increase, the aggregate error of interventions increase unless the decisions may be revised at the same pace.

Chapter 9 – Discussion of results

Introduction – The lessons learned from the case studies are discussed in with consideration of what can be learned, and considerations are given to the problem of distributed information processing where people & computers performing batch-information-processing are two side of the same problem.

The idea deduced from *Chapter 6 – Formulating the New Supply Chain Model*, was that the "New Supply Chain Model" should be characterised by:

- Distributed decision making,
- Real time decision making,
- Maximum delayed commitment,
- Only updates resource allocations affected by changes,
- Solution of optimisation problems by exchange of messages,
- Is batch-free,
- Minimises delay amongst people by transferring human preferences (customer satisfaction & profitability) to agents which vigilantly maintain optimality,
- Permits human override of system allocations,
- Permits simulation as decision support in parallel with handling live transactions.

And, that these design criteria should result in the most productive method of intervention in a complex economic system.

The three case studies presented measureable consequences of removing delay of propagation of information in the network in which the information was needed to take actions.

- In "*Case Study I: LEGO System*" delay was caused by a democratic process which allocated resources to demand. Optimisation did not utilise the supply chain. The overall impact of the delay caused a loss of sales of € 80m.
- In "*Case Study II: LBR*" delay was caused by method used for information processing. Resources were assigned to demand based on out-of-date information. Optimisation did not consider the supply chains constraints. The overall impact was a loss of sales (40%) and significantly reduced profitability (56%)
- In "*Case Study III: FMCG*" delay in information exchange remained an obstacle for coordination. Resources remained unutilised, though demand across the network was confirmed. The overall impact was a loss of sales of 19%-points.

The set of strategies which contribute to the potential increase in performance were novel, but nothing inhibits the ordinary organisation of adopting its principles:

Classic Approach	New Supply Chain Model
Pursue local optimisation.	Pursue coordinated distributed decision making.
Information can wait and is not urgent. Use batch-processing for updating information repositories	Information is critical. Use asynchronous information exchange to assure that everyone knows what is happening.
Up-/down-stream agents can wait for somebody/something to make a decision.	Pursue real-time decision making. Transparency of a bad solution makes people come up with more creative solutions.
Human decisions cannot be converted into algorithms. Waiting for people to make decisions is better than delaying coordination.	Human design agents, so that agents can help coordinate 24/7/365.
Make the perfect decision based on what is known	Identify and negotiate towards the most productive compromise.
Commit to longest lead-time to freeze plans.	Use maximum delayed commitment.
If the situation changes, then re-plan beyond the frozen horizon.	If the situation changes, then re-plan as much as possible to avoid waste of resources.
Use batch-processing to reschedule everything effectively, then command compliance to plan.	Isolate disruptive events by their propagation path, then patch the solution with a delta update and inform only those who are affected.
Search for solutions in available information only.	Exchange messages to solve problems.

Table 27 Classic approach versus the New Supply Chain Model.

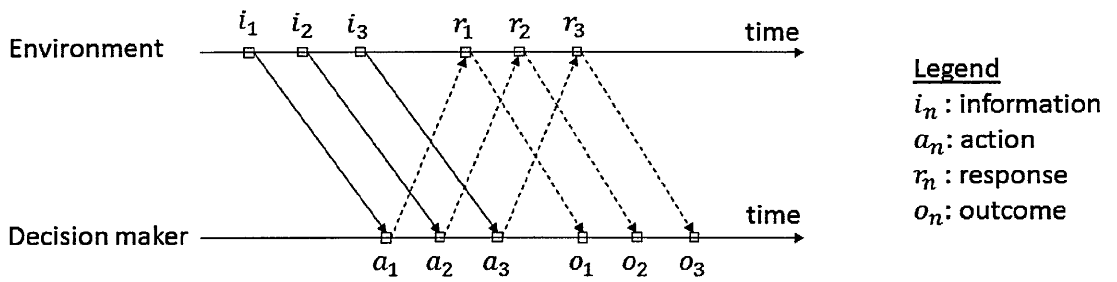
As departure from ideas of batch-processing of information and the interactive role of the people, play a significant role, additional reflections seem appropriate.

9.1 Batch processing

The view that the supply chain is an example of a complex economic system, implies that the system is irreducible (Miller & Page 2007; Prokopenko et al. 2009; Rzevski 2011). It may be possible to focus on parts which have defined borders, but their interaction with the broader economic system cannot justifiably be ignored. This is due to events between the focal system and its connections to the wider economy, which produce the emergent behaviour that is characteristic of complex systems. This has first been described by Neumann et al. (1944) in the attempt to explain the role of transactions, which previously were not included in theories of how the economy works[32]. As the field of SCM has been dominated by

[32] Commentary by Harold Kuhn in the 60th Anniversary edition of Neumann et al. (1944).

a deterministic approach to decisions and has assumed some equilibrium state, one could argue that in this sense, all transactions are events that are disruptive to the system. The notion of how a sequence of interactions may lead to an emergent future state implies that the future is uncertain and therefore incomputable (Prokopenko et al. 2009). General patterns may be deduced from the rule sets by which the interactions happen, such as convergence, but the outcome for the atomic entity in the system must be considered unpredictable (Rosen 1978; Dooley 1997). This is important as a prevailing assumption for optimisation is, that optimisation cannot happen, without perfect information. In the real-worlds complex economic system, however perfect information can only be available in retrospect. Not in real-time. So to claim that the most productive intervention also needs to be the global retrospective optimum is from a complex systems perspective (Prokopenko et al. 2009; Rzevski 2011) a logical fallacy (Popper 2002). To argue of optimisation in complex systems can therefore only refer to *temporal optimality* (Simon 1978) as perfect information cannot be available. An attempt to illustrate this dilemma is given in Figure 43 below.



A decision maker who assigns resources only after receipt of information a_1, will be ignorant of information $\{a_2, \ldots, a_n\}$ which the environment has produced, but that has not yet propagated through the social networks to the decision maker. Thereby the consequence of making a decision which is based on information available at time a_1 will lead to a response from the environment r_1, which in turn will have the outcome o_1 for the decision maker. However at time o_1, the decision maker will have had the possibility to include information $\{i_2, i_3, \ldots, i_n \leq o_1\}$ which may render the decision obsolete, upon which action a_1 was based. This problem is described by Simon (1978) but not formalized as illustrated above.

Figure 43 Information exchange between an agent and the environment.

With this background, speaking of "optimisation" using batch-based systems is an oxymoron under conditions of complexity as queueing of information for batch processing implies "not responding to an event until later" and there accepting that the information will be out-of-date.

To be explicit about the conditions of complexity, the thesis includes that any and all systems where:

- The relationships (dependencies and connectivity) of the system may evolve.
- The decision making units have autonomy to collaborate or defect, including the operational spectrum between these two game-theoretical extrema.

- The system exhibits emergent properties, such as through innovation processes which change the capability of the system.
- The system is a state of non-equilibrium and therefore only provides the appearance of stability through self-organisation.
- The system can (co-)evolve as a response to events.
- The system changes non-linearly.

A system which exhibits these properties *is* complex. Complicated systems, on the other hand are in contrast considered deterministic, disregarding how many parts and subsystems they have. This implies that the complex system temporarily can exhibit behaviour which gives the appearance of a complicated system, through behaviour which reacts to events with limited propagation range. The seasoned scholar might want to dwell on this claim, as the author is tired of defending the idea that "complexity" is a specific set of system properties and often speculates "How hard can this be to understand?" A "complex system" is not a system that is in-transparent (visit "black box system" instead) nor – in its most appalling form – a system which the researcher simply finds "hard to understand" (For limited cognitive capacity visit Kahneman (2011)). For detailed examples on definitions of complexity visit Rzevski & Skobelev (2014).

9.2 The interactive role of the people

In section *3.3 Feedback loop* the question was raised on "what to do" when an agent in a distributed system requests information and is awaiting a response. The fact that people cannot be available to respond 24/7/365 – because, well – because humans are only humans. However as global supply chains operate with the time zones around the clock an implied solution was to establish agent-based models as proxies for people who make their assumptions explicit so that the proxy can respond and make decisions in their absence. Hereby two persons in different time zones can interact with one another's proxies and thereby have no need to wait for the other person to respond directly.

First hand experiences with supply chain planning shows us that it is all about which quantities of what SKU to what time. Coordination of this type of queries is computationally trivial. The development of an agent-based proxy which can resolve a significant amount of evaluations of numerical alternatives for planning will thereby not only contribute to a significant improvement in productivity, but also leave time to improve the decision making method by which the alternatives are evaluated by the proxy.

This makes deployment of the New Supply Chain Model (NSCM) an enabler for a meta-intervention at the level of decision making:

- Proxies in the NSCM evaluate information and make decision about alternatives.
- People engage in continuous improvement of the proxies.

Children and adults do this every day in computer games where they train to outsmart an agent based model within the rules of the game engine. In the NSCM, the role of people is to obtain information which needs to be considered but which the NSCM does not have access to. The translation of tacit knowledge, access to data and conversion these into rules and properties which contribute to improvement of realism of the NSCM is thereby the "new role" of the supply chain manager:

The role changes from the stressful task of making decisions under time pressure towards developing improved decision making methods for the proxy – mainly as a product of teamwork, as the improved insight requires tacit & experimental investigations.

That being said, the NSCM is suitable for simulation as well as transactional decision making. The former was shown in "*Case Study II: LBR*" and the latter in "*Case Study III: FMCG*". Reflecting upon Shen et al. (2006):

> "*Many researchers (particularly Ph.D. students) working on agent-based manufacturing are still focusing on the fundamental research to enhance the rationality or intelligence of software agents and develop more efficient and effective coordination and negotiation mechanisms. While this kind of research is important and still needed, we believe that the future R&D work should focus on the integration of agent-based planning and scheduling systems with existing systems used in manufacturing enterprises. The most important integration is with real time data collection systems, ... Another important integration is with existing ERP and MRP systems. Note that a certification is required for integrating or interfacing with some commercial ERP/MRP systems. **Only when such integrations are achieved and validated in industrial settings, will the agent technology be widely applied in manufacturing industry**.*" (Shen et al. 2006, p.427)

It can be argued that the New Supply Chain Model appears to have passed the "academic" tests as <u>*it is integrated*</u> in industrial settings. The revised role of the supply chain manager as "trainer" for the proxy is as a conclusion not science fiction. Revisit the figure below:

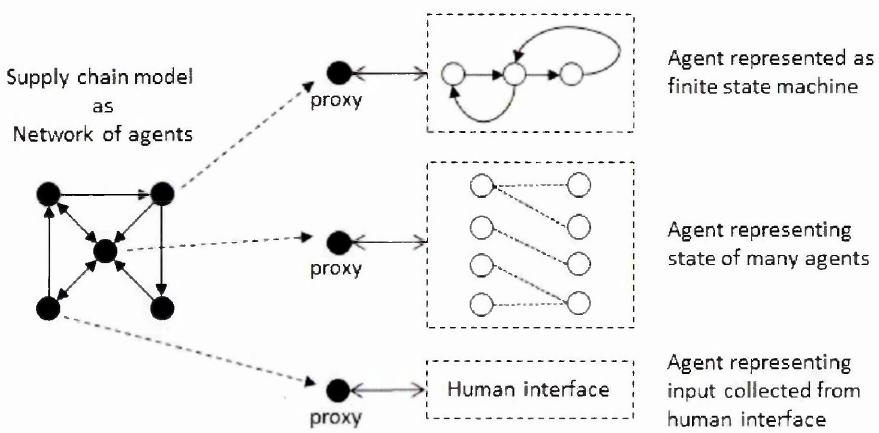


Figure 44 asynchronous communicating agents in a heterogenous agent environment

The operational capability which removes the human from being an obstacle in the *queueing network of information processors* and redeploys the human competencies at a higher order of intellectual engagement is a property of the designed system.

Reflecting upon the fact the NSCM provided evidence that reassigning people from making decisions to perform oversight and improving decision making methods governed by proxies leads to the conclusion that the principles of the modern organisation is about to change. The thesis is thereby not about people OR computing, but about the total latency of the system of network of queues that are moving closer to real-time.

The question of how to replicate these studies in similar industrial contexts shall be subject to commercialisation of the NSCM and future research.

Chapter 10 – Conclusions

Overview – This chapter presents the main conclusions and recommend issues for further research. It summarises the contribution of the work to, primarily, the supply chain management literature.

10.1 Main conclusions

In the context of the supply chain as a recognised example of a complex economic system the thesis asked: How does one make the most productive intervention in a complex economic system? Three decades of supply chain management have passed (1984-2014) where the intention has been to improve collaboration amongst business partners. Meta-interventions have proven themselves successful for at least 15 years. A wide range of IT systems were developed to support the coordination at the most detailed level. Network design, transactional- and functional-oriented models have been used to argue for improved collaboration, and simulation have been used to verify business cases.

Yet practice today is for customers to propagate their sub-optimised demand signals naively to suppliers and command compliance, which has nothing to do with collaboration. Customers who do not get their orders fulfilled introduce contracts with service level requirements and penalties for non-compliance, which results in the lowest level of service being provided. Compliance leaves no space for feedback. Without feedback, the solution space for making the most productive intervention is reduced from a solution landscape to a point. To comply stock is increased, unnecessarily expensive transport methods are used, and production lines stand idle. In summary – a practice where the customer penalises him-/herself and reduces the productivity of the supply chain that s/he is a part of.

The collective of information systems operate as a queueing network in which information processors (human and machine) are stressed to make decisions as quickly as possible. Software must run in microseconds and humans are given no time for reviewing the data that was sent to their email box a few minutes ago. Every minute spent in the queueing network delays a decision maker somewhere in the system as the information does not move. In other words: Supply Chain has for 30 years been focused on optimisation of the physical activities in the supply chain, and not the flow of information which triggers the activities.

The thesis investigated the impact of adding the supply-side feedback loop and sharing information effectively, so that it is not delayed from arriving to those who need it.

The three 3 case studies found:

- With LEGO System that delay was the cause of 80m EUR of stock not reaching the retailers in time.
- With LEGO Brand Retail that usage of outdated data and premature commitment of resources, resulted in stock being sent to the wrong stores or

in inadequate quantities, causing a 24%-point increase in lost sales and nearly 25%-point drop in profitability.
- With a FMCG megabrand the lack of ability to share information about inventory within the same department resulted in order fulfilment rate 22% lower than needed.

Three findings stand out:

First, that models of the SC "optimisation" have been focused on the physical logistics, and not the information which triggers the logistic activities. This is almost embarrassing, as precise calculations with outdated information produces conclusions which do not match the requirements of the environment.

Second, that the distributed nature of decision making in supply chains require rigorous evaluation of the feedback that suppliers can give, as naïve propagation does not result in any optimisation at all. Optimisation without inclusion of feedback is abuse of mathematics to prove optimality of a solution to a problem that the supply chain is not concerned with.

Third, that five information processing strategies can reduce the problem significantly:

1. Perform distributed decision making. Not propagation of decisions. There are no technical obstacles for this.
2. Real-time decision making. Do not delay sharing of information.
3. Commit to resource allocation with maximum delayed commitment. Avoid premature commitment as this otherwise locks resources from being deployed elsewhere.
4. Update only resource allocations that are affected by changes (exception or delta approach). Complete rescheduling causes more disruptions, than it resolves coordination problems.
5. Exchange messages to solve problem. Do not use search!

These findings were implemented as the New Supply Chain Model, which takes its outset in the flow of information, so that it may include feedback and through negotiation can identify "optimality".

The challenge with implementation in two of the case studies found that agent based models are superior in detailed decision making, whilst it is much more productive to let people with access to tacit information and experiment on the higher order role by developing options which can be rigorously exploited by the agent based model/computer.

The conclusion to the research question "How to make the most productive intervention in a complex economic system" remains as a thesis a meta-intervention which deploys a set of strategies, which elevates the role of the human decision maker from "making decisions" to "improving the decision making method" and at the same time use computerised agent based models (ABM) for operational decision-making. In addition, and in contrast to previous research on supply chain optimisation and as a contribution to knowledge, this research project found the flow of information and, in particular, the delay of information as a significant factor for creating the most productive intervention.

10.2 Vision for further research

During the thesis several opportunities for further research were raised. In summary:

- SC models which are not based on maximisation of profitability, will be wrong as (i) revenue maximisation alone drives unprofitable activities, just as narrow (ii) cost focus will lead to short sighted exploitation. Identification of the right balance will be important for long term investments.
- Future R&D work should focus on the integration of agent-based planning and scheduling systems with existing systems used in manufacturing enterprises. The most important integration is with real time data collection systems, ... (Shen et al. 2006, p.427)
- It is realized that research at the inter-organizational level is less prevalent. The objective of SCM is to integrate ... and treat them as a single entity (global supply chain). (Badole et al. 2012, p.75)
- Memoization of results for update without the need for batch processing provides a significant source of speed up (Burckhardt et al. 2011).
- The observation that the evolution of solutions in multi-agent system bears resemblance of the mechanisms observable in the immune system (Kauffman 1993).
- How to provide tools to educate software developers in development of distributed systems, such that the effort of domain experts may be spend on solving modelling problems and not searching for coding errors.
- How to increase the common usage of GPGPU for parallel computing.

Most notably none of the literature presented a model which takes the game theoretic predicate at its foundation[33]. No rigorous stance towards assumptions for all data analysis is given, whereby the publications remain implicitly naïve about the subject of whether the agents of the supply chain remain loyally collaborative or may have more incentive chose to otherwise. From dialogue with doctoral students, associate professors and professors the general understanding of game-theory as a fundamental evaluation of options appear deeply misunderstood. An example hereof is that studies of time series presume that the parties in the system continues to behave the same way. However when everyone behaves the same way the incentive to defect increases, whereby the fundamental assumptions of time-series analysis are no longer valid and only the game-theoretical foundations of the strategic choice remains. As a researcher, this is cause for deep concerns, as it means that Supply Chains are focused on exploitation and risk management in the physical logistics and generally ignorant of the source of origin of the information.

The advice for future research in system design is thereby to include explicit game theoretical predicates before focusing on optimisation of given performance metrics. A very simple example of this is to include the consideration that unless a supplier is profitable, it is highly likely that they may reduce order fulfilment, resulting in reduced ability to fulfil key customer orders of the focal supply chain.

[33] From dialogue with doctoral students, associate - and professors, I have quietly developed the impression that the misunderstanding of game theoretical foundations remains my deepest disappointment throughout this research project.

10.3 Summary of contributions

The thesis contains a set of contributions:

1. **Critique of existing approaches to supply chain optimisation.** The critique took a strong departure from claims of optimisation based on out-dated data, and highlighted the problems that follows when retrospective analysis is used to describe potential opportunities, as decisions made in the real-world must be chronologically progressive. As retrospective analysis is used all over the world by consultancies, this has significant impact for practice.
   The critique also raised awareness of the delay in information exchange which occurs when a network of batch-processors is used as transmission medium for information. The contribution of chapters 2 & 3 to supply chain management literature makes the case for change explicit, as the research focus must shift from intervening in the physical activities towards *intervening in the information* which triggers the physical activities.
2. **Recasting of the supply chain coordination problem as Agent Based Model** which is *c-competitive* and batch processing free (chapter 6.1 & 6.2). To be exact the "new (agent based) supply chain model" permits:
   a. Multi-objective optimisation under chronologically progressive elapse,
   b. Asynchronous exchange of information, and,
   c. Resolves the problem associated with "information processing in a queueing network" which cause a costly delay of information.

   And only depends on implementation of three key strategies:
   a. Maintain a solution such that the error term of responding to changes is minimised.
   b. Operate with maximum delayed commitment
   c. Maximize external connectivity.
3. **The detailed specification *new supply chain model*** (which subsequently was implemented by professional software engineers) contributed to transfer of the role of the Supply Chain Manager from decision making, to oversight and continuous improvement of the agent-based proxy which makes decisions. This addresses issues raised by Melo et al. 2009; Badole et al. 2012; Shen et al. 2006 and creates a novel form of meta-intervention which is complimentary to Goldratt & Cox 2004; Womack 2008; Christopher & Rutherford 2004 (Chapter 3).

Two case studies which illustrated significant influence of delay of information on selected performance metrics (chapters 7 & 8). Two different large scale real-life well-known global corporations sponsored and verified the model, where the author both collected and collated the data and ran the simulations, which were audited by a global supply chain consultancy. In this context each case study addresses Stadtler & Kilger (2005) notion that "the last years many SCM projects and APS implementations failed or at least did not fully meet expectations"

(Chapter 2.3) as the delay in information processing will inhibit the synergies pursued by SCM & APS.

References

Van der Aalst, W.M., 1998. The application of Petri nets to workflow management. *Journal of circuits, systems and computers*, 8(1), pp.21–66.

Aho, A. V., Hopcroft, J. & Ullman, J., 1974. *The design and analysis of algorithms*,

Aitken, J.M., 1998. *Supply Chain Integration within the Context of a Supplier Association: Case Studies of Four Supplier Associations*. Cranfield University.

Ajtai, M. et al., 1994. A theory of competitive analysis for distributed algorithms. In *35th Annual Symposium on Foundations of Computer Science*. IEEE, pp. 401–411.

Akanle, O.M.M. & Zhang, D.Z.Z.Ã., 2008. Agent-based model for optimising supply-chain configurations. *International Journal of Production Economics*, 115(2), pp.444–460.

Albert, R., Jeong, H. & Barabasi, A.-L., 2001. Errata: Error and attack tolerance of Complex Networks. *Nature*, 409.

Albert, R., Jeong, H. & Barabasi, A.-L., 2000. Error and attack tolerance of complex networks. *Nature*, 406(6794), pp.378–382.

Allan, R., 2009. *Survey of Agent Based Modelling and Simulation Tools*,

Allesina, S. et al., 2010. Performance measurement in supply chains: new network analysis and entropic indexes. *International Journal of Production Research*, 48(8), pp.2297–2321.

Andreev, M. et al., 2007. Adaptive Planning for Supply Chain Networks. In *HoloMAS*. pp. 215–224.

Angerhofer, B.J. & Angelides, M.C., 2000. System Dynamics Modelling in supply chain management: Research Review. In J. A. Joines et al., eds. *Proceedings of the 2000 Winter Simulation Conference*. pp. 342–351.

Anosike, A.I. & Zhang, D.Z.Ã., 2007. An agent-based approach for integrating manufacturing operations. *International Journal of Production Economics*, 121(2), pp.333–352.

Arora, S. & Barak, B., 2009. *Computational complexity: a modern approach*, Cambridge University Press.

Aspnes, J., 1998. Competitive analysis of distributed algorithms. *Online Algorithms*, pp.118–146.

Astor, J.C. & Adami, C., 2000. A Developmental Model for the Evolution of Artificial Neural Networks. In *Artificial life*. pp. 189–218.

Axelrod, R., 1997. *The complexity of cooperation: agent-based models of competition and ...*, Princeton University Press.

Axtell, R.L. et al., 1996. Aligning simulation models: A case study and results. *Computational and Mathematical Organization Theory*, 1(2), pp.123–141.

Badole, C.M. et al., 2012. Research and Opportunities in Supply Chain Modeling: A Review. *International Journal of Supply Chain Management*, 1(3), pp.63–86.

Bakhouya, M. & Gaber, J., 2007. An immune inspired-based optimization algorithm: Application to the traveling salesman problem. *Advanced Modelling and Optimization*, 9(1), pp.105–116.

Barabási, A.L. & Bonabeau, E., 2003. Scalefree networks. *Scientific American*, 288(May), pp.50–59.

Beer, S., 1984. The Viable System Model: Its Provenance, Development, Methodology and Pathology. *Journal of the Operational Research Society*, 35(1), pp.7–25.

Beinhocker, E.D., 2007. *The origin of wealth: evolution, complexity, and the radical remaking of ...*, Random House Business Books.

Bellman, R.E., 1986. *The Bellman Continuum: A Collection of the Works of Richard E. Bellman*, World Scientific.

Bertsekas, D.P., 1979. *A distributed algorithm for the assignment problem*,

Bertsekas, D.P., 1992. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1(1), pp.1–54.

Bertsekas, D.P. & Castañon, D.A., 1991. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 17(6), pp.707–732.

Bidot, J. et al., 2008. A theoretic and practical framework for scheduling in a stochastic environment. *Journal of Scheduling*, 12(3), pp.315–344.

Biswas, W. & Narahari, Y., 2004. Object oriented modeling and decision support for supply chains. *European Journal of Operational Research*, 153, pp.704–726.

Blume, L.E. & Durlauf, S.N., 2005. *The Economy as an Evolving Complex System, III: Current Perspectives and Future Directions: v. 3 (Santa Fe Institute Studies on the Sciences of Complexity)*, Oxford University Press.

De Boer, P.T. & Boer, P.T. de, 2000. *Analysis and efficient simulation of queueing models of telecommunication systems*. Centre for Telematics and Information Technology, University of Twente.

Bollen, J., Mao, H. & Zeng, X.-J., 2010. Twitter mood predicts the stock market . *Computer*, 1010(3003v1), pp.1–8.

Borrill, P.L. & Tesfatsion, L.S., 2010. Agent-Based Modeling: The Right Mathematics for the Social Sciences? *Staff General Research Papers*. Available at: http://www.econ.iastate.edu/tesfatsi/ABMRightMath.PBLTWP.pdf [Accessed November 20, 2010].

Brintrup, A., 2010. Behaviour adaptation in the multi-agent , multi-objective and multi-role supply chain. *Computers in Industry*, 61(7), pp.636–645.

Brooks, F., 1995. *The mythical man-month: Essays on software engineering*, Addison Wesley Longman, Inc.

Burckhardt, S. et al., 2011. Two for the price of one: A model for parallel and incremental computation. *ACM SIGPLAN Notices*, 46(10), pp.427–444.

Burgess, K., Singh, P.J. & Koroglu, R., 2006. Supply chain management: a structured literature review and implications for future research. *International Journal of Operations & Production Management*, 26(7), pp.703–729.

Caro, F. & Martinez-de-Albeniz, V., 2010. The impact of quick response in inventory-based competition. *Manufacturing & Service Operations Management*, 12(3), pp.409–429.

Cavana, R.Y., Delahaye, B.L. & Sekaran, U., 2001. *Applied Business Research: Qualitative and Quantitative Methods*, John Wiley & Sons. Available at: http://www.amazon.co.uk/Applied-Business-Research-Qualitative-Quantitative/dp/0471341266 [Accessed November 25, 2010].

Chaib-draa, B. & Muller, J.P., 2010. *Multiagent-Based Supply Chain Management*, Springer.

Chan, F.T.S. & Chung, S.H., 2004. A multi-criterion genetic algorithm for order distribution in a demand driven supply chain. *International Journal of Computer Integrated Manufacturing*, 17(4), pp.339–351.

Chan, H.K. & Chung, S.H., 2013. Optimisation approaches for distributed scheduling problems. *International Journal of Production Research*, 51(9), pp.2571–2577.

Chatfield, D.C., Hayya, J.C. & Harrison, T.P., 2007. A multi-formalism architecture for agent-based , order-centric supply chain simulation. *Simulation Modeling Practice and Theory*, 15, pp.153–174.

Chen, P. et al., 2007. Finding scientific gems with Google's PageRank algorithm. *Journal of Informetrics*, 1(1), pp.8–15.

Choi, T.Y., Dooley, K.J. & Rungtusanatham, M., 2001. Supply networks and complex adaptive systems: control versus emergence. *Journal of Operations Management*, 19(3), pp.351–366.

Christopher, B.M. & Rutherford, C., 2004. Supply Chain Six Sigma. , (August), pp.24–28.

Christopher, M., 2001. An integrated model for the design of agile supply chains. *International Journal Of Physical Distribution*, pp.1–17.

Christopher, M., 2004. Building the resilient supply chain. *International Journal of Logistics*, p.xxx.

Christopher, M., 2005. *Logistics and supply chain management: creating value-added networks*, Pearson Education.

Christopher, M. & Gattorna, J., 2005. Supply chain cost management and value-based pricing. *Industrial marketing management*, 34(2), pp.115–121.

Christopher, M. & Lee, H., 2004. *Mitigating supply chain risk through improved confidence*,

Coello, C.A.C., 2000. An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys (CSUR)*, 32(2), pp.109–143.

Coello, C.A.C., 2006. Evolutionary Multi-Objective Optimization: A Historical View of the Field. *IEEE Computational Intelligence Magazine*, (February), pp.28–36.

Conway, M.E., 1963. Design of a Separable Transition-Diagram Compiler. *Communications of the ACM*, 6(7), pp.396–408.

Cooper, M.C., Lambert, D.M. & Pagh, J.D., 1997. Supply Chain Management: More than a New Name for Logistics. *The International Journal of Logistics Management*, 8(1), pp.1–14.

Cormen, T.H.T.H. et al., 2009. *Introduction to algorithms* 2nd ed., The MIT press.

Croom, S., Romano, P. & Giannakis, M., 2000. Supply chain management: an analytical framework for critical literature review. *European Journal of Purchasing & Supply*, 6, pp.67–83.

Cunningham, J.E. & Fiume, O.J., 2003. *Real Numbers: Management Accounting in a Lean Organization*, Managing Times Press.

Daley, D.J. & Gani, J.M., 2001. *Epidemic Modelling: An Introduction*, Cambridge University Press.

Dezső, Z., Barabási, A.-L.L. & Barabasi, A.-L., 2002. Halting viruses in scale-free networks. *Physical Review E*, 65(5), pp.1–4.

Dickersbach, J.T., 2009. *Supply Chain Management with SAP APOTM: Structures, Modelling Approaches and Implementation of SAP SCMTM 2008 (Google eBook)*, Springer.

Dimitrov, T. & Baumann, M., 2011. Genetic algorithm with genetic engineering technology for multi-objective dynamic job shop scheduling problems. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. pp. 833–834.

Dooley, K.J., 1997. A Complex Adaptive Systems Model of Organization Change. *Nonlinear Dynamics Psychology and Life Sciences*, 1(1), pp.69–97.

Dooley, K.J., 2009. the Empiricism-Modeling Dichotomy in Operations and Supply Chain Management *. *Journal of Supply Chain Management*, 45(1), pp.38–43.

Dorigo, M., 1992. *Ant colony optimization*, WikiBooks.

Dugdale, D. & Jones, T.C., 1998. Throughput accounting: Transforming practives? *British Accounting Review*, 30, pp.203–220.

Ellram, L.M. & Cooper, M.C., 2014. Supply Chain Management: It's all about the journey, not the destination. *Journal of Supply Chain Management*, 50(1), pp.8–20.

Engelbrecht, A.P., 2005. *Fundamentals of Computational Swarm Intelligence*, Wiley.

Fidanova, S., 2005. Heuristics for multiple knapsack problem. In *IADIS International Conference on Applied Computing 2005*. pp. 255–260.

Fowler, J.W. & Rose, O., 2004. Grand challenges in modeling and simulation of complex manufacturing systems. *Simulation*, 80(9), pp.469–476.

Fox, M.S., Barbuceanu, M. & Teigen, R., 2000. Agent-Oriented Supply-Chain Management. *The International Journal of Flexible Manufacturing Systems*, (12), pp.165–188.

Fu, M.C., 2002. Optimization for Simulation: Theory vs. Practice. *Journal on Computing*, 14(3), pp.192–215.

Gath, M., Edelkamp, S. & Herzog, O., 2013. Agent-Based Dispatching in Groupage Traffic. *Computational Intelligence in Production and Logistics Systems*, pp.54–60.

Gattorna, J., 2006. *Living Supply Chains: How to Mobilize the Enterprise Around Delivering What Your Customers Want*, Pearson Education.

Geri, N. & Ronen, B., 2005. Relevance lost : the rise and fall of activity-based costing. *Human Systems Management*, 24, pp.133–144.

Ghosh, A. & Dehuri, S., 2004. Evolutionary algorithms for multi-criterion optimization: A survey. *International Journal of Computing & Information*, 2(1), pp.38–57.

Giovannucci, A. et al., 2007. Mixed multi-unit combinatorial auctions for supply chain management. *ACM SIGecom Exchanges*, 7(1), pp.58–60.

Goldratt, E.M. & Cox, J., 2004. *The Goal: A Process of Ongoing Improvement*, North River Press Publishing Corporation.

Goldstone, J.A. et al., 1985. Cort, Goldstone,., pp.1439–1443.

Grljevic, O. & Bosnjak, Z., 2011. Time series analysis by neural networks as support to strategic planning. *Strategic Management*, 16(2), pp.46–55.

Gunasekaran, A. & Ngai, E.W.T., 2005. Build-to-order supply chain management: a literature review and framework for development. *Journal of Operations Management*, 23, pp.423–451.

Gunasekaran, A. & Ngai, E.W.T., 2004. Information systems in supply chain integration and management. *European Journal of Operational Research*, 59(1), pp.269–295.

Harrington, J.E., 2008. *Games, strategies, and decision making*, Worth.

Harrison, A. & Hoek, R.I. van, 2008. *Logistics management and strategy: competing through the supply chain*, Pearson Education.

Herlihy, M. & Shavit, N., 2012. *The Art of Multiprocessor Programming*,

Holmgren, J., 2008. *Multi-agent-based simulation and optimization of production and transportation*, Blekinge Institute of Technology Licentiate Dissertation Series, School of Engineering.

Horn, J., Nafpliotis, N. & Goldberg, D.E., 1994. A niched Pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress on Computational Intelligence*. IEEE, pp. 82–87.

Huang, G.Q., Lau, J.S.K. & Mak, K.L., 2003. impacts of sharing production information on supply chain dynamics: a review of literature. *International Journal of Production Research2*, 41(7), pp.1483–1517.

Huang, G.Q. & Zhang, X.Y., 2007. Integrated configuration of platform products and supply chains for mass customization: a game-theoretic approach. *IEEE Transactions on Engineering Management*, 54(1), pp.156–171.

Iba, T. et al., 2002. Resolving the Existing Problems by Boxed Economy Simulation Platform. In A. Namatame, T. Takao, & K. Kurumatani, eds. *Agent-Based Approaches in Economic and Social Complex Systems*. IOS Press, pp. 59–68.

Ilie, S. et al., 2010. Effectiveness of solving traveling salesman problem using ant colony optimization on distributed multi-agent middleware. In *Proceedings of the International Multiconference on Computer Science and Information Technology*. pp. 197–203.

Iridia, T.R. et al., 1996. Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. , 1(1), pp.1–24.

Iridia, T.R. et al., 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE*, 1(1), pp.1–24.

Ivanov, D. & Sokolov, B., 2013. Dynamic co-ordinated scheduling in the supply chain under a process modernisation. *International Journal of Production Research*, 51(9), pp.2680–2697.

Ivanov, D., Sokolov, B. & Kaeschel, J., 2010. A multi-structural framework for adaptive supply chain planning and operations control with structure dynamics considerations. *European Journal of Operational Research*, 200(2), pp.409–420.

Johnson, J., 2012. Hypernetworks for Policy Design in Systems of Systems of Systems. In K. Glass et al., eds. *Proceedings of COMPLEX 2012. Santa Fe. 2nd International Conference on Complex Sciences: Theory and Applications. Dec 5-7 2012*. Springer, pp. 179–189.

Johnson, S.M., 1954. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1), pp.61–68.

Kahneman, D., 2011. *Thinking, Fast and Slow*, Penguin UK.

Kaplan, R.S. & Anderson, S.R., 2004. Time-driven activity-based costing. *Harvard business review*, 82(11), pp.131–138,150.

Kaplan, S. & Anderson, S.R., 2003. *Time-Driven Activity-Based Costing Robert*,

Karp, R.M., 1972. Reducibility among combinatorial problems. In R. E. Miller & J. W. Thatcher, eds. *Complexity of Computer Computations*. Plenum Press, pp. 85–103.

Katz, D.S., 2013. Citation and Attribution of Digital Products: Social and Technological Concerns. Available at: http://figshare.com/articles/Citation_and_Attribution_of_Digital_Products_Social_and_Technological_Concerns/791606 [Accessed October 3, 2013].

Kauffman, S.A., 1995. Technology and evolution: Escaping the red queen effect. *McKinsey Quarterly*, (1), pp.118–129.

Kauffman, S.A., 1993. *The Origins of Order*, Oxford University Press.

Kirkpartick, S., Gelatt Jr., C.D. & Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, 220(4598), pp.671–680.

Kleinberg, J. & Easley, D., 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press. Available at: http://www.amazon.co.uk/Networks-Crowds-Markets-Reasoning-Connected/dp/0521195330 [Accessed March 20, 2013].

Klemmt, A. et al., 2009. Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. *Robotics and Computer-Integrated Manufacturing*, 25(6), pp.917–925.

Kogan, K. & Tapiero, C.S., 2007. *Supply Chain Games*, Springer.

Konak, A., Coit, D.W. & Smith, A.E., 2006. Multi-objective optimization using genetic algorithms: A Tutorial. *Reliability Engineering and System Safety*, pp.992–1007.

Kopetz, H., 2011. *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Springer; 2nd Edition. edition.

Lambert, D.M. & Cooper, M.C., 2000. Issues in Supply Chain Management. *Industrial marketing management*, 29, pp.65–83.

Lambert, D.M., Cooper, M.C. & Pagh, J.D., 1998. Supply Chain Management: Implementation Issues and Research Opportunities. *The International Journal of Logistics Management*, 9(2), p.19.

Laplante, P.A. & Ovaska, S.J., 2011. *Real-Time Systems Design and Analysis: Tools for the Practitioner [Hardcover]*, Wiley-Blackwell; 4th Edition edition.

Lau, J.S.K. et al., 2006. Agent-based modeling of supply chains for distributed scheduling. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(5), pp.847–861.

Leitão, P., 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7), pp.979–991.

Leitao, P. & Vrba, P., 2011. Recent Developments and Future Trends of Industrial Agents. In *HoloMAS*. pp. 15–28.

Leung, J.Y.-T., 2004. Handbook of Scheduling Algortihms., p.1157.

Lidestam, H. & Ronnqvist, M., 2011. Use of Lagrangian decomposition in supply chain planning. *Mathematical and computer modelling*, 54(9-10), pp.2428–2442.

Madsen, B. et al., 2012. Real-Time Multi-agent Forecasting and Replenishment Solution for LEGOs Branded Retail Outlets. In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. IEEE, pp. 451–456.

Martin, L.R., 2013. Rethinking the Decision Factory - Harvard Business Review. *Harvard business review*, Reprint R1(October), pp.1–9.

Martinez, S.Z. & Coello, C.A.C., 2011. Swarm intelligence guided by multi-objective mathematical programming techniques. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation - GECCO '11*. pp. 771–774.

Matuszek, J. & Mleczko, J., 2009. Specificity of bottlenecks in conditions of unit and small-batch production. *Journal of Automation, Mobile Robotics & Intelligent Systems*, 3(3), pp.102–109.

Max Gath, Stefan Edelkamp, and O.H., 2013. Optimal Decision Making in Agent-Based Autonomous Groupage Traffic. *Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART)*, pp.248–254.

Meixell, M.J. & Gargeya, V.B., 2005. Global supply chain design: A literature review and critique. *Transportation Research*, pp.531–550.

Melo, M.T., Nickel, S. & Saldanha-da-Gama, F., 2009. Facility location and supply chain management - A review. *European Journal of Operational Research*, 196, pp.401–412.

Menezes, M. & Barabasi, A.-L., 2008. Fluctuations in network dynamics. *Nature*, 46556(1), pp.1–4.

Mentzer, J.T. et al., 2001. Defining supply chain management. *Journal of Business Logistics*, 22(2), pp.1–25.

Meuleau, N. & Dorigo, M., 2002. Ant colony optimization and stochastic gradient descent. *Artificial life*, 8(2), pp.103–21.

Michie, D., 1968. Memo functions and machine learning. *Nature*, 218(5136), pp.19–22.

Miller, J.H. & Page, S.E., 2007. *Complex adaptive systems: an introduction to computational models of social life*, Princeton University Press.

Mintzberg, H., 1983. *Structure in Fives: Designing Effective Organizations*, Prentice Hall PTR.

Mohemmed, A.W., Sahoo, N.C. & Goek, T.K., 2007. A New Particle Swarm Optimization Based Algorithm for Solving Shortest-Paths Tree Problem. In *IEEE Congres on Evolutionary Computation*. pp. 3221–3225.

Mönch, L. et al., 2011. A survey of challenges in modelling and decision-making for discrete event logistics systems. *Computers in Industry*, 62(6), pp.557–567.

Moon, Y.B. & Phatak, D., 2005. Enhancing ERP system's functionality with discrete event simulation. *Industrial Management & Data Systems*, 105(9), pp.1206–1224.

Nair, A. & Vidal, J.M., 2011. Supply network topology and robustness against disruptions – an investigation using multi-agent model. *International Journal of Production Research*, 49(5), pp.1391–1404.

Nash, J.F., 1950. Equilibrium Points in n-Person Games. *Proceedings of the National Academy of Sciences*, 36(1), pp.48–49.

Neagu, N. et al., 2006. Solving Distributed Delivery Problems with Agent-Based Technologies and Position Statement. *AAAI Spring Symposium: Distributed Plan and Schedule Management*, pp.159–160.

Nemirovski, A. et al., 2009. Tobust Stochastic Approximation Approach to Stochastic Programming. *SIAM Journal on Optimization*, 19(4), p.1574–.

Neumann, J. Von et al., 1944. *Theory of games and economic behavior* 60th anniv., Greenwood Publishing Group.

Neumann, J. Von, Churchland, P.M. & Churchland, P.S., 2000. *The computer and the brain*, Yale University Press.

Oliveira, A. & Gimeno, A., 2014. *Supply Chain Management Strategy: Using SCM to Create Grater Corporate Efficiency and Profits* kindle., Pearson Education.

Oliver, K., Samakh, E. & Heckmann, P., 2007. Rebuilding Lego, Brick by Brick. *Strategy+Business*, p.reprint 07306.

Oliver, K.R. & Weber, M.D., 1982. *Supply Chain Management: Logistics catches up with Strategy* M. Christopher, ed., London: Chapman & Hall.

Özkarabacaka, B., Çevikb, E. & Gökşen, Y., 2014. A comparison analysis between ERP and EAI. *Procedia Economics and Finance 9*, 9(Ebeec 2013), pp.488–500.

Parsons, S. & Klein, M., 2011. Auctions and bidding : A guide for computer scientists. *ACM Computing Surveys (CSUR)*, 43(2), p.10.

Pathak, S.D. et al., 2007. Complexity and Adaptivity in Supply Networks: Building Supply Network Theory Using a Complex Adaptive Systems Perspective. *Decision Sciences*, 38(4), pp.547–580.

Pepper, M.P.J. & Spedding, T.A., 2010. The evolution of lean Six Sigma. *International Journal of Quality Reliability Management*, 27(2), pp.138–155.

Poli, R., Langdon, W.B. & Mcphee, N.F., 2008. *A Field Guide to Genetic Programming with contributions by*,

Polya, G., 1945. *How to solve it: A new aspect of mathematical method*, Princeton University Press.

Popper, K.R., 2002. *The logic of scientific discovery*, Routledge.

Porter, M.E., 2008. *Competitive Strategy: Techniques for Analyzing Industries and Competitors*, Amazon Kindle.

Power, D.J. & Sharda, R., 2007. Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, 43(3), pp.1044–1061.

Prokopenko, M., Boschetti, F. & Ryan, A.J., 2009. An information-theoretic primer on complexity, self-organisation and emergence. *Complexity*, 15(1), pp.11–28.

Raisinghani, M.S. et al., 2005. Six Sigma: concepts, tools, and applications. *Industrial Management & Data Systems*, 105(4), pp.491–505.

Ramsay, J. & Croom, S., 2008. The impact of evolutionary and developmental metaphors on Purchasing and Supply Management: A critique. *Journal of Purchasing and Supply Management*, 14(3), pp.192–204.

Robbins, H. & Monro, S., 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3), p.400–.

Rosen, R., 1978. Review of: "SELF-ORGANIZATION IN NONEQUILIBRIUM SYSTEMS", by G. Nicolis and I. Prigogine, John Wiley, New York, 1977, 491 pp. *International Journal of General Systems*, 4(4), pp.266–269.

Rosen, S., 1981. The economics of superstars. *The American Economic Review*, pp.845–858.

Runarsson, T.P. & Yao, X., 2000. Stochastic Ranking for Constrained Evolutionary Optimization. *Transactions on Evolutionary Computation*, 4(3), pp.284–294.

Russell, S. & Norvig, P., 2009. *Artificial Intelligence: A Modern Approach (Prentice Hall Series in Artificial Intelligence)*, Prentice Hall.

Rzevski, G., 2011. A Practical Methodology for Managing Complexity. *Complexity*, 13(1-2), pp.38–56.

Rzevski, G., 1998. Elephants Rarely Pirouette (ERP). *Europe*, 6(9), pp.1–7.

Rzevski, G. & Skobelev, P., 2007. Emergent Intelligence in Large Scale Multi-Agent Systems. *International Journal of Education ....* Available at: http://www.magenta-technology.ru/docs/conferences/EIT_Journal/Emergent Intelligence Paper EIT Journal.pdf [Accessed November 11, 2012].

Rzevski, G. & Skobelev, P., 2014. *Managing Complexity* 1st Editio., WIT Press.

Shapiro, J.F., 2007. *Modeling the supply chain*, Thomson Brooks/Cole.

Shen, W., Yoon, H.J. & Norrie, D.H., 2006. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced Engineering Informatics*, 20, pp.415–431.

Shoham, Y., 1993. Agent-oriented programming. *Artificial Intelligence*, 60(1), pp.51–92.

Shoham, Y. & Leyton-Brown, K., 2008. *Multiagent systems: algorithmic, game-theoretic, and logical foundations*, Cambridge Univ Pr.

Siebers, P.-O. et al., 2010. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, 4(3), pp.204–210.

Silva, C. et al., 2002. Ant colonies as logistic processes optimizers. In *ANTS 2002, LNCS 2463*. Springer, pp. 76–87.

Da Silveria, G., Borenstein, D. & Fogliatto, F.S., 2001. Mass customization: Literature review and research directions. *International Journal of Production Economics*, 72(1), pp.1–13.

Simon, H.A., 1978. Rationality as Process and as Product of Thought. *The American Economic Review*, 68(2), pp.1–16.

Skobelev, P., 2011. Bio-Inspired Multi-Agent Technology for Industrial Applications. In E. A. M. and I. A. D. Faisal Alkhateeb, ed. *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications*. InTech, pp. 495–522.

Slak, A., Tavčar, J. & Duhovnik, J., 2011. Application of Genetic Algorithm into Multicriteria Batch Manufacturing Scheduling. *Mechanical Engineering*, 57(2), pp.110–124.

Smith, A., 2010. A Practical Multiagent Model for Resilience in Commercial Supply Networks. *Agent-Mediated Electronic Commerce*, p.14.

Snapp, S., 2009. *Discover SAP SCM*, SAP Press.

Srivastava, S.K., 2007. Green supply-chain management: A state-of-the-art literature review. *International journal of management reviews*, 9(1), pp.53–80.

Stadtler, H., 2005. Supply chain management and advanced planning - basic, overview and challenges. *European Journal of Operational Research*, 163, pp.575–588.

Stadtler, H. & Kilger, C., 2005. *Supply Chain Management and Advanced Planning* 3rd ed., Springer Verlag.

Swaminathan, J.M., Smith, S.F. & Sadeh, N.M., 1998. Modeling Supply Chain Dynamics: A Multiagent Approach. *Decision Sciences*, 29(3), p.607.

Tako, A.A. & Robinson, S., 2011. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision Support Systems2*, 52(4), pp.802–815.

Tesfatsion, L.S., 2006. Agent-based computational economics: A constructive approach to economic theory. In *Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics*. Elsevier, pp. 831–880.

Turgay, S., 2009. Agent-based FMS control. *Robotics and Computer-Integrated Manufacturing*, 25(2), pp.470–480.

Turing, A.M., 1936. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), pp.230–265.

Van De Ven, A.H., 2005. Alternative Approaches for Studying Organizational Change. *Organization Studies*, 26(9), pp.1377–1404.

Vidal, C.J. & Goetschalckx, M., 1997. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, pp.1–18.

Viswanadham, N. & Raghaven, N.S., 2000. Performance analysis and design of supply chains: A Petri net approach. *Journal of Operational Research Society*, 51, pp.1158–1169.

Vollmann, T.E. et al., 2005. *Manufacturing planning and control systems for supply chain management: The definitive guide for professionals* Fifth Edit., McGraw-Hill Professional.

Watson, K.J., Blackstone, J.H. & Gardiner, S.C., 2007. The evolution of a management philosophy: The theory of constraints. *Journal of Operations Management*, 25(2), pp.387–402.

Womack, J.P., 2008. *The Machine That Changed the World*, Simon & Schuster Ltd.

Wooldridge, M. & Jennings, N.R., 1998. Pitfalls of agent-oriented development. In K. P. Sycara & M. Wooldridge, eds. *AGENTS '98 Proceedings of the second international conference on Autonomous agents*. ACM New York, NY, USA, pp. 385–391.

Wooldridge, M.J., 2009. *An Introduction to MultiAgent Systems* 2nd ed., John Wiley & Sons, Ltd.

Wooldridge, M.J. & Ciancarini, P., 2001. Agent-oriented software engineering: The state of the art. In *Agent-oriented software engineering*. pp. 1–28.

Wu, S.D., Byeon, E.-S. & Storer, R.H., 1999. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations Research*, 47(1), pp.113–124.

Zhang, D.Z.Ã. et al., 2006. An agent-based approach for e-manufacturing and supply chain integration. *Computers & Industrial Engineering*, 51(2), pp.343–360.

Zhang, X., Koshimura, M. & Fujita, H., 2011. Combining PSO and local search to solve scheduling problems. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. pp. 347–354.

Appendices

A.1 Practical principles for ABM design (extension)

Detailed discussions of how the chosen design deals problems raised by the SCM literature, such as –

a) Accounting and the associated challenge of determining how to maximise order fulfilment and profitability;
b) Manufacturing Resource Planning (MRP) including the transformations of bills-of-materials and calculation of materials requirement planning; and,
c) How to connect the computer environment with human decision makers

– have been included in the appendices, as they may contribute to convince the reader of sufficiently comprehensive consideration of the required concepts.

A.1.1 Accounting principles

A common challenge in improving effectiveness and efficiency of supply chains is that our accounting practices tend to pool costs to reflect the invoices they drive. The dilemma this brings is that once costs have been pooled they are very difficult to break down to individual transactions which were prompted by commitments made in the supply chain. Typically methods are:

- Activity Based Costing pursues a segmentation of direct and indirect costs and allocates the indirect costs based on correlations. Though this approach does reveal some comparative averages, it commonly hides the outliers and ill designed processes (Geri & Ronen 2005).
- Time-Driven Activity Based Costing takes a similar approach but precision is again compromised by the granularity of time (Kaplan & Anderson 2003; Kaplan & Anderson 2004).
- Cost Accounting, which compares a total costs with total number of activities, giving the coarsest level we find documented (Cunningham & Fiume 2003).

These accounting practices make it very difficult to verify and improve decisions, as time commonly is spent on understanding the variances from the average cost and then performing secondary studies which again ought to improve the granularity of detail and visibility of what drives profits and losses. Often practitioners end up in vicious cycles where the data they use for their studies are outdated before they can reach a pragmatic conclusion about what to do. Notably, the problem is that the methods are all retrospective instead of being oriented towards the individual transaction where an allocation choice is made. This brings us to the practice of Throughput Accounting.

Throughput accounting

Throughput accounting is a method oriented towards making a choice about a pending option. Before diving into a detailed description of how this is used in distributed decision making, an example that compares Cost-Accounting with Throughput Accounting will provide more clarity of the concepts.

A supplier to an airplane manufacturer was offered a contract to make 16 cargo-plane bodies a year, using a design that requires special titanium installation, but none of the interior installation needed to produce a normal passenger plane (windows, etc.). The buyer offered to pay €1,150,000 per cargo-plane, and the company already had orders for 38 passenger planes for the year for € 1,435,000 per plane. Using cost-accounting the cost of operating the titanium vs. the installation would be:

Cost by Department	Total Cost	Man months per year	Cost per man month
Titanium works	***€ 29,930,000***	***492***	€ 60,833
Installations	***€ 13,530,000***	***612***	€ 22,108
Total	€ 43,460,000	1104	€ 39,366

Using cost accounting (as shown below) the calculation method would indicate that the company would lose money on any cargo plane produced. This is based on the analysis of the average estimated production costs.

Cost-Accounting Analysis	Cargo Plane	Passenger Plane
Annual Demand	***16***	***38***
Price (€)	1,150,000	1,535,000
Titanium Time (man months)	***12***	***8***
Installation Time (man months)	***6***	***16***
Total Time (man months)	18	24
Titanium Cost (€)	730,000	486,667
Installation Cost (€)	132,647	353,725
Raw Material Cost (€)	**392,000**	**306,000**
Total Cost (€)	1,254,647	1,146,392
Profit per Unit (€)	-104,647	388,608

However, when the capacity is included in the decision model – such as the sequence in which the work is performed – it may become visible that the production system is not fully utilized. In this example, the resources used for installation are fully utilized, even though the production setup only requires 304 of the available 492 man months. This is a common effect of automation in production facilities where load-balancing around bottleneck operations are difficult to schedule.

An analysis of the discrete choice, instead of using averages will reveal that the supplier could determine the profitability of products by calculating "throughput" (revenue minus variable cost) in each discrete case (accept or reject the new contract).

Throughput Accounting Analysis	Decline Contract	Accept Contract
Passenger planes	38	***32***
Cargo planes	**0**	16
Titanium Time (man months)	304	448
Installation Time (man months)	608	608
Revenue - Passenger planes (€)	58,330,000	49,120,000
Revenue – Cargo planes (€)	0	18,400,000

Raw Material Cost - Passenger plane (€)	11,628,000	9,792,000
Raw Material Cost - Cargo plane (€)	0	6,272,000
Throughput Value (€)	46,702,000	51,456,000
Overhead Expense (€)	-43,460,000	-43,460,000
Profit (€)	3,242,000	7,996,000

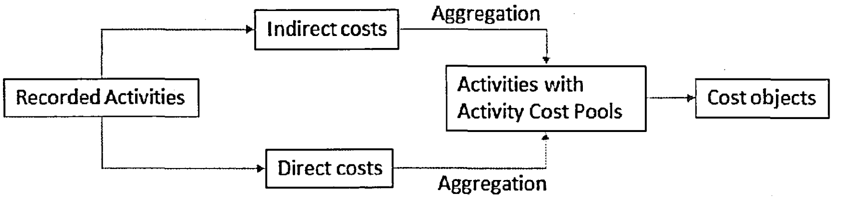
The simulation of the discrete choice (accept or reject) using explicit information about capacities within each node in the supply network allows for more appropriate decisions than averages allow for. In the example above, accepting the contract would yield an increase in the profit with 146%. This is important as throughput accounting is not yet a widely taught and used approach for valuations.

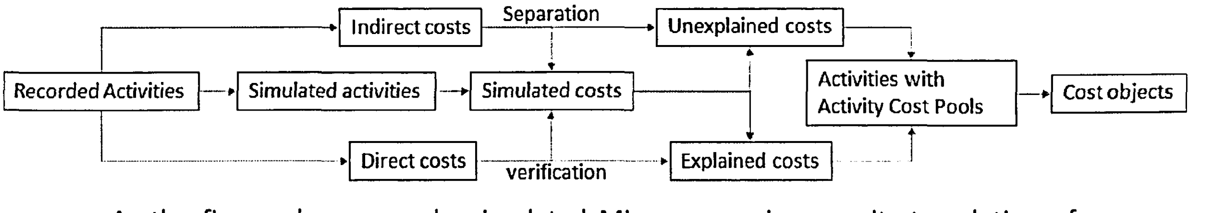
Simulated microeconomics

The nature of choice implies that a discrete event is pending, whereby the example above should make it clear to the reader why Throughput Accounting is the recommend approach when evaluating microeconomic decisions in distributed decision making systems instead of using Cost-Accounting.

By tracking individual choices, the objects, and their paths through the supply network based on the principles of Throughput Accounting, it is possible to evaluate combinations of options and maximize order fulfilment and profitability in a systematic and rigorous way. This means that it overcomes the challenge of cost-allocation as it provides transparency of how, when and where the items accumulate costs along their path until they leave the decision makers supply chain at the final point of sale. These paths of costs are referred to as cost-to-serve, cost-to-deliver and profitability at item level.

Below is an example (Madsen, 2007), of the conceptual differences between typical Activity-Based-Costing and Simulated Microeconomics.





As the figure above reveals, simulated Microeconomics permits translation of indirect and direct costs into explained and unexplained costs, for each transaction and thereby overcomes the challenge of allocating costs by using simulation.

This method gives the finest level of granularity, as the usage of simulation permits both verification of expected direct costs and separation of indirect costs into simulated costs and unaccounted costs. This results in much lower variance and can be done in real-time as pending decisions are changed or committed for execution.

Simulated Microeconomics permit, for example, the ability to verify if a transport invoice reflects the agreed transport rates, by simulating the transports which should be ordered based on the customer demand and the transport rates. Similarly, costs of using a 3rd party logistics provider, who sends just one invoice per month, can be simulated to assess whether the agreed activity based rates reflect the demand for them. Many managers do this in isolation in their respective departments using spreadsheets, but as the example describing that Throughput Accounting showed, it is difficult to make the right decision for the supply chain as a whole. In fact, we often see self-interested sub-optimisation.

The reason why this is not used at greater scale today is arguably because the amount of detail at the transactional level can be daunting and far beyond the accountants domain knowledge. In addition with common accounting practices, the workload associated with performing such modelling makes the process infeasible at large transaction volumes.

So far the assumption has been that maximisation of profits is the overarching assumption. Whilst this may be justified, it is not exhaustive, and does therefore not cover cases where investments contradict the idea of short term exploitation of opportunities. Investments may cover research, new facilities, and other costs which are expected to result in control over more resources on a longer term, than the return of investments that are initially indicated. These investments may increase productivity significantly, but are not profitable in the individual short term decision. This calls for a more nuanced interpretation of the concept of performing the "most productive intervention".

A.1.2 MRP

To evaluate the scalability of the job-shop example, the supply chain is extended to a hypothetical car manufacturer as illustrated in Figure 45, below:

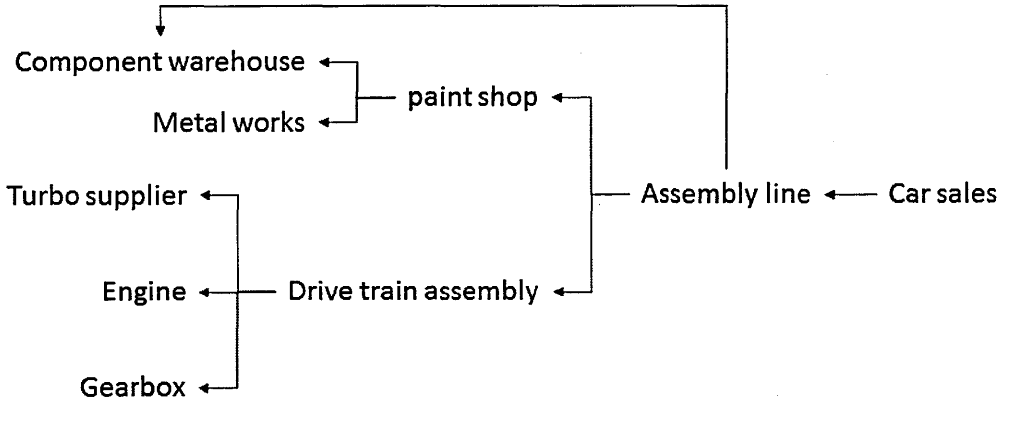


Figure 45 Small car manufacturer

The relationship between the entities may now be described as in the previous "job-shop" example where orders {a, b, c, d, e, f, g} were given:

Supply (M1)	Demand (M2)
Assembly line	Car sales
Paint shop	Assembly line
Drive train assembly	Assembly line
Component warehouse	Assembly line
Component warehouse	Paint shop
Metal works	Paint shop
Turbo supplier	Drive train assembly
Engine	Drive train assembly
Gearbox	Drive train assembly

The supply chain model may now be extended to reveal the internal operations of each node: Upon receiving the sales order, the node representing ***assembly line*** creates a job-list and decomposes this into a material requirement plan for each of its suppliers:

Table 28 List of material requirements for the car manufacturer

Car	Component warehouse	Paint shop	Drive train assembly
a	Type A kit	Red body kit	Model 2.5 Turbo
b	Type A kit	Red body kit	Model 2.5 Turbo
c	Type X luxury kit	Metallic Red body kit	Model 3.0 Turbo
d	Type B kit	White body kit	Model 2.5 Sport
e	Type B kit	White body kit	Model 2.5 Sport
f	Type A kit	White body kit	Model 2.5 Sport
g	Type A kit	White body kit	Model 2.5 Sport

The information for this can be stored either within the node representing the assembly line or it may be available via an external service provided by engineering, sales, etc.

The assembly line now creates and sends an order to each of its suppliers, so that they only receive relevant information, which includes loss of information about which car each part is delivered for.

The component warehouse will therefor receive an order for:

SKU	Quantity
Type A kit	4
Type X luxury kit	1
Type B kit	2

The paint shop will receive an order for:

SKU	Quantity

Red body kit	2
Metallic Red body kit	1
White body kit	4

And the drive train assembly will receive an order for:

SKU	Quantity
Model 2.5 Turbo	2
Model 3.0 Turbo	1
Model 2.5 Sport	4

Following the path this will result in the additional supply request, for example from the drive train assembly, for the following SKUs.

Turbo supplier

SKU	Quantity
Turbo 3.0	1
Turbo 2.5	2

Engine

SKU	Quantity
2.5 L	6
3.0 L	1

Gearbox

SKU	Quantity
2.5 Turbo Automatic	2
2.5 Turbo Manual	2
3.0 Turbo Pedal-shift	1
2.5 Sport Pedal-shift	2

Likewise the paint shop will order the metal works to produce 7 body kits:

SKU	Quantity
Porsche Cayman Body Kit	7

Together with body kits, the paint shop will order the appropriate colour kit from the component warehouse.

SKU	Quantity
Type A kit	4
Type X luxury kit	1
Type B kit	2

Using the relationship from the previous example, one may argue that M1 or "left side" is "upstream" and M2 or "right side" is "downstream". Using these terms, the most "upstream" side of the supply chain may commence to create the schedule, and send supply schedule "downstream" (towards the right), so that the "downstream" side may commence the calculation of their production schedule. This may then propagate until it reaches the most "downstream" operation.

In coherence with the "Theory of Constraints" (Goldratt & Cox 2004) absence of conflicts in achieving the most downstream requests (i.e. the delivery date promised by the car salesmen) would render any need for optimisation void. On the other hand, any problems in fulfilling the most downstream demand would immediately result in requirements to reschedule as illustrated in the first example.

This can easily be extended with delivery dates, dependency within operations, etc., in the same manner as the idle-time was used in the first example:

Table 29 Joblist from job-shop example M1+M2 in state 9

Joblist	a	b	c	d	e	f	g
Supply time	2	6	11	19	28	38	50
Runtime	14	5	7	10	6	3	6
Idle time	2	0	0	0	0	0	3
Start (job)	2	16	21	28	38	44	50
Finish (job)	16	21	28	38	44	50	56

Table 30 Joblist from job-shop example M1+M2 state 9 extended to include flags for delivery time

Joblist	a	b	c	d	e	f	g
Supply time	2	6	11	19	28	38	50
Runtime	14	5	7	10	6	3	6
Idle time	2	0	0	0	0	0	3
Requested Delivery time	***40***	***40***	***40***	***50***	***50***	***60***	***50***
Start (job)	2	16	21	28	38	44	50
Finish (job)	16	21	28	38	44	50	56

The key point is that Boolean checks are easy to perform once the operational requirements are clear.

A technical dependency for scalable systems

Assuming that a monitoring service would log all the exchanged messages in the car manufacturing example, the list of generated messages would look as follows:

TimeStep	Active node	Operation	Receiver	Direction
1	Car Sales	Message exchange	Assembly Line	upstream
2	Assembly Line	Schedule	Self	n/a
3	Assembly Line	Message exchange	PaintShop	upstream

3	Assembly Line	Message exchange	Drive train assembly	upstream
3	Assembly Line	Message exchange	Component Warehouse	upstream
4	PaintShop	Schedule	Self	n/a
4	Drive train assembly	Schedule	Self	n/a
4	Component Warehouse	Schedule	Self	n/a
5	PaintShop	Message exchange	Metal works	upstream
5	PaintShop	Message exchange	Component Warehouse	upstream
5	Drive train assembly	Message exchange	Turbo Supplier	upstream
5	Drive train assembly	Message exchange	Engine	upstream
5	Drive train assembly	Message exchange	Gearbox	upstream
6	Metal works	Schedule	Self	n/a
6	Component Warehouse	Schedule	Self	n/a
6	Turbo Supplier	Schedule	Self	n/a
6	Engine	Schedule	Self	n/a
6	Gearbox	Schedule	Self	n/a
7	Metal works	Message exchange	Paint shop	downstream
7	Component Warehouse	Message exchange	Paint shop	downstream
7	Component Warehouse	Message exchange	Assembly Line	downstream
7	Turbo Supplier	Message exchange	Drive train assembly	downstream
7	Engine	Message exchange	Drive train assembly	downstream
7	Gearbox	Message exchange	Drive train assembly	downstream
8	Paintshop	Schedule	Self	n/a
8	Drive train assembly	Schedule	Self	n/a
8	Assembly	Schedule	Self	n/a
9	Paintshop	Message exchange	Assembly Line	downstream
9	Drive train assembly	Message exchange	Assembly Line	downstream
9	Assembly	Message exchange	Car Sales	downstream
10	Assembly Line	Schedule	Self	n/a
11	Assembly	Message exchange	Car Sales	downstream
12	Assembly Line	Schedule	Self	n/a

The attentive reader will immediately notice that the local operation involve in creating the schedule is identical disregarding which unit creates it.

In order for the New Supply Chain Model to meet the critique of being scalable – in terms of capable to handle complex networks of any topology and to be fast when the system has many components – it must be designed with physical system constraints in mind.

A.2 Test program (extension)

A.2.1 Overview of test program

Table with tests missing

A.2.2 Consistency of transactions and quality of schedule

This rest of this sections presents the summary the quality assurance process which the *New Supply Chain Model* was validated against. Each test is documented in the following framework:

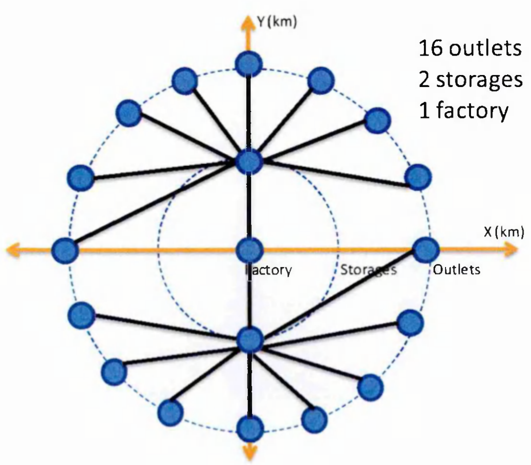
Case ID	Title of the test
Category	Position on the test-tree
Description	Description of the test
Network	Description of the network used in the test
Assertions	List of assertions the test makes
Results	Illustration of results visualizing the assertions
Results Data	Table of data generated by the test
Conclusion	Conclusion derived from the test

All synthetic cases are based either on a ***circular network*** of O outlets, connected by N channels to S storages through M channels to F factories. The distances used in the testing is based on Euclidian distance calculation, so that the length of a channel {A,B} is given by the coordinates A(x,y) and B(x,y), whereby the distance d is given by:

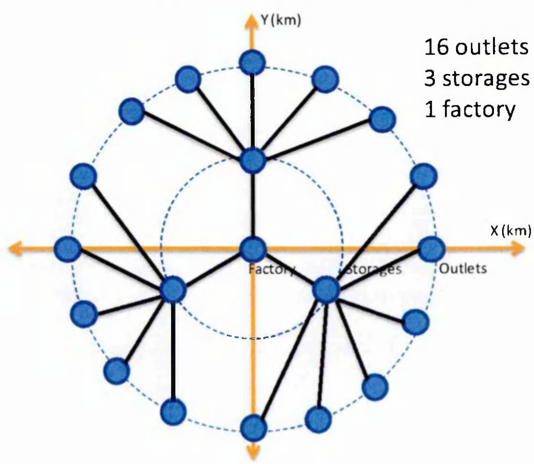
$$d = \sqrt[2]{(y_B - y_A)^2 + (x_B - x_A)^2}$$

In the generation of the synthetic network, the channel length is always minimized. In certain cases (such as 1-2-16 below) the distance from factory to outlet is shorter than |factory; storage|+ |storage; outlet|, but this route is not permitted.

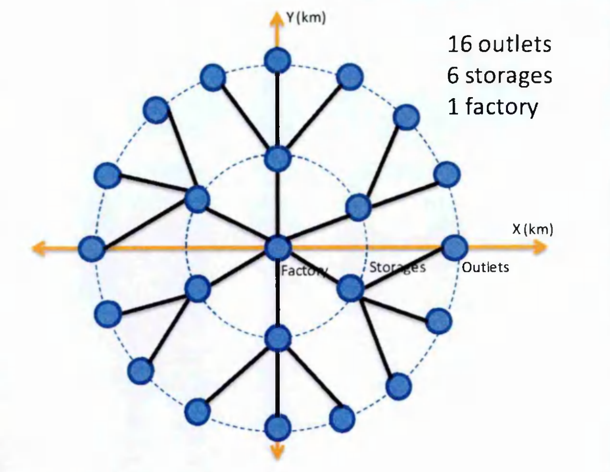
Examples are illustrated below.



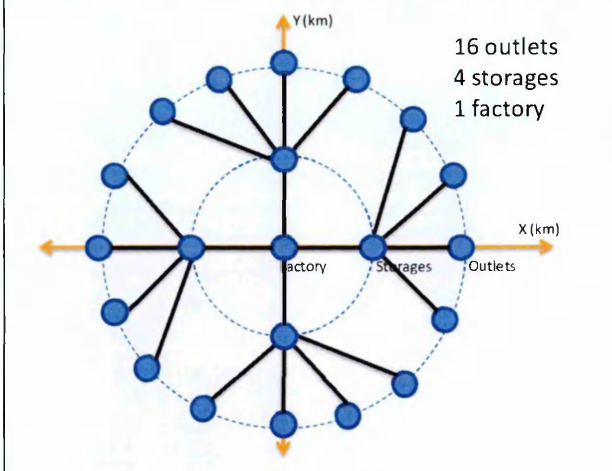
"Circular 1-2-16"



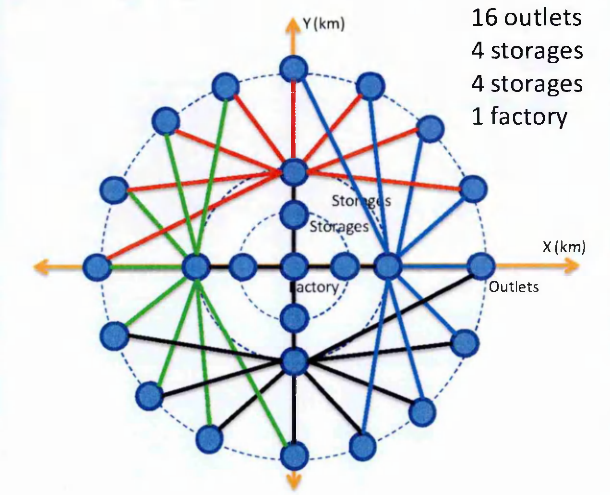
"Circular 1-3-16"



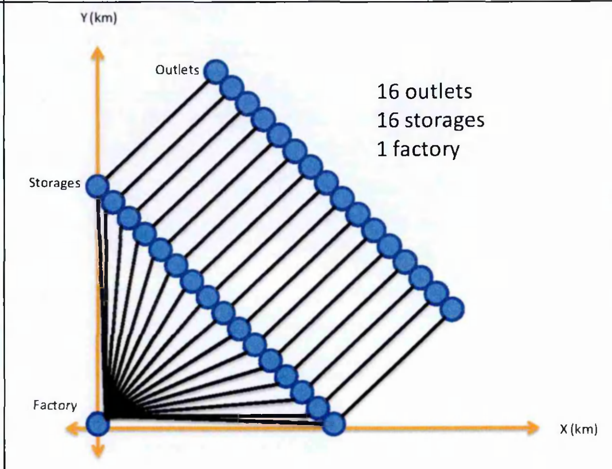
"Circular 1-6-16"



"Circular 1-4-16"



"Circular 1-4-4-16"



"Linear 1-16-16"

A.2.3 Increasing problem size

The following tests all refer to the group of tests on "increasing problem size". These are followed on page 183 by the tests referring to Increasing problem complexity.

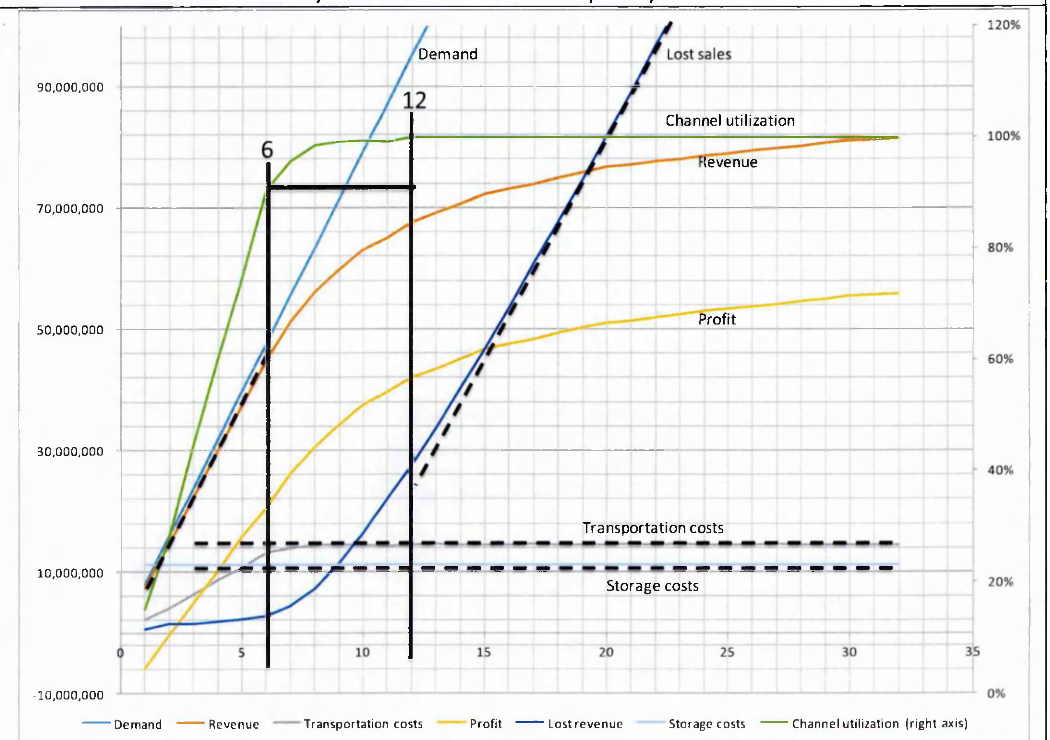
Consistency of transactions

The purpose of this category of tests is to assure that the schedulers constraints are respected correctly under different scenarios. The scenarios are:

1. Under increasing demand:
    a. That transport capacity limits are respected when all other capacities are infinite
    b. That storage capacity limits are respected when all other capacities are infinite
    c. That production capacity limits are respected when all other capacities are infinite
2. Under increasing costs
    a. That breakpoint is reached as <u>transport</u> costs increase linearly
    b. That breakpoint is reached as <u>storage</u> costs increase linearly

3. Under increasing number of storage sites that the profile below may be replicated.
4. Under different transport cost models
5. Under increasing volatility of demand
6. Under reducing lead-time that total costs drop (less cost of stock)
7. Under different network topologies

Increasing demand

Case ID	Increasing demand w. consistency across all KPIs (DA)
Category	Increasing problem size – increasing demand
Description	That transport capacity limits (channel) are respected when all other capacities are infinite under increasing demand.
Network	Circular 1-4-16
Assertions	Assert that: Revenue is correlated to demand. Storage costs are constant. Transport capacity limit is not exceeded. Transport utilization keeps increasing. Transport cost converge towards maximum. Maximum transport capacity is reached (the bottleneck). Profit increases after limit of network capacity has been reached. Lost sales increase linearly after the networks capacity limit is reached.
Results	

Results Data

Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	additional analysis
7,920,000	7,406,700	2,205,755	- 5,924,055	513,300	15%	11,125,000	
15,840,000	14,453,200	3,947,431	- 619,231	1,386,800	28%	11,125,000	
23,760,000	22,343,700	6,392,610	4,826,090	1,416,300	45%	11,125,000	
31,680,000	29,867,600	8,643,199	10,099,401	1,812,400	60%	11,125,000	<< SA4
39,600,000	37,376,500	10,626,811	15,624,689	2,223,500	74%	11,125,000	
47,520,000	44,838,000	13,114,802	20,598,198	2,682,000	91%	11,125,000	
55,440,000	51,132,900	13,893,675	26,114,225	4,307,100	96%	11,125,000	
63,360,000	56,171,200	14,349,873	30,696,327	7,188,800	98%	11,125,000	<< SA8
71,280,000	59,846,400	14,428,932	34,292,468	11,433,600	99%	11,125,000	
79,200,000	62,993,000	14,368,034	37,499,966	16,207,000	99%	11,125,000	
87,120,000	65,081,500	14,254,627	39,701,873	22,038,500	99%	11,125,000	
95,040,000	67,634,400	14,489,197	42,020,203	27,405,600	100%	11,125,000	
102,960,000	69,240,400	14,638,640	43,476,760	33,719,600	100%	11,125,000	
110,880,000	70,698,400	14,489,197	45,084,203	40,181,600	100%	11,125,000	
118,800,000	72,312,000	14,489,197	46,697,803	46,488,000	100%	11,125,000	
126,720,000	73,145,600	14,489,197	47,531,403	53,574,400	100%	11,125,000	
134,640,000	73,909,200	14,593,107	48,191,093	60,730,800	100%	11,125,000	
142,560,000	74,918,400	14,489,197	49,304,203	67,641,600	100%	11,125,000	
150,480,000	75,832,000	14,489,197	50,217,803	74,648,000	100%	11,125,000	
158,400,000	76,672,000	14,489,197	51,057,803	81,728,000	100%	11,125,000	<< SA20
166,320,000	77,116,200	14,638,640	51,352,560	89,203,800	100%	11,125,000	
174,240,000	77,550,400	14,489,197	51,936,203	96,689,600	100%	11,125,000	
182,160,000	78,029,300	14,529,197	52,375,103	104,130,700	100%	11,125,000	
190,080,000	78,489,600	14,489,197	52,875,403	111,590,400	100%	11,125,000	
198,000,000	78,952,000	14,489,197	53,337,803	119,048,000	100%	11,125,000	
205,920,000	79,350,400	14,489,197	53,736,203	126,569,600	100%	11,125,000	
213,840,000	79,776,300	14,658,640	53,992,660	134,063,700	100%	11,125,000	
221,760,000	80,208,800	14,489,197	54,594,603	141,551,200	100%	11,125,000	
229,680,000	80,608,000	14,489,197	54,993,803	149,072,000	100%	11,125,000	
237,600,000	81,120,000	14,489,197	55,505,803	156,480,000	100%	11,125,000	
245,520,000	81,264,000	14,489,197	55,649,803	164,256,000	100%	11,125,000	
253,440,000	81,408,000	14,489,197	55,793,803	172,032,000	100%	11,125,000	

Note: SA4, SA8 and SA20 refers to Storage Analysis 4, 8 & 20.

Conclusion

All assertions evaluate as true.

Observations:
The demand increases linearly throughout the modelling iterations {1-32}.

Most notable is the iteration 12 where the channel utilization reaches 100% and never exceeded. At this point (12) the transport cost reaches its maximum.
From this point (12): lost revenue grows linearly, only distorted visually by the variation in product prices where the lower priced products become lost revenue and the more profitable products are favoured. Finally lost revenue is perfectly correlated with demand from this point (Iteration>12, c= 0.9997).

Also notable is the change between iteration 6 and 12, where the channel utilization converges from linear growth (iteration <6) to maximum (> iteration 12).

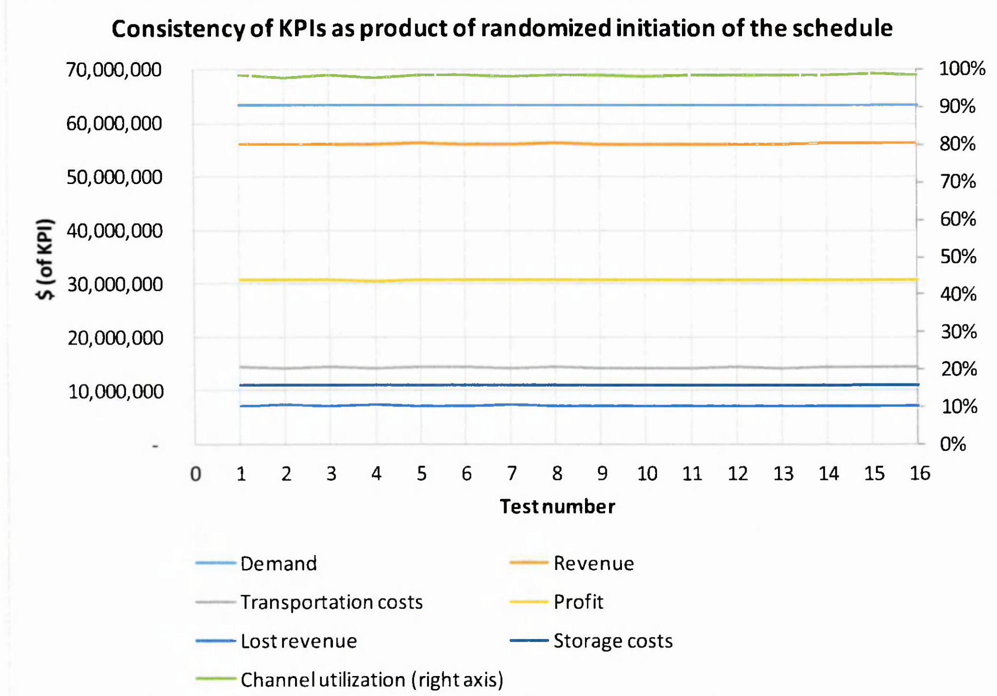
The revenue is perfectly correlated (c=0.9999) with demand until iteration 6, from where the growth of revenue slows down with convergence towards iteration 32 (c=0.9719 for i≥12).

As the storage costs are constant (€ 11.25m) the revenue (€ 7.4m) minus the transport (€ 2.2m) and storage costs reduce the losses from € 11.25m to € 5.924m, just as expected.

	The additional analysis SA4, SA8 and SA20 are selected as all points are in profitable iterations, but not in a transition zone.

Case ID	Resilience towards randomisation (VAR)
Category	Increasing problem size – increasing demand
Description	This test measures how randomized initiation of the scheduling affects the final KPIs.
Network	Circular 1-4-16
Assertions	Verify that the signal to noise ratio $\frac{\sigma}{\mu} \leq \frac{1}{100}$ over a set of iterations with randomized initiation.

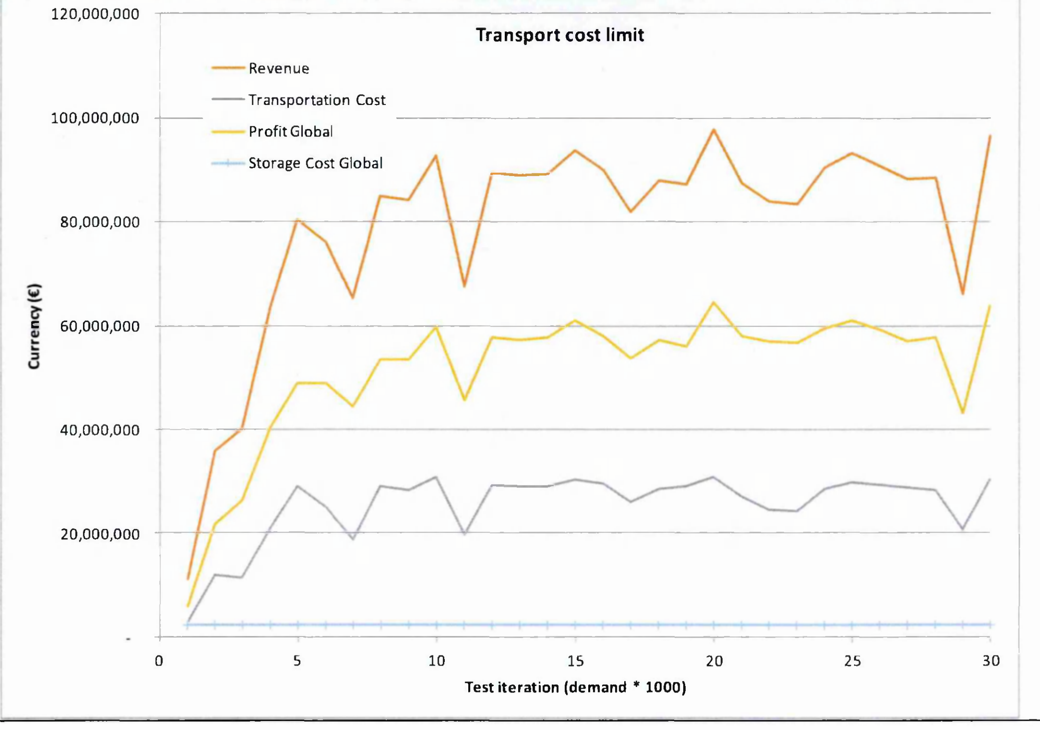
Results



Results Data

Iteration	Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	
1	63,360,000	56,190,800	14,366,871	30,698,929	7,169,200	98.5955%	11,125,000	
2	63,360,000	56,015,600	14,208,452	30,682,148	7,344,400	97.8933%	11,125,000	
3	63,360,000	56,208,400	14,355,406	30,727,994	7,151,600	98.5253%	11,125,000	
4	63,360,000	56,018,400	14,251,242	30,642,158	7,341,600	97.9635%	11,125,000	
5	63,360,000	56,244,800	14,372,404	30,747,396	7,115,200	98.6657%	11,125,000	
6	63,360,000	56,224,000	14,360,939	30,738,061	7,136,000	98.5955%	11,125,000	
7	63,360,000	56,034,000	14,217,985	30,691,015	7,326,000	98.0337%	11,125,000	
8	63,360,000	56,229,600	14,385,403	30,719,197	7,130,400	98.7360%	11,125,000	
9	63,360,000	56,186,000	14,294,372	30,766,628	7,174,000	98.4551%	11,125,000	
10	63,360,000	56,079,200	14,264,875	30,689,325	7,280,800	98.1742%	11,125,000	
11	63,360,000	56,161,600	14,295,805	30,740,795	7,198,400	98.4551%	11,125,000	
12	63,360,000	56,220,400	14,380,904	30,714,496	7,139,600	98.7360%	11,125,000	
13	63,360,000	56,196,400	14,294,372	30,777,028	7,163,600	98.4551%	11,125,000	
14	63,360,000	56,230,000	14,380,904	30,724,096	7,130,000	98.7360%	11,125,000	
15	63,360,000	56,261,200	14,394,936	30,741,264	7,098,800	98.8764%	11,125,000	
16	63,360,000	56,243,600	14,383,870	30,734,730	7,116,400	98.7360%	11,125,000	
stdev	-	85,008	63,920	34,077	85,008	0.3042%	-	**Average**
mean	63,360,000	56,171,500	14,325,546	30,720,954	7,188,500	98.4770%	11,125,000	**SNR**
SNR	0.00%	0.15%	0.45%	0.11%	1.18%	0.31%	0.00%	**0.31%**

Conclusion	The assertion evaluate as true. The signal to noise ratio across all KPI's evaluates to 0.31% which is less than the 1.0% threshold.

Case ID	Increasing demand under transport capacity limits (TCL)
Category	Increasing problem size – increasing demand
Description	This test verifies that transport capacity limits are respected in a network with very volatile product revenue and tight transport capacity limits.
Network	Quadratic 1-4-16
Assertions	Assert that transport capacity limits are respected under increasing demand.
Results	

Results Data

Iteration	Sales Demand	Revenue	Transportation Cost	Profit Global	Lost Sales	average Utilisation	Storage Cost Global
1	26,730,000	11,048,400	2,871,305	5,811,095	15,681,600	15%	2,366,000
2	53,460,000	35,847,600	11,850,616	21,630,984	17,612,400	52%	2,366,000
3	80,190,000	39,999,300	11,478,914	26,154,386	40,190,700	51%	2,366,000
4	106,920,000	63,629,400	21,012,864	40,250,536	43,290,600	78%	2,366,000
5	133,650,000	80,482,500	29,103,628	49,012,872	53,167,500	96%	2,366,000
6	160,380,000	76,224,000	24,915,445	48,942,555	84,156,000	87%	2,366,000
7	187,110,000	65,304,600	18,636,848	44,301,752	121,805,400	73%	2,366,000
8	213,840,000	84,838,200	28,967,254	53,504,946	129,001,800	95%	2,366,000
9	240,570,000	84,073,200	28,323,772	53,383,428	156,496,800	93%	2,366,000
10	267,300,000	92,838,000	30,774,930	59,697,070	174,462,000	100%	2,366,000
11	294,030,000	67,650,300	19,625,765	45,658,535	226,379,700	75%	2,366,000
12	320,760,000	89,364,600	29,372,379	57,626,221	231,395,400	96%	2,366,000
13	347,490,000	88,882,500	29,148,793	57,367,707	258,607,500	95%	2,366,000
14	374,220,000	89,175,600	29,053,674	57,755,926	285,044,400	95%	2,366,000
15	400,950,000	93,654,000	30,222,838	61,065,162	307,296,000	98%	2,366,000
16	427,680,000	89,887,200	29,605,683	57,915,517	337,792,800	96%	2,366,000
17	454,410,000	81,989,400	25,951,927	53,671,473	372,420,600	89%	2,366,000
18	481,140,000	88,081,200	28,439,761	57,275,439	393,058,800	94%	2,366,000
19	507,870,000	87,233,700	28,972,600	55,895,100	420,636,300	95%	2,366,000
20	534,600,000	97,794,000	30,773,386	64,654,614	436,806,000	100%	2,366,000
21	561,330,000	87,507,000	27,113,382	58,027,618	473,823,000	91%	2,366,000
22	588,060,000	83,883,600	24,476,281	57,041,319	504,176,400	87%	2,366,000
23	614,790,000	83,311,500	24,279,100	56,666,400	531,478,500	86%	2,366,000
24	641,520,000	90,513,600	28,649,518	59,498,082	551,006,400	94%	2,366,000
25	668,250,000	93,214,500	29,725,553	61,122,947	575,035,500	96%	2,366,000
26	694,980,000	90,732,600	29,219,691	59,146,909	604,247,400	95%	2,366,000
27	721,710,000	88,171,500	28,719,861	57,085,639	633,538,500	94%	2,366,000
28	748,440,000	88,445,400	28,315,022	57,764,378	659,994,600	93%	2,366,000
29	775,170,000	66,120,000	20,685,307	43,068,693	709,050,000	75%	2,366,000
30	801,900,000	96,591,000	30,388,550	63,836,450	705,309,000	98%	2,366,000

Conclusion	observation: At linearly growing demand and constant storage costs, the revenue increases as more expensive products are moved through the chain, whilst transport costs reach a maximum at iteration 5. From this point the utilization of the transport channels are up to, but not beyond 100%.

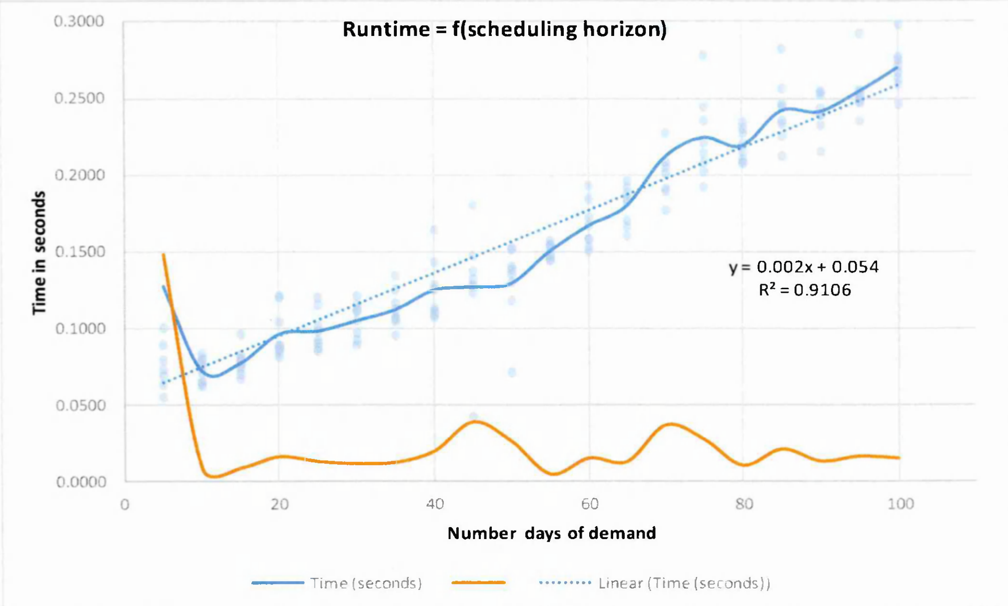
Case ID	Increasing demand under storage capacity limits (SCL --)
Category	Increasing problem size – increasing demand
Description	The test verifies that the storage capacity limits are respected under increasing demand. The test increases the storage capacity in linear steps under constant demand, which in steps is expected to be reflected as linear increase in sales (or inverse decrease of lost sales).
Network	Circular 1-4-16
Assertions	Assert that: Signal to noise ratio (SNR) < 1% That increase in storage capacity is perfectly correlated to sales (c=1.000) That increase in storage capacity is perfectly correlated to lost sales (c=-1.000)
Results	The SNR=0% whereby the assertion hold true
Results Data	N/A

Conclusion	The other tests reveal that this test is unnecessary but it is maintained for consistency of the test framework.

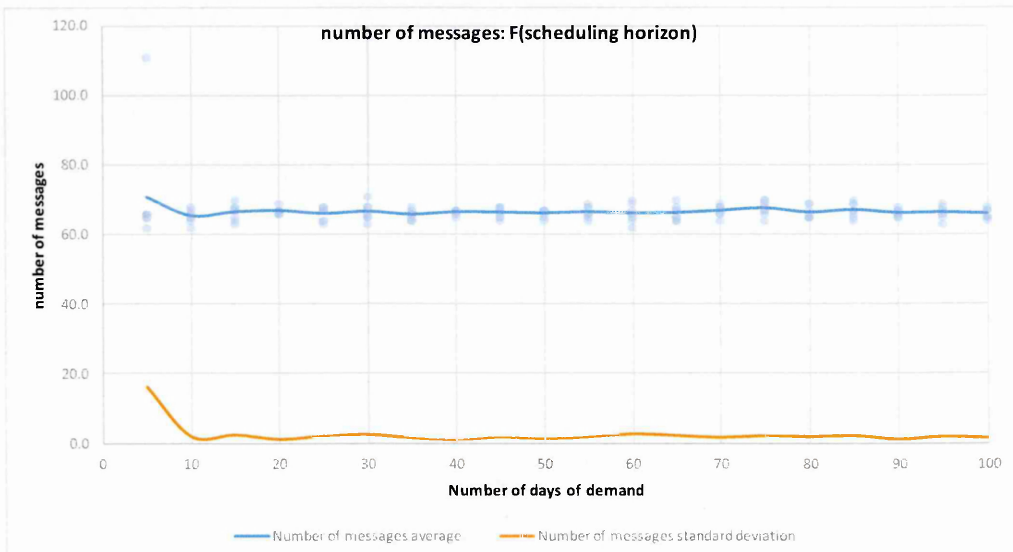
Case ID	Increasing demand under production capacity limits (PCL --)
Category	Increasing problem size – increasing demand
Description	We inspect that the KPI's of the network behave as expected given an increasing demand in a static network based on 16 shops, 4 storages and 1 factory
Network	Circular 1-4-16
Assertions	Assert that: Revenue is correlated to demand. Transport utilization keeps increasing. Maximum production capacity is reached (the bottleneck). Profit increases after limit of network capacity has been reached. Lost sales increase linearly after the networks capacity limit is reached.
Results	TBA
Results Data	TBA
Conclusion	TBA – other tests reveal that this test is unnecessary but let's add it for consistency

Case ID	Linear increase of scheduling horizon (TD,MS,OD)
Category	Increasing problem size – increasing demand
Description	This test verifies that: The numbers of solutions discovered grow linearly with the length of the solution landscape (time horizon). The trend and variation of number of messages exchanged when scheduling N days of demand ranging from 5 to 100 days. The trend and variation of time to schedule N days of demand ranging from 5 to 100 days. As 1 message can contain several SKUs (at any quantity for any time) the number of messages exchanged should be constant.
Network	Circular 1-4-16
Assertions	Assert that: The increment in time (seconds) is linear and predictable (R^2>0.9) as output from range in days of demand. The average number of solutions grows linearly (no significant trend line) The SNR < 10% The average number of messages is constant and that, The standard deviation < 1% or < 2 messages

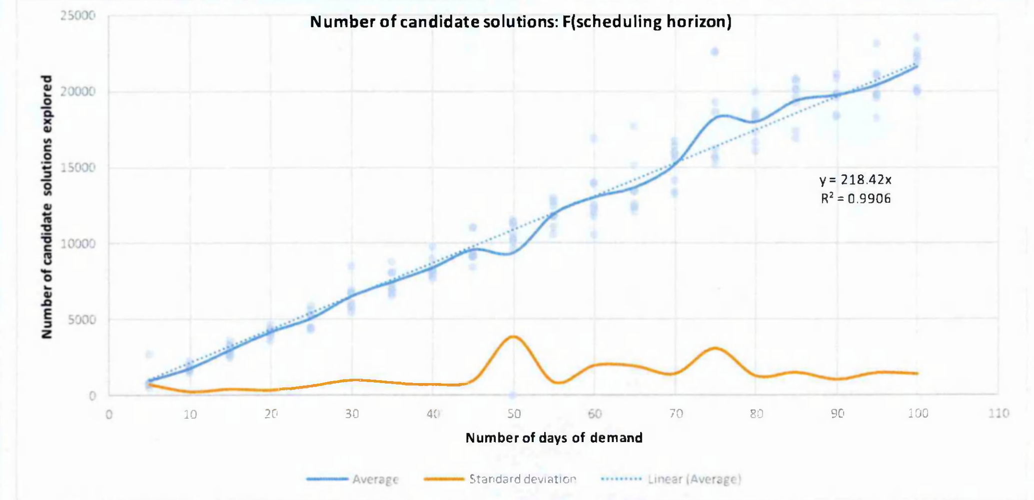
Results



The standard deviation of the runtime is relatively constant in comparison to the growing duration, which is linearly correlated to the length of the scheduling horizon. The shows that a network of this complexity does not need to have exponential runtime complexity.



The number of messages exchanged is constant.



The number of candidate solution grows linearly with the length of the scheduling horizon.

Results Data

Days of Demands	signal to noise ratio	Time (seconds) average	Time (seconds) standard deviation	Experiment No. 1	2	3	4	5	6	7	8
5	116.2%	0.1276	0.1483	0.4930	0.0890	0.0790	0.0630	0.0730	0.1000	0.0690	0.0550
10	11.2%	0.0724	0.0081	0.0830	0.0750	0.0620	0.0790	0.0640	0.0650	0.0800	0.0710
15	11.4%	0.0779	0.0089	0.0820	0.0780	0.0960	0.0700	0.0760	0.0740	0.0670	0.0800
20	16.8%	0.0966	0.0162	0.1210	0.0860	0.1200	0.0880	0.0890	0.0810	0.1040	0.0840
25	13.1%	0.0985	0.0129	0.0900	0.0920	0.1200	0.1010	0.0850	0.1150	0.0870	0.0980
30	10.9%	0.1053	0.0115	0.1210	0.1150	0.1050	0.0890	0.1110	0.0970	0.1120	0.0920
35	11.1%	0.1124	0.0124	0.1120	0.1060	0.0950	0.1340	0.1070	0.1160	0.1040	0.1250
40	15.9%	0.1250	0.0199	0.1430	0.1110	0.1130	0.1090	0.1280	0.1250	0.1640	0.1070
45	30.6%	0.1270	0.0389	0.1280	0.1270	0.1370	0.1470	0.1230	0.1800	0.0420	0.1320
50	20.1%	0.1294	0.0260	0.1400	0.1380	0.1350	0.1520	0.1510	0.1300	0.0710	0.1180
55	3.2%	0.1504	0.0048	0.1440	0.1570	0.1540	0.1450	0.1490	0.1540	0.1530	0.1470
60	9.2%	0.1669	0.0153	0.1580	0.1530	0.1670	0.1580	0.1930	0.1840	0.1500	0.1720
65	7.2%	0.1798	0.0130	0.1880	0.1820	0.1660	0.1840	0.1920	0.1960	0.1700	0.1600
70	17.4%	0.2116	0.0368	0.2950	0.2010	0.1910	0.1890	0.1770	0.2050	0.2270	0.2080
75	12.3%	0.2243	0.0275	0.2350	0.2780	0.2020	0.2140	0.1920	0.2440	0.2210	0.2080
80	4.8%	0.2189	0.0105	0.2340	0.2070	0.2200	0.2270	0.2100	0.2300	0.2150	0.2080
85	8.8%	0.2416	0.0212	0.2440	0.2250	0.2560	0.2120	0.2460	0.2330	0.2350	0.2820
90	5.5%	0.2410	0.0132	0.2320	0.2390	0.2540	0.2150	0.2390	0.2520	0.2540	0.2430
95	6.5%	0.2541	0.0165	0.2540	0.2470	0.2550	0.2500	0.2520	0.2480	0.2350	0.2920
100	5.6%	0.2695	0.0152	0.2750	0.2710	0.2770	0.2670	0.2460	0.2590	0.2630	0.2980

The heat map highlights the outlier of experiment no.1, from which the extreme value 116% arises. If this value is excluded the SNR drops to 20.4% for the 5 days of demands. Average SNR for non-outlier is 11.7%

Days of Demand	signal to noise ratio	Number of average	standard deviation	Experiment No. 1	2	3	4	5	6	7	8
5	23.0%	70.9	16.3	62.0	111.0	66.0	66.0	65.0	66.0	65.0	66.0
10	2.8%	65.4	1.8	66.0	64.0	62.0	65.0	68.0	66.0	65.0	67.0
15	3.5%	66.5	2.3	65.0	63.0	68.0	64.0	68.0	67.0	67.0	70.0
20	1.5%	66.9	1.0	66.0	66.0	67.0	66.0	67.0	67.0	69.0	67.0
25	3.0%	66.0	2.0	63.0	68.0	64.0	67.0	68.0	67.0	67.0	64.0
30	3.7%	66.6	2.4	71.0	63.0	65.0	68.0	67.0	65.0	68.0	66.0
35	2.1%	65.8	1.4	65.0	64.0	66.0	67.0	66.0	68.0	64.0	66.0
40	1.1%	66.5	0.8	67.0	67.0	67.0	67.0	66.0	65.0	66.0	67.0
45	2.4%	66.4	1.6	65.0	68.0	68.0	66.0	68.0	64.0	65.0	67.0
50	1.7%	66.1	1.1	66.0	67.0	67.0	66.0	64.0	67.0	65.0	67.0
55	2.5%	66.5	1.7	68.0	66.0	65.0	66.0	66.0	64.0	69.0	68.0
60	3.9%	66.1	2.6	67.0	64.0	66.0	65.0	69.0	70.0	62.0	66.0
65	3.2%	66.3	2.1	68.0	67.0	64.0	64.0	67.0	70.0	65.0	65.0
70	2.3%	66.9	1.6	67.0	69.0	66.0	64.0	67.0	68.0	66.0	68.0
75	3.1%	67.6	2.1	69.0	70.0	64.0	66.0	67.0	70.0	67.0	68.0
80	2.7%	66.4	1.8	67.0	65.0	65.0	69.0	65.0	69.0	66.0	65.0
85	3.1%	67.1	2.1	69.0	67.0	70.0	65.0	66.0	64.0	67.0	69.0
90	1.6%	66.3	1.0	65.0	67.0	66.0	68.0	66.0	66.0	67.0	65.0
95	2.8%	66.4	1.8	68.0	66.0	69.0	66.0	67.0	65.0	63.0	67.0
100	2.1%	66.0	1.4	67.0	65.0	67.0	65.0	65.0	67.0	64.0	68.0

The heat map illustrates the outlier of experiment 2 @ 5 days of demand. If this is excluded the SNR drops to 2.2% for the series.

Heat maps are used to highlight outliers in the dataset (see below, value: 0)

Days of Demands	signal to noise ratio	Number of candidate solutions: Average	Standard deviation	Experiment No. 1	2	3	4	5	6	7	8
5	71.1%	978	696	530	2686	743	748	826	749	804	740
10	12.1%	1,776	216	1738	1702	1504	1682	2242	1765	1690	1882
15	13.1%	2,989	391	2629	2472	3250	2703	3318	2984	2923	3632
20	7.6%	4,163	317	4130	3582	4632	3925	4223	4314	4399	4097
25	11.6%	5,053	585	4328	5610	4434	5250	5909	5284	5141	4465
30	14.8%	6,511	961	8513	5487	5801	6714	6930	6076	6725	5841
35	10.5%	7,453	785	7102	6735	6946	8034	8100	8804	6542	7364
40	8.2%	8,382	690	8995	8096	8302	9783	8301	8069	7690	7818
45	9.9%	9,574	949	9170	11020	11068	9310	9320	8439	9138	9127
50	41.0%	9,361	3837	10619	11366	11549	11200	9674	10323	0	10156
55	6.9%	11,915	817	12762	11082	11826	11846	11751	10591	12471	12993
60	14.9%	13,014	1935	14011	11801	12309	12513	13934	16909	10538	12095
65	13.9%	13,649	1893	15135	12382	13392	12107	13468	17675	12413	12623
70	9.1%	15,162	1382	16713	15883	14141	13204	13380	16392	15594	15991
75	16.8%	18,225	3069	19277	22618	15169	15731	16224	22570	15594	18620
80	6.9%	17,964	1234	18361	18699	16065	18208	17327	19939	18468	16644
85	7.6%	19,366	1471	20103	19268	20680	16887	20147	17386	20821	19632
90	5.1%	19,722	1002	19788	21159	19822	20846	18310	19608	19813	18430
95	7.1%	20,386	1445	23126	21140	20422	19808	19770	19558	18216	21046
100	6.3%	21,577	1361	22606	20175	23503	21849	19986	22322	19948	22224

Once more experiment 2 has an extreme value in SNR, which, if excluded results in a signal to noise ratio of 13.1% In addition in experiment 7 @ 50 days of demand, an extreme value is present (recording error) which, if excluded, brings the SNR down to 6.5% for the series.

Conclusion

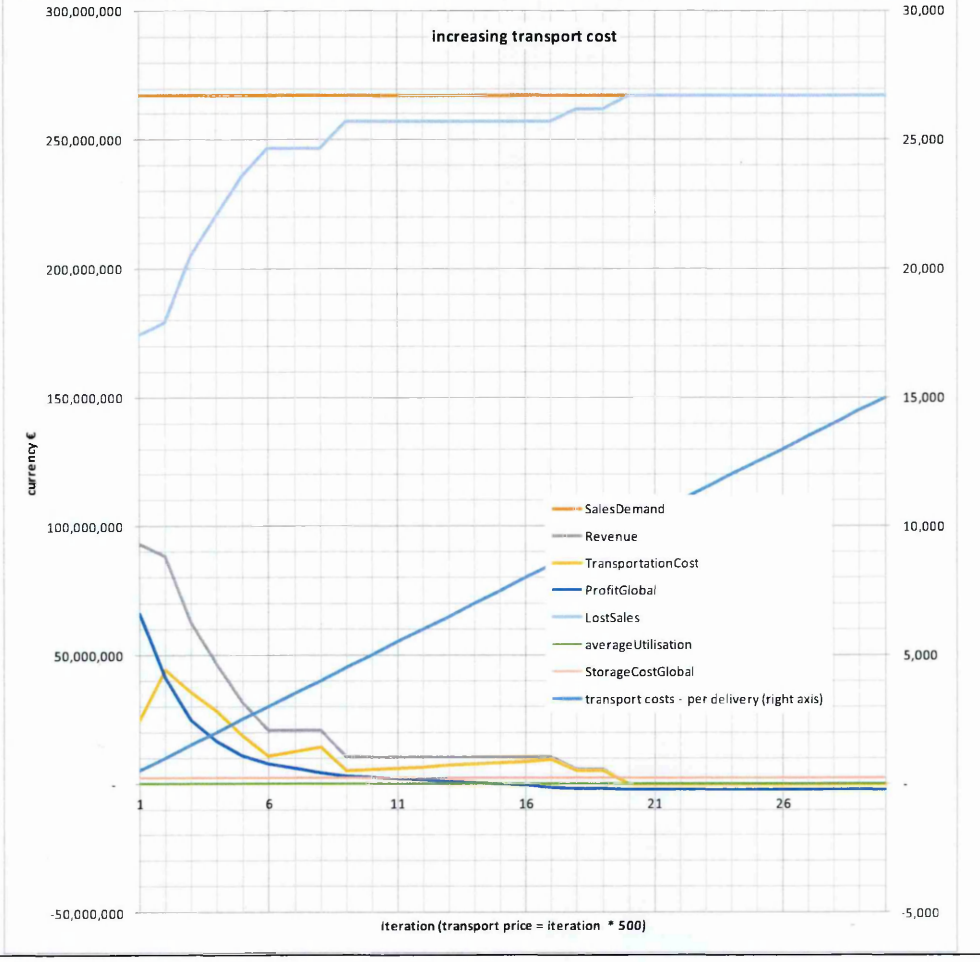
The assertions hold true, with exception of runtime and candidate solution prediction. This may be due to the small sample size. Whereby if the sample is larger, the SNR will drop further.

The increment in time (seconds) is linear and predictable (R^2=0.9106 > 0.9) as output from range in days of demand.

The average number of solutions grows linearly at α=218x, R^2=0.9906.

The signal to noise ratio is < 11.6% on prediction of runtime, whilst 2.6% on number of messages and 11.8% on messages.

Increasing costs

Case ID	Increasing transport costs (ITC)
Category	Increasing problem size – increasing demand
Description	The test provides evidence of how the scheduler behaves if the cost increase
Network	Circular 1-4-16
Assertions	Asserts that: Transportation cost and revenue are inversely correlated. Lost sales + revenue = demand (a constant) Global profit drops stepwise as costs increase (until zero revenue).
Results	

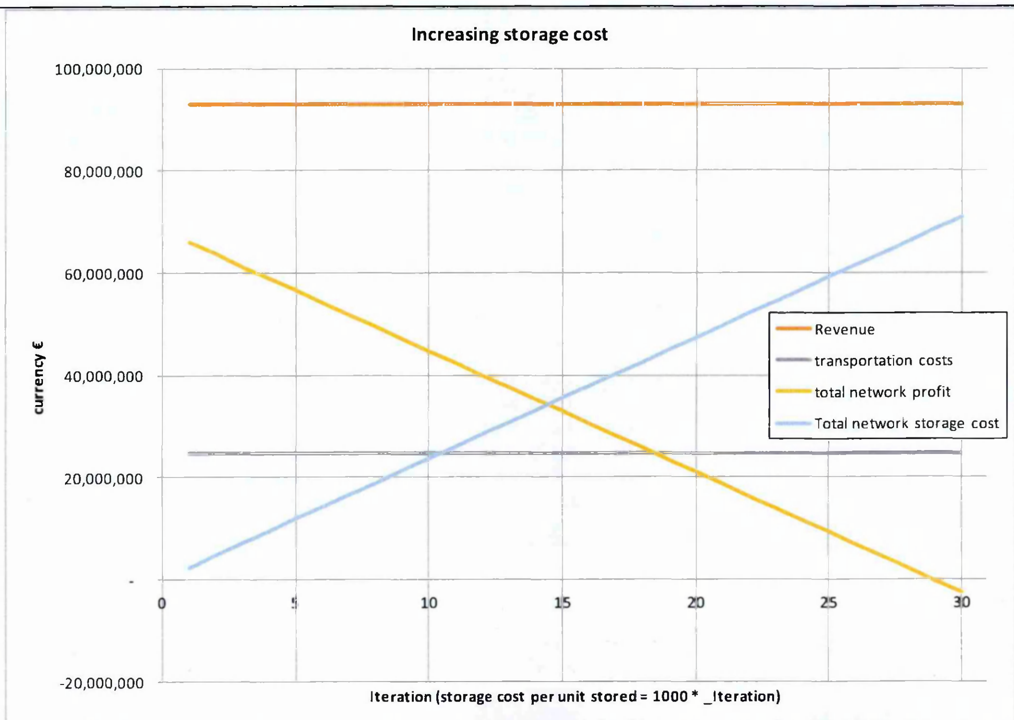
Results Data

Iteration	transport costs - per delivery (right axis)	SalesDemand	Revenue	Transportation Cost	ProfitGlobal	LostSales	average utilisation	Storage Cost Global
1	500	267,300,000	93,204,000	24,757,816	66,080,184	174,096,000	100%	2,366,000
2	1,000	267,300,000	88,248,000	44,428,560	41,453,440	179,052,000	94%	2,366,000
3	1,500	267,300,000	62,550,000	35,721,817	24,462,183	204,750,000	67%	2,366,000
4	2,000	267,300,000	46,698,000	28,142,800	16,189,200	220,602,000	50%	2,366,000
5	2,500	267,300,000	31,512,000	18,634,850	10,511,150	235,788,000	33%	2,366,000
6	3,000	267,300,000	20,712,000	10,631,136	7,714,864	246,588,000	22%	2,366,000
7	3,500	267,300,000	20,712,000	12,402,992	5,943,008	246,588,000	22%	2,366,000
8	4,000	267,300,000	20,616,000	14,047,520	4,202,480	246,684,000	22%	2,366,000
9	4,500	267,300,000	10,356,000	4,959,000	3,031,000	256,944,000	11%	2,366,000
10	5,000	267,300,000	10,356,000	5,510,000	2,480,000	256,944,000	11%	2,366,000
11	5,500	267,300,000	10,356,000	6,061,000	1,929,000	256,944,000	11%	2,366,000
12	6,000	267,300,000	10,260,000	6,510,000	1,384,000	257,040,000	11%	2,366,000
13	6,500	267,300,000	10,260,000	7,052,500	841,500	257,040,000	11%	2,366,000
14	7,000	267,300,000	10,260,000	7,595,000	299,000	257,040,000	11%	2,366,000
15	7,500	267,300,000	10,260,000	8,137,500	- 243,500	257,040,000	11%	2,366,000
16	8,000	267,300,000	10,260,000	8,680,000	- 786,000	257,040,000	11%	2,366,000
17	8,500	267,300,000	10,260,000	9,222,500	- 1,328,500	257,040,000	11%	2,366,000
18	9,000	267,300,000	5,400,000	4,905,000	- 1,871,000	261,900,000	5%	2,366,000
19	9,500	267,300,000	5,400,000	5,177,500	- 2,143,500	261,900,000	5%	2,366,000
20	10,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
21	10,500	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
22	11,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
23	11,500	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
24	12,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
25	12,500	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
26	13,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
27	13,500	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
28	14,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
29	14,500	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000
30	15,000	267,300,000	-	-	- 2,366,000	267,300,000	0%	2,366,000

Conclusion	The assertion evaluates true. The correlation between profit and total storage costs is perfect (c=-1.000)

Case ID	Increasing storage costs (ISC)
Category	Increasing problem size – increasing demand
Description	Under increasing costs of storage, it is expected that a breakpoint in profitability is reached as costs increase linearly. At the breakpoint, the supply of stock should be brought to a halt.
Network	Circular 1-4-16
Assertions	Assert that there is perfect inverse correlation (c=-1.000) between total network profit and total network storage costs as cost increase per stored unit.

Results



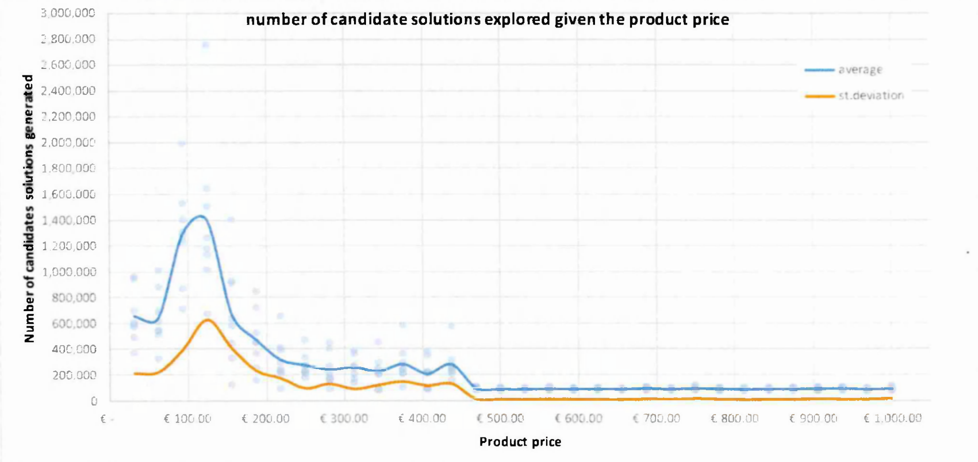
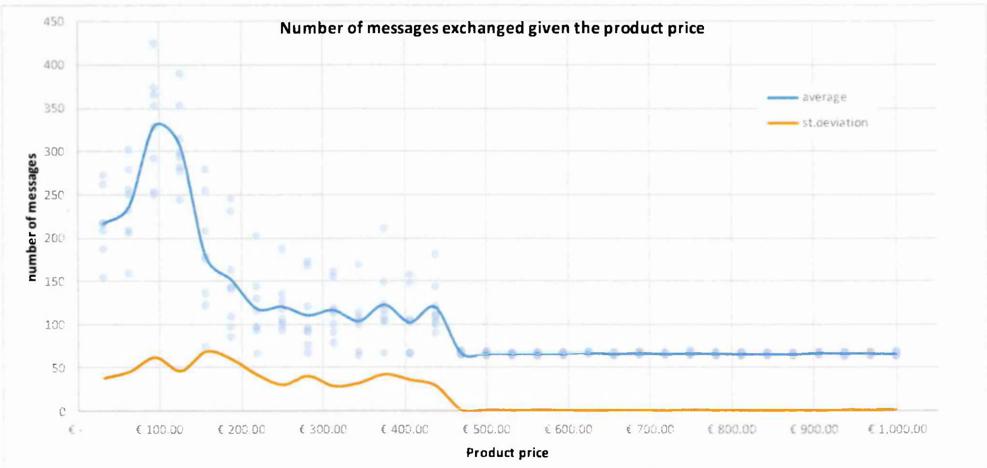
Results Data

Iteration	Demand	Revenue	transportation costs	total network profit	lost sales	average utilisation	Total network storage cost
1	267,300,000	93,204,000	24,757,816	66,080,184	174,096,000	100%	2,366,000
2	267,300,000	93,204,000	24,757,816	63,714,184	174,096,000	100%	4,732,000
3	267,300,000	93,189,000	24,757,816	61,333,184	174,111,000	100%	7,098,000
4	267,300,000	93,189,000	24,757,816	58,967,184	174,111,000	100%	9,464,000
5	267,300,000	93,189,000	24,757,816	56,601,184	174,111,000	100%	11,830,000
6	267,300,000	93,189,000	24,757,816	54,235,184	174,111,000	100%	14,196,000
7	267,300,000	93,189,000	24,757,816	51,869,184	174,111,000	100%	16,562,000
8	267,300,000	93,189,000	24,757,816	49,503,184	174,111,000	100%	18,928,000
9	267,300,000	93,189,000	24,757,816	47,137,184	174,111,000	100%	21,294,000
10	267,300,000	93,189,000	24,757,816	44,771,184	174,111,000	100%	23,660,000
11	267,300,000	93,192,000	24,757,816	42,408,184	174,108,000	100%	26,026,000
12	267,300,000	93,189,000	24,757,816	40,039,184	174,111,000	100%	28,392,000
13	267,300,000	93,192,000	24,757,816	37,676,184	174,108,000	100%	30,758,000
14	267,300,000	93,189,000	24,757,816	35,307,184	174,111,000	100%	33,124,000
15	267,300,000	93,192,000	24,757,816	32,944,184	174,108,000	100%	35,490,000
16	267,300,000	93,189,000	24,757,816	30,575,184	174,111,000	100%	37,856,000
17	267,300,000	93,189,000	24,757,816	28,209,184	174,111,000	100%	40,222,000
18	267,300,000	93,189,000	24,757,816	25,843,184	174,111,000	100%	42,588,000
19	267,300,000	93,189,000	24,757,816	23,477,184	174,111,000	100%	44,954,000
20	267,300,000	93,189,000	24,757,816	21,111,184	174,111,000	100%	47,320,000
21	267,300,000	93,189,000	24,757,816	18,745,184	174,111,000	100%	49,686,000
22	267,300,000	93,189,000	24,757,816	16,379,184	174,111,000	100%	52,052,000
23	267,300,000	93,189,000	24,757,816	14,013,184	174,111,000	100%	54,418,000
24	267,300,000	93,189,000	24,757,816	11,647,184	174,111,000	100%	56,784,000
25	267,300,000	93,189,000	24,757,816	9,281,184	174,111,000	100%	59,150,000
26	267,300,000	93,189,000	24,757,816	6,915,184	174,111,000	100%	61,516,000
27	267,300,000	93,189,000	24,757,816	4,549,184	174,111,000	100%	63,882,000
28	267,300,000	93,189,000	24,757,816	2,183,184	174,111,000	100%	66,248,000
29	267,300,000	93,189,000	24,757,816	- 182,816	174,111,000	100%	68,614,000
30	267,300,000	93,189,000	24,757,816	- 2,548,816	174,111,000	100%	70,980,000

Conclusion

The assertion holds true. The correlation between profit and storage costs is perfect (c = -1.0000).

Increasing price of products

Case ID	Increasing profitability (Pprof)
Category	Increasing problem size – increasing demand
Description	The test evaluates the number of solutions explored in a given network and the number of messages exchange given that 16 products, all of same price are reduced over 30 iterations from 1000$ to 31.25 $ (i/30*1000$)
Network	Circular 1-4-16
Assertions	Assert that the number of candidate solutions is constant (minimal) once the price increases to a generally profitable level.
Results	 
Results & Data	

product price	number of options average	number of options st.deviation	Experiment No. 1	2	3	4	5	6	7	8
€ 1,000.00	95,987	13,868	84,610	89,910	85,319	91,210	90,717	109,767	91,933	124,432
€ 968.75	90,903	10,028	73,642	91,806	89,555	85,631	89,854	104,284	104,554	87,899
€ 937.50	98,255	10,030	98,595	85,874	86,973	103,559	102,999	89,481	114,417	104,142
€ 906.25	96,026	11,558	85,822	103,024	88,116	113,797	79,139	96,939	94,599	106,775
€ 875.00	93,802	9,515	106,604	91,244	88,964	84,356	86,998	94,929	109,971	87,346
€ 843.75	93,220	8,296	92,378	84,071	110,234	87,178	88,175	97,794	96,492	89,441
€ 812.50	90,617	7,699	83,252	88,026	86,805	89,281	107,860	94,958	87,901	86,854
€ 781.25	94,520	9,638	79,456	104,832	89,753	86,834	102,196	89,374	97,264	106,448
€ 750.00	98,311	13,377	84,531	88,484	88,025	92,570	97,391	101,667	109,053	124,768
€ 718.75	91,348	7,963	83,700	84,788	85,399	88,348	91,075	91,632	106,788	99,055
€ 687.50	98,363	11,725	98,429	91,624	88,588	78,038	105,340	110,567	112,445	101,872
€ 656.25	91,526	5,987	91,626	86,166	85,391	84,699	90,339	100,309	98,044	95,631
€ 625.00	94,193	8,600	87,578	88,814	85,507	84,539	100,378	97,227	106,486	103,014
€ 593.75	91,375	7,387	85,172	87,562	86,146	81,114	93,935	98,121	102,027	96,925
€ 562.50	95,013	10,278	87,061	114,548	89,922	85,305	91,053	87,898	103,894	100,426
€ 531.25	90,716	7,006	88,982	79,845	90,915	88,273	104,396	88,797	89,035	95,483
€ 500.00	92,572	8,964	83,867	83,109	92,021	82,950	99,573	93,003	98,758	107,297
€ 468.75	99,383	9,519	103,826	87,461	102,505	85,930	104,656	102,973	93,829	113,881
€ 437.50	284,041	127,326	223,655	256,464	319,674	242,944	578,378	214,258	273,326	163,626
€ 406.25	211,687	114,103	231,440	386,812	352,369	87,483	100,439	100,304	219,994	214,658
€ 375.00	285,388	144,262	213,061	369,105	243,970	215,702	106,476	590,515	245,625	298,647
€ 343.75	228,938	117,656	216,729	249,948	231,568	84,138	455,303	296,999	91,433	205,382
€ 312.50	260,842	86,265	212,550	390,300	173,952	262,637	262,214	143,029	274,541	367,509
€ 281.25	243,982	126,815	170,746	271,814	407,778	91,547	157,211	183,530	214,702	454,528
€ 250.00	278,385	92,860	476,216	272,932	275,752	180,139	338,864	234,695	217,010	231,468
€ 218.75	320,604	171,048	299,618	244,225	659,763	397,675	230,751	97,471	218,469	416,856
€ 187.50	475,555	228,574	529,547	368,868	458,697	452,876	850,651	726,132	258,136	159,531
€ 156.25	672,521	402,242	585,488	126,776	638,534	930,533	911,713	1,405,178	332,174	449,769
€ 125.00	1,399,218	624,318	2,759,830	1,647,787	1,509,323	1,181,216	1,267,131	1,018,096	1,134,153	676,205
€ 93.75	1,293,466	392,558	1,532,517	1,403,741	715,388	872,502	1,311,026	1,235,358	1,996,441	1,280,758
€ 62.50	643,216	218,326	694,753	1,014,029	883,811	547,272	515,287	549,054	326,364	615,158
€ 31.25	657,865	208,123	701,261	949,252	495,326	374,000	967,371	592,040	574,934	608,734

product price	number of messages average	number of messages st.deviation	Experiment No. 1	2	3	4	5	6	7	8
€ 1,000.00	66	2	64	65	67	64	64	68	64	69
€ 968.75	67	2	64	68	66	64	66	68	68	68
€ 937.50	66	2	66	65	65	68	68	63	68	68
€ 906.25	67	1	65	67	68	67	65	68	68	67
€ 875.00	66	2	68	66	64	64	66	65	67	64
€ 843.75	66	1	64	65	68	66	66	67	66	65
€ 812.50	66	1	65	67	66	68	67	66	64	64
€ 781.25	66	1	65	66	67	65	68	64	67	67
€ 750.00	67	2	66	65	65	66	67	65	68	70
€ 718.75	66	1	65	66	66	66	65	66	68	65
€ 687.50	67	1	67	66	65	65	68	67	69	67
€ 656.25	66	1	68	66	65	65	66	67	66	64
€ 625.00	67	1	67	68	66	67	66	68	68	67
€ 593.75	66	1	68	65	64	66	67	67	67	67
€ 562.50	66	2	65	67	68	67	67	63	66	67
€ 531.25	66	1	65	64	67	67	67	67	66	66
€ 500.00	66	2	65	64	66	64	69	66	68	67
€ 468.75	67	2	65	67	68	65	68	66	70	67
€ 437.50	120	29	104	121	144	111	181	100	110	90
€ 406.25	102	36	104	158	148	66	67	67	106	103
€ 375.00	123	42	103	149	117	104	67	211	110	123
€ 343.75	104	32	100	108	104	64	169	114	68	103
€ 312.50	117	29	100	155	91	118	113	79	116	161
€ 281.25	111	40	91	121	168	67	76	93	96	173
€ 250.00	121	30	187	118	126	93	135	102	106	99
€ 218.75	118	42	116	97	202	130	97	66	93	144
€ 187.50	152	59	163	143	109	141	231	246	97	85
€ 156.25	179	68	178	74	208	255	176	279	122	136
€ 125.00	307	46	390	277	353	294	299	282	315	244
€ 93.75	330	62	425	366	250	253	374	353	329	292
€ 62.50	237	45	256	302	279	232	206	209	159	250
€ 31.25	217	38	217	273	187	154	262	216	208	217

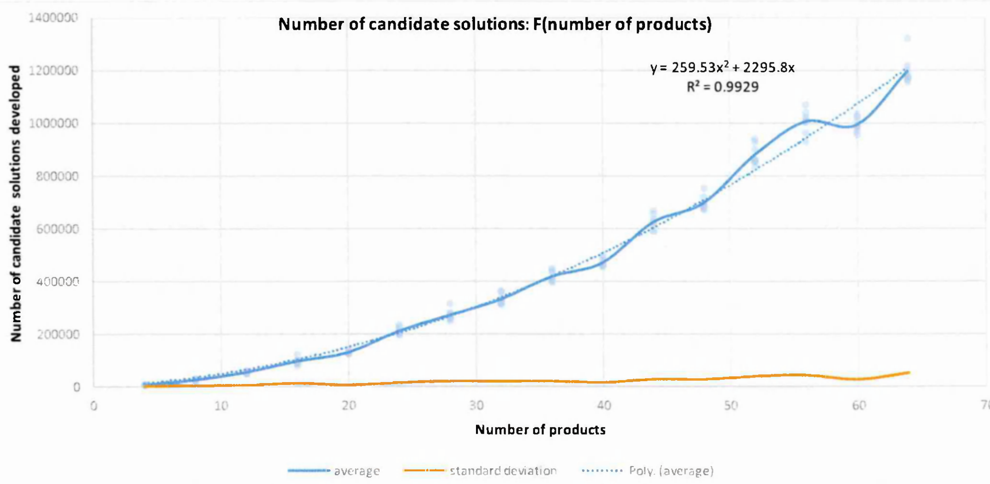
Conclusion	The assertion holds true. A clear phase shift occurs at the transition-point where the product becomes profitable.

Increasing number SKUs

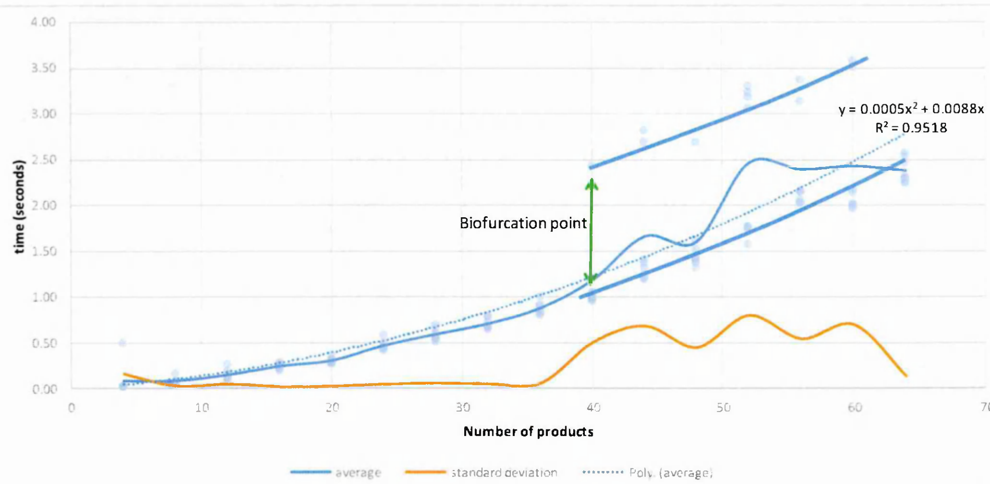
Case ID	Same price (PP)
Category	Increasing problem size – increasing demand

Description	In a 1-4-16 network where the price is constant across all products and profitable, the channel capacity sufficient and no need for network capacity negotiations (unconstrained network capacity), changing the number of products should according to theory on computational complexity exhibit strong exponential growth.
Network	1-4-16
Assertions	Assert that an increase in number of SKUs has computational complexity <$O(n^m)$

Results



The numbers of candidate solutions are identified at quadratic growth.



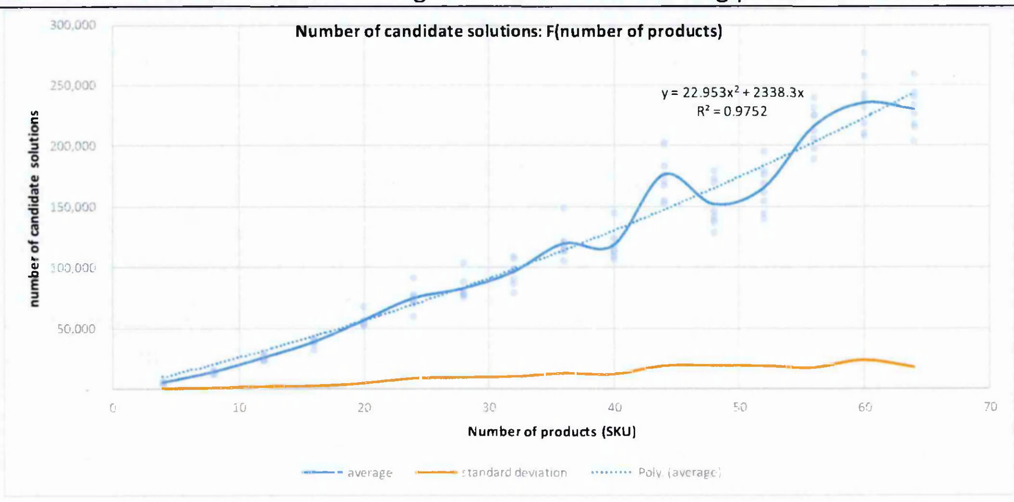
The scheduling problem (at this network complexity) is solved at quadratic runtime with very low growth factor (α=0.0005), indicating the benefit of solving the problem in a highly parallelized manner.

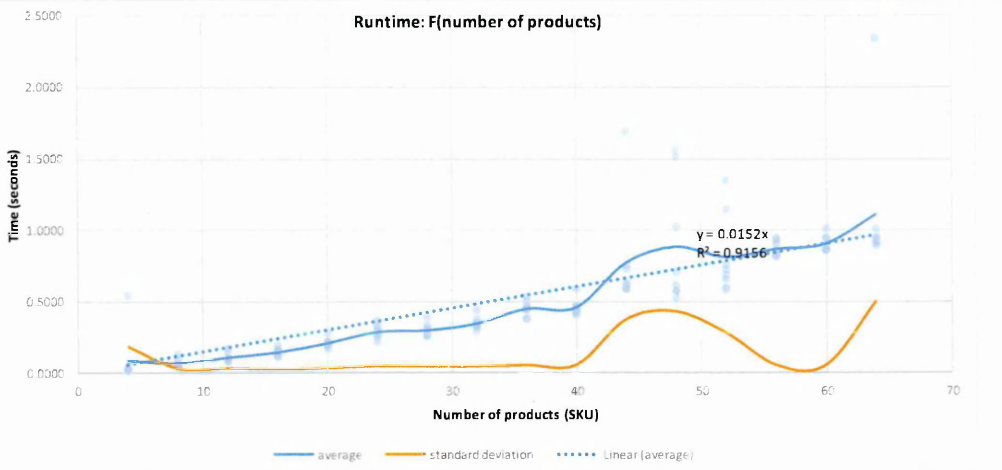
Results Data

Number of products	Number of candidate solutions		Experiment No.							
	average	standard deviation	1	2	3	4	5	6	7	8
4	8,101	1267.67	6912	7372	9274	7623	10435	6651	8485	8058
8	27,027	2767.25	26723	25097	31674	25464	27139	30471	23380	26269
12	55,675	4217.72	63416	59132	53116	54431	51526	57880	50998	54898
16	97,430	12720.65	87983	94317	100089	90767	100278	123315	80619	102072
20	129,963	4974.95	130616	127028	125624	130801	130098	141211	126052	128275
24	211,769	14586.46	218755	198285	208899	236308	197636	227996	207396	198879
28	272,392	20212.47	273758	251270	261526	316611	265873	280017	273093	256988
32	332,629	19693.92	364121	329861	319402	313835	321701	334147	316793	361175
36	419,418	20427.72	441732	409720	409256	451018	406867	436891	401481	398378
40	471,745	15401.57	461399	460605	489091	463606	500463	474569	457424	466805
44	626,518	28284.30	657244	591754	593104	622184	611680	638536	670646	626998
48	698,649	27108.14	694107	681841	754863	721880	686593	679192	696845	673871
52	880,829	39636.14	861737	902536	863417	942236	834280	854166	931917	856339
56	1,007,638	43886.33	1006162	1025174	1004499	1016538	964800	1071801	930384	1041743
60	995,456	27331.57	990948	969753	1026310	1013125	980542	990833	1035051	957084
64	1,198,586	53661.00	1321980	1156679	1170799	1174262	1179089	1219250	1166930	1199698

Number of products	time (seconds)		Experiment No.							
	average	standard deviation	1	2	3	4	5	6	7	8
4	0.0867	0.1668	0.4996	0.0248	0.0274	0.0277	0.0300	0.0264	0.0324	0.0256
8	0.0869	0.0347	0.1692	0.0747	0.0794	0.0796	0.0660	0.0862	0.0564	0.0836
12	0.1525	0.0563	0.2725	0.1944	0.1125	0.1162	0.1174	0.1607	0.1299	0.1164
16	0.2471	0.0286	0.2762	0.2500	0.2452	0.2062	0.2303	0.2901	0.2609	0.2176
20	0.3053	0.0314	0.3236	0.3173	0.2830	0.2850	0.3036	0.3692	0.2714	0.2892
24	0.4694	0.0526	0.4566	0.4473	0.4665	0.5007	0.4371	0.5851	0.4178	0.4443
28	0.5915	0.0636	0.7002	0.5128	0.5632	0.6076	0.5539	0.6612	0.5959	0.5372
32	0.7077	0.0553	0.7854	0.6684	0.6686	0.6488	0.6817	0.7936	0.6880	0.7269
36	0.8787	0.0649	1.0086	0.8645	0.8652	0.9206	0.8389	0.9015	0.8258	0.8042
40	1.1831	0.5086	1.0536	0.9603	2.4390	0.9710	1.0485	1.0100	0.9992	0.9830
44	1.6607	0.6837	1.3780	1.1969	2.6963	1.2107	1.4117	2.8240	1.3233	1.2444
48	1.6004	0.4505	1.4004	1.3832	1.4332	1.6254	1.5249	2.6914	1.4223	1.3225
52	2.4593	0.7989	3.0624	3.2393	1.7682	1.7747	1.5810	3.1917	1.7439	3.3135
56	2.3915	0.5437	3.3811	2.0367	2.0290	2.0666	3.1427	2.1624	2.1620	2.1513
60	2.4294	0.6980	1.9739	1.9870	2.0154	3.5900	3.5170	2.1505	2.1774	2.0237
64	2.3757	0.1346	2.5765	2.5495	2.2935	2.3002	2.3140	2.4699	2.2561	2.2458

Conclusion	The assertion evaluates true. $O(n^2)$ is a subset of $O(n^m)$

Case ID	Different price (PP2)
Category	Increasing problem size – increasing demand
Description	In a network where the price is varying across all products and profitable, the channel capacity sufficient and no need for network capacity negotiations (unconstrained network capacity), changing the number of products should according to theory on computational complexity exhibit strong exponential growth.
Network	1-4-16
Assertions	Assertion 1: Where the price is different across all products and profitable, the channel capacity sufficient and no need for network capacity negotiations (unconstrained network capacity), changing the number of products should according to theory on computational complexity exhibit strong exponential growth. Assertion 2: The test with different SKU prices (PP2) should come out at a lower runtime growth than one where the SKU prices are identical (PP) because the price differences result in a faster ranking of the bids in the bidding process in the RDN.
Results	



Results Data

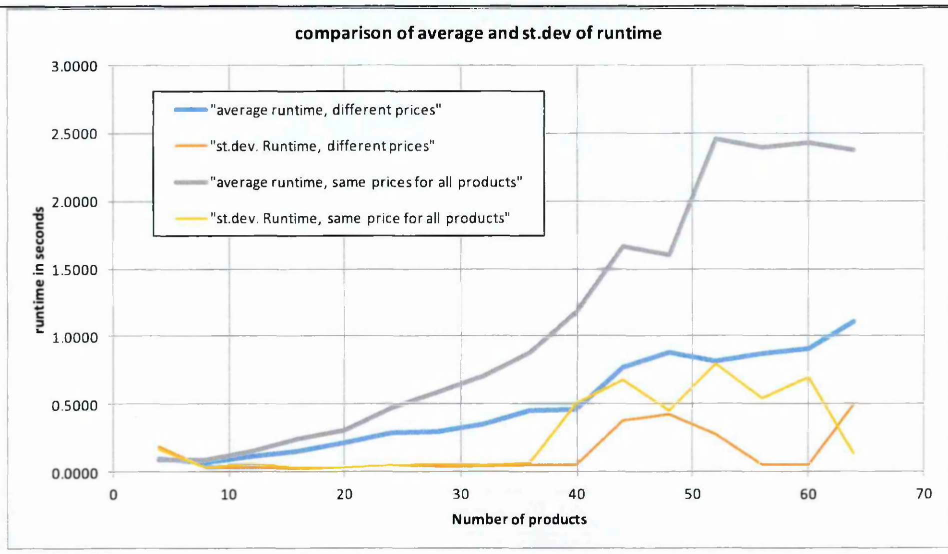
Number of products	Number of candidate solutions average	standard deviation	Experiment No. 1	2	3	4	5	6	7	8
4	5,439	731	4,435	4,941	6,706	5,986	5,413	5,867	4,819	5,345
8	14,035	1,043	14,404	16,071	12,479	14,041	14,218	14,073	13,110	13,885
12	25,787	2,541	30,574	27,850	24,636	27,373	23,790	24,719	23,403	23,947
16	38,556	2,911	38,988	36,723	39,568	40,283	40,875	39,377	32,094	40,543
20	56,292	4,986	55,636	67,991	56,796	55,188	55,844	53,704	53,266	51,911
24	74,693	8,914	91,839	75,914	76,261	72,903	59,885	70,389	72,276	78,074
28	82,730	9,456	88,483	77,848	104,135	75,761	77,637	79,773	79,901	78,302
32	96,300	10,225	107,724	98,826	98,967	86,915	109,636	79,270	98,109	90,954
36	119,715	12,976	149,470	117,130	105,866	113,517	121,787	114,505	120,646	114,800
40	118,345	12,110	118,448	145,327	115,106	109,952	106,997	124,412	112,942	113,573
44	176,704	18,906	184,104	171,780	176,642	203,372	152,936	167,821	201,527	155,450
48	152,345	19,205	146,666	173,504	170,182	179,901	139,610	142,263	129,045	137,592
52	165,587	18,743	162,059	180,034	176,916	140,422	169,137	196,083	145,060	154,986
56	216,278	17,527	240,235	206,244	231,842	198,147	225,702	225,675	212,857	189,519
60	235,904	23,587	277,049	257,902	219,405	238,067	208,443	242,927	232,557	210,880
64	230,548	17,706	259,369	227,197	241,096	218,433	204,293	234,084	243,956	215,952

Number of products	time (seconds) average	standard deviation	Experiment No. 1	2	3	4	5	6	7	8
4	0.0924	0.1835	0.5459	0.0214	0.0250	0.0499	0.0234	0.0280	0.0223	0.0231
8	0.0698	0.0304	0.1354	0.0509	0.0497	0.0941	0.0550	0.0515	0.0685	0.0535
12	0.1121	0.0322	0.1516	0.0931	0.1015	0.1724	0.0998	0.0882	0.0828	0.1072
16	0.1474	0.0256	0.1436	0.1307	0.1235	0.1933	0.1638	0.1618	0.1137	0.1486
20	0.2113	0.0315	0.2795	0.2199	0.1986	0.2125	0.1968	0.1942	0.2168	0.1721
24	0.2886	0.0470	0.3653	0.2490	0.2780	0.3296	0.2155	0.2688	0.3095	0.2935
28	0.3008	0.0445	0.3901	0.2744	0.3334	0.2735	0.3040	0.2599	0.3101	0.2610
32	0.3474	0.0448	0.3391	0.3744	0.3531	0.3329	0.3170	0.2925	0.4410	0.3294
36	0.4507	0.0532	0.5344	0.4462	0.3792	0.4676	0.4423	0.4546	0.5002	0.3813
40	0.4608	0.0554	0.4716	0.5868	0.4079	0.4217	0.4400	0.4701	0.4522	0.4363
44	0.7747	0.3758	0.6204	0.6329	0.5908	0.7290	0.5905	0.5893	0.7532	1.6913
48	0.8848	0.4293	1.5092	0.7071	0.5683	0.6154	0.5781	1.5560	0.5212	1.0227
52	0.8127	0.2795	1.1471	0.7233	0.6847	0.5973	0.6578	1.3473	0.5856	0.7586
56	0.8696	0.0526	0.9069	0.8609	0.8121	0.8251	0.9336	0.8469	0.8246	0.9467
60	0.9066	0.0543	0.9398	0.8674	0.8651	0.8978	1.0102	0.8605	0.8649	0.9471
64	1.1111	0.4995	1.0091	0.9263	2.3440	0.8921	0.9477	0.9038	0.9478	0.9181

Conclusion

The assertions 1 & 2 evaluates as True.

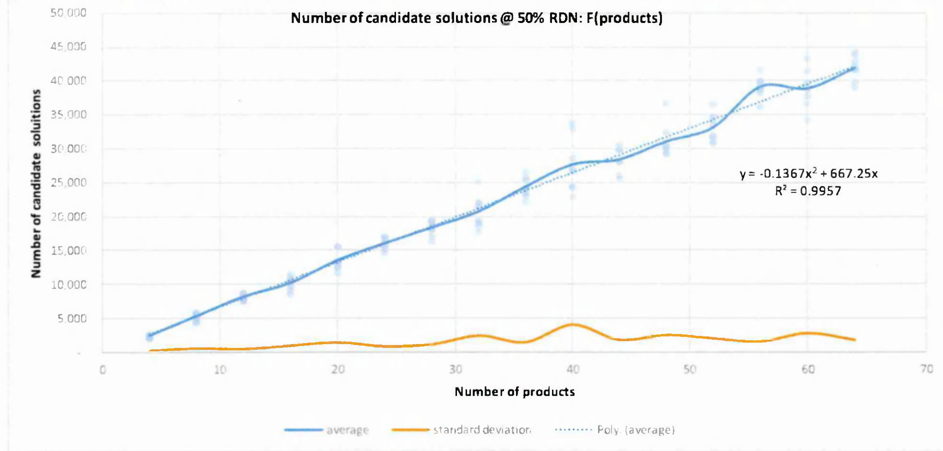
Most interestingly the second assertion (different prices) evaluates as 1 order of magnitude faster than the first (same prices). See comparison of absolute runtime below.

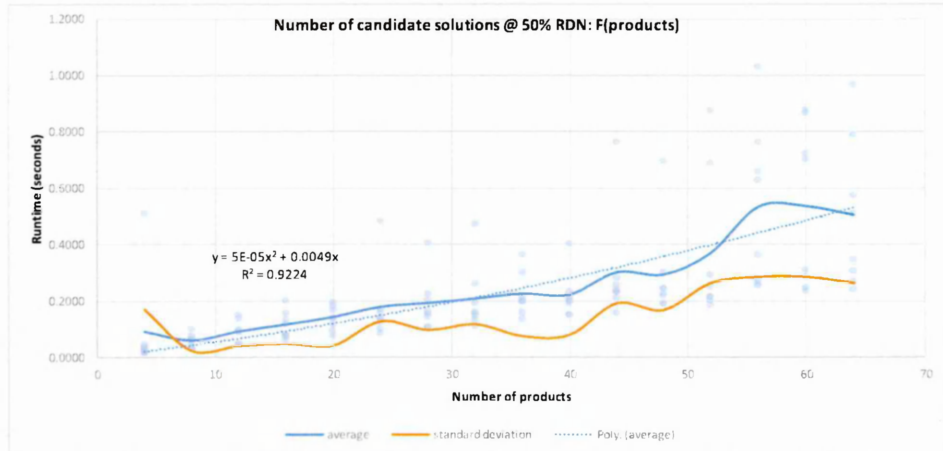


The logical conclusion is that there is no benefit of comparing more offers of the same prices, whereby they might as well be clustered into groups. On the other hand when there is a difference, ranking the most beneficial prices first, makes the decision a lot easier. This leads to the investigation that reduction of length of alternative may be an efficient way of controlling runtime (PP3)

Case ID	Extra Case: Option to reduce resource demand networks (RDN) bidding size (PP3)
Category	Increasing problem size – increasing demand
Description	The tests PP and PP2 prompted for the ability to control/manage the runtime based on the size of the list of options in the bidding process in the RDN network. From earlier: *The network is a 1-4-16 network where the price is constant across all products and profitable, the channel capacity sufficient and no need for network capacity negotiations (unconstrained network capacity), changing the number of products should according to theory on computational complexity exhibit strong exponential growth.* The test is to evaluate the effect of reducing the RDN network to 50% of its listed size in the test. As the RDN will have a candidate matrix of 100% * 100% of the candidates, the reduction of candidates to 50% * 50% is expected to narrow the scope effectively to 25%. This should result in a relative improvement in runtime with a factor of 4.
Network	1-4-16
Assertions	Assert whether the runtime decreases relatively to case PP2 with a factor of 2.

Results





Results Data

Number of products	Number of candidate solutions		Experiment No.							
	average	standard deviation	1	2	3	4	5	6	7	8
4	2,422	237	2,049	2,573	2,535	2,721	2,283	2,665	2,337	2,213
8	5,276	581	5,810	5,440	5,946	4,430	4,978	4,573	5,824	5,208
12	8,146	465	7,671	7,948	7,617	8,343	8,757	8,072	8,849	7,910
16	10,162	954	8,606	10,603	11,504	10,216	11,056	9,180	9,770	10,362
20	13,538	1,407	15,685	13,474	12,693	12,482	15,530	13,436	13,309	11,696
24	16,033	839	15,220	16,912	14,668	15,679	16,146	16,027	16,421	17,192
28	18,358	1,134	19,211	18,190	18,718	19,382	18,125	17,172	19,673	16,390
32	20,742	2,421	21,461	18,680	19,455	17,798	22,126	19,146	22,102	25,170
36	24,396	1,451	24,189	23,534	25,775	23,163	24,143	25,430	26,634	22,302
40	27,669	4,017	28,646	24,574	33,897	22,945	33,052	26,720	27,200	24,320
44	28,415	1,793	29,899	25,689	28,548	26,199	28,192	30,756	28,104	29,935
48	31,081	2,516	30,131	30,479	29,711	36,796	32,476	30,368	29,285	29,399
52	33,141	2,020	31,955	34,230	34,899	36,593	30,864	31,844	31,163	33,578
56	39,192	1,580	41,711	36,239	40,110	39,937	39,101	38,779	38,300	39,360
60	38,906	2,812	38,715	34,230	41,530	36,699	37,810	39,131	43,340	39,793
64	41,942	1,794	42,855	39,094	42,060	44,320	41,780	41,647	43,865	39,918

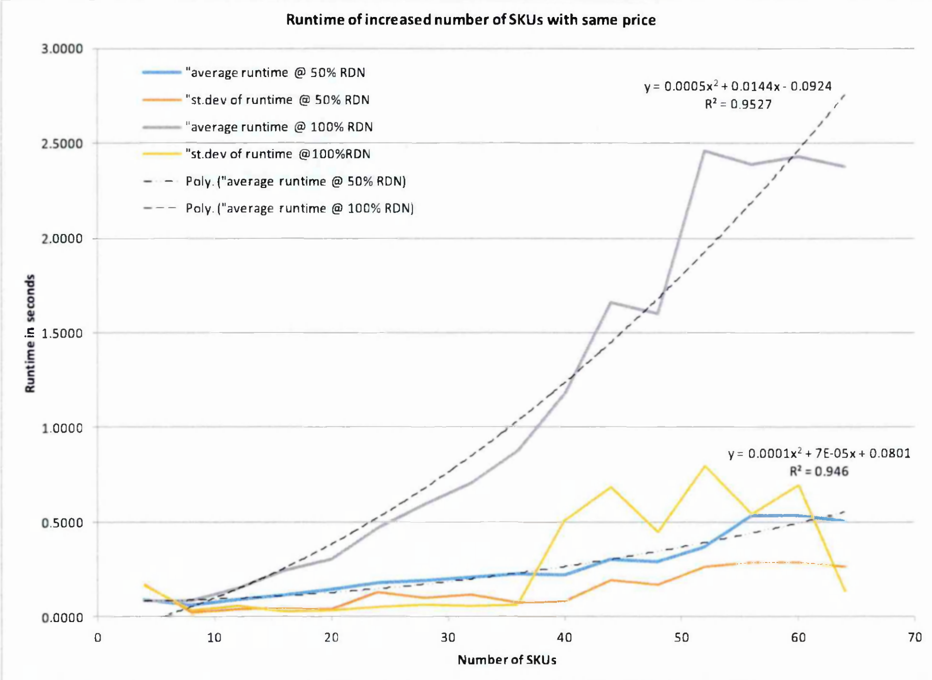
Heat maps are used to illustrate outliers in the experiments (below)

Number of products	time (seconds)		Experiment No.							
	average	standard deviation	1	2	3	4	5	6	7	8
4	0.0905	0.1714	0.5140	0.0186	0.0196	0.0368	0.0473	0.0269	0.0377	0.0232
8	0.0595	0.0242	0.1030	0.0388	0.0664	0.0563	0.0779	0.0336	0.0659	0.0339
12	0.0910	0.0409	0.0917	0.1002	0.0474	0.1409	0.1528	0.0509	0.0955	0.0485
16	0.1154	0.0489	0.1591	0.0847	0.0765	0.1362	0.2039	0.0628	0.1205	0.0791
20	0.1414	0.0422	0.1701	0.0962	0.0789	0.1418	0.1973	0.1416	0.1866	0.1186
24	0.1775	0.1279	0.1029	0.1263	0.0872	0.1738	0.4859	0.1366	0.1640	0.1430
28	0.1913	0.0973	0.1462	0.1573	0.1659	0.2053	0.4091	0.1107	0.2284	0.1078
32	0.2071	0.1177	0.1494	0.1346	0.1141	0.1587	0.4756	0.1639	0.2643	0.1959
36	0.2246	0.0754	0.2008	0.1623	0.2050	0.2168	0.3679	0.2004	0.3049	0.1391
40	0.2213	0.0797	0.2051	0.2041	0.1964	0.1530	0.4056	0.2359	0.2155	0.1545
44	0.3008	0.1922	0.2614	0.1610	0.7678	0.2345	0.2863	0.2370	0.2129	0.2456
48	0.2926	0.1683	0.1929	0.2482	0.6996	0.2486	0.3056	0.2292	0.2214	0.1950
52	0.3686	0.2645	0.1876	0.2171	0.6924	0.2959	0.2184	0.8797	0.2040	0.2539
56	0.5326	0.2868	0.3671	0.2576	0.6631	1.0352	0.2637	0.6321	0.7671	0.2753
60	0.5349	0.2866	0.3139	0.2897	0.8707	0.7062	0.2512	0.7264	0.8810	0.2398
64	0.5037	0.2648	0.3095	0.2434	0.5772	0.9709	0.3509	0.7934	0.5135	0.2706

<table>
<tr><td>Conclusion</td><td>The assertion holds true.

The table below illustrates the difference
<table>
<tr><td></td><td>Price model</td><td>Solution @ 60</td><td>Runtime @ 60</td></tr>
<tr><td>PP</td><td>Same price</td><td>1000,000</td><td>2.4294 sec</td></tr>
<tr><td>PP2</td><td>Different prices</td><td>225,000</td><td>0.9066 sec</td></tr>
<tr><td>PP3</td><td>Same price</td><td>40,000</td><td>0.5349 sec</td></tr>
</table>

As PP to PP3 the ratio is 2.4294 / 0.5349 = 4.54 x runtime reduction.



The use of the feature called "limited product options", which limits the number of options in the RDN bidding round from a exponent of $0.0005x^2$ to $0.0001x^2$. Though the sample size is sensitive to noise (see st.dev) the result is within a 25% signal-to-noise ration.</td></tr>
</table>

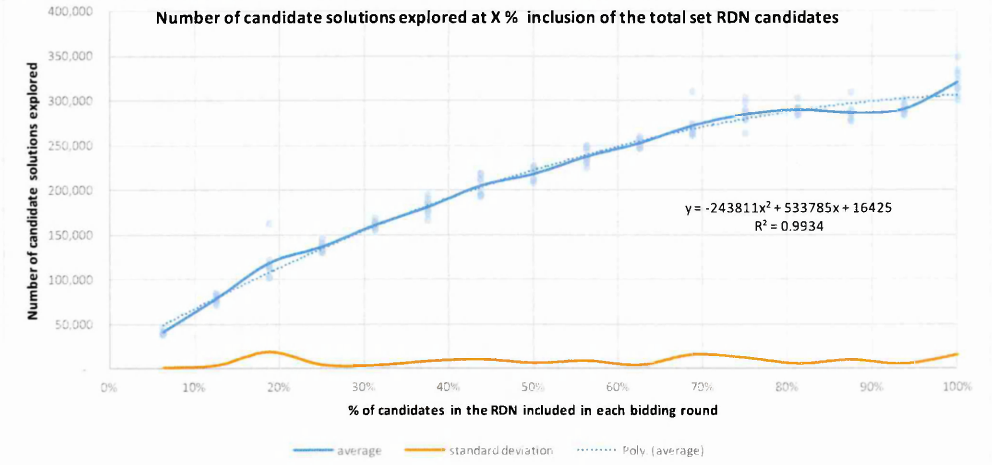
Case ID	Extra Case: Relaxation of reduced RND bidding size (PPO)
Category	Increasing problem size – increasing demand
Description	This is a logical extension of the previous test, From earlier: *The network is a 1-4-16 network where the price is constant across all products and profitable, the channel capacity sufficient and no need for network capacity negotiations (unconstrained network capacity), changing the number of products should according to theory on computational complexity exhibit strong exponential growth.* Where the number of options invited in the bidding round in the RDN round is gradually relaxed from 94% to 0% exclusion. Not about the size of the problem, but about logical extension of the number of options % percentage represents the length of the option-list for each competitive round.
Network	1-4-16

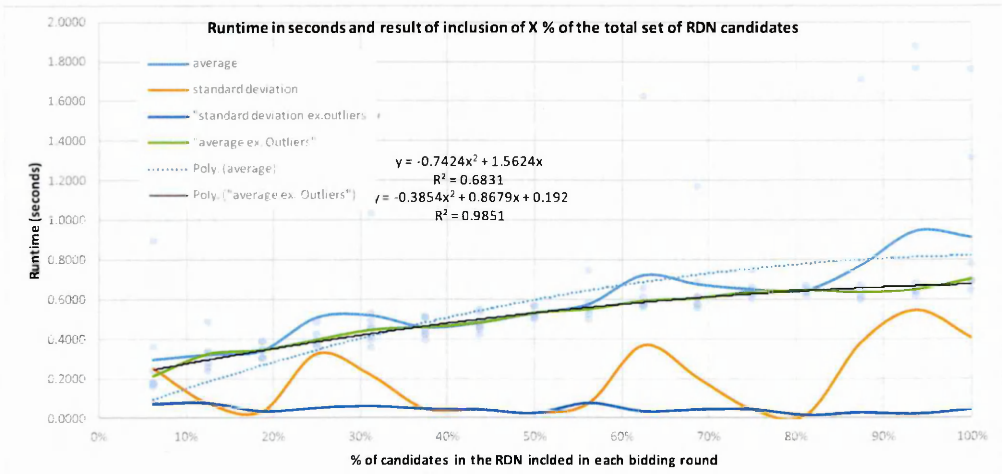
Assertions

The test asserts that usage of the API option "limited product options" will exhibit the following behaviour:

Setting	Scheduling speed	Result variation
Value close to 0	Faster	Higher variation
Value close to 1	Slower	Low variation

Results





Results Data

% of products in RDN	Number of candidate solutions		Experiment No.							
	average	standard deviation	1	2	3	4	5	6	7	8
6%	41,034	2,184	39,446	41,299	40,964	40,612	45,826	41,081	38,281	40,761
13%	78,298	4,526	84,339	84,051	73,570	77,713	78,570	76,079	80,263	71,798
19%	118,496	19,086	112,706	103,411	115,770	162,831	101,851	113,397	121,467	116,534
25%	136,704	5,088	140,769	145,692	139,413	130,807	133,313	136,749	135,570	131,318
31%	160,868	4,418	168,487	158,861	163,974	163,560	156,275	161,833	158,611	155,342
38%	181,012	8,885	189,310	176,476	174,018	185,451	179,012	166,517	194,294	183,021
44%	205,032	10,872	218,590	192,810	209,820	212,966	194,485	217,474	199,802	194,309
50%	218,019	6,920	224,442	220,120	211,454	222,255	211,084	208,050	219,938	226,811
56%	237,567	9,155	248,289	245,053	230,320	224,590	249,622	236,666	234,856	231,140
63%	252,684	4,696	252,081	248,951	246,905	253,755	257,629	247,609	254,692	259,846
69%	272,192	15,901	309,953	273,809	262,379	261,814	271,785	269,436	264,947	263,409
75%	284,900	12,452	263,215	289,836	303,403	280,133	278,691	297,588	286,339	279,994
81%	290,082	6,166	284,846	290,773	291,643	303,302	289,707	291,147	284,470	284,766
88%	286,748	10,337	284,568	278,521	279,923	288,065	289,679	277,113	309,675	286,436
94%	290,852	5,958	284,166	287,162	284,779	290,735	295,274	288,424	294,841	301,431
100%	320,718	15,716	329,102	305,266	314,064	300,718	321,120	333,465	348,345	313,660

Heat maps are used to highlight outliers in the dataset (see below).

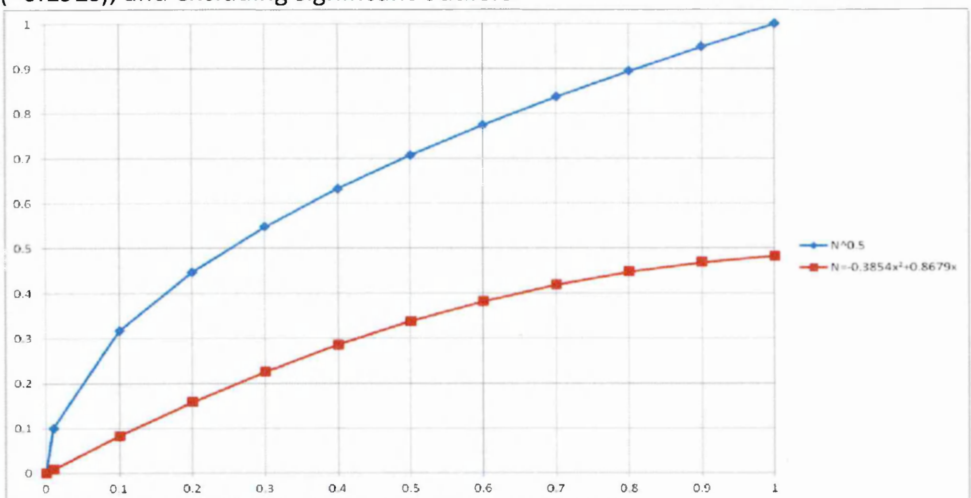
% of products in RDN	time (seconds) average	time (seconds) standard deviation	Experiment No. 1	2	3	4	5	6	7	8
6%	0.2972	0.2521	0.8991	0.1733	0.1602	0.1755	0.1788	0.3627	0.1855	0.2424
13%	0.3226	0.0759	0.4893	0.3323	0.2414	0.3398	0.3166	0.3200	0.2810	0.2608
19%	0.3424	0.0349	0.3931	0.3092	0.3039	0.3880	0.3076	0.3427	0.3445	0.3498
25%	0.5083	0.3259	0.4878	0.3935	1.3062	0.3246	0.3743	0.4268	0.3863	0.3666
31%	0.5185	0.2167	0.4946	0.4122	1.0366	0.4659	0.3580	0.4548	0.5319	0.3935
38%	0.4595	0.0465	0.5118	0.4242	0.5088	0.4548	0.3966	0.4243	0.5156	0.4398
44%	0.4816	0.0429	0.5378	0.4499	0.5499	0.4843	0.4622	0.4812	0.4631	0.4241
50%	0.5302	0.0250	0.5654	0.5248	0.5241	0.5240	0.5015	0.5229	0.5081	0.5709
56%	0.5747	0.0752	0.5864	0.7488	0.5636	0.5270	0.5683	0.5460	0.4999	0.5577
63%	0.7201	0.3681	0.5715	0.6113	0.5704	0.6583	0.5629	1.6278	0.5837	0.5752
69%	0.6747	0.2043	0.6859	0.6151	0.5561	0.5791	0.5716	1.1708	0.6128	0.6058
75%	0.6498	0.0438	0.6051	0.6322	0.6505	0.6631	0.6280	0.7498	0.6349	0.6345
81%	0.6449	0.0152	0.6718	0.6198	0.6572	0.6415	0.6461	0.6438	0.6441	0.6352
88%	0.7697	0.3816	0.6478	0.6091	0.6465	0.6162	1.7121	0.5981	0.6747	0.6535
94%	0.9428	0.5461	0.6835	0.6208	0.6382	0.6456	1.7686	0.6451	1.8828	0.6581
100%	0.9115	0.4076	0.7833	0.6981	0.6993	0.6483	1.7637	0.6982	1.3162	0.6852

Conclusion

The assertion holds true.

The expectation is that the explored solution landscape grows with square root characteristics, as the $solution\ space = demand\ points * products * \sqrt{\% * \ length\ of\ RDNlists}$,

which is visible when the recorded data is presented without the context creation time (+0.192s), and excluding significant outliers:



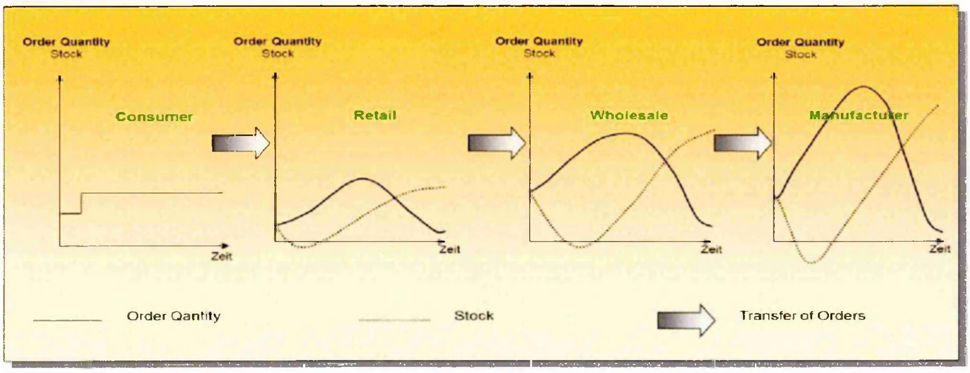
N	N^0.5	N=-0.3854x²+0.8679x
0%	0	0
1%	0.1	0.00864046
10%	0.316227766	0.082936
20%	0.447213595	0.158164
30%	0.547722558	0.225684
40%	0.632455532	0.285496
50%	0.707106781	0.3376
60%	0.774596669	0.381996
70%	0.836660027	0.418684
80%	0.894427191	0.447664
90%	0.948683298	0.468936
100%	1	0.4825
	correlation	0.99085862

This presents that the polynomial with least R^2 has a growth curve which is identical to the square-root function.

Increasing volatility of demand

Case ID	Increasing volatility of demand (XOX)
Category	Increasing problem size – increasing demand

<table>
<tr><td>Description</td><td>The objective of this test was to verify whether it is possible to re-create the bullwhip effect.



However this is not possible as the scheduling sites do not create their own forecast, nor operates based on responding to single orders.

The scheduler uses the full demand horizon of any downstream nodes which prevents the oscillations that may result in the bullwhip effect.

To re-create the bullwhip effect it would require that the schedulers are set up in isolation and not parsed a whole scheduling horizon. However as just doing "order processing" would defeat the purpose of having a schedule, this test doesn't make sense.</td></tr>
<tr><td>Network</td><td>1-1-1-1</td></tr>
<tr><td>Assertions</td><td>N/a</td></tr>
<tr><td>Results</td><td>N/a</td></tr>
<tr><td>Results Data</td><td>N/a</td></tr>
<tr><td>Conclusion</td><td>N/a</td></tr>
</table>

Increasing problem complexity

Assertions of consistency of scheduling quality and speed

The purpose of measuring scheduling quality and speed is to develop evidence of how quickly the scheduler incorporates changes for a generic network. In addition to verifying the quality of the scheduling process based on:

- The number of messages exchanged,
- The number of options explored,
- The time to reach a conclusion.

The real world application of this knowledge will be required to perform runtime predictions types of network (industrial feature). Because of the different levels of network complexity and the stochastic nature of the demand 3 cases are considered:

1. Synthetic network
    a. For a given synthetic network we test the variance of the KPI's over multiple iterations with similar demand and capacity.
    b. Performance for increasing number of products.
    c. Performance for increasing number of demands (horizon).
    d. Performance for increasing number of channels.
    e. Sensitivity of runtime + KPIs under competing limits.

2. For large network (TV Group case data)
    a. Linearly increase the number of orders until lost sales occur.
    b. Freeze orders at a high utilization level and add more network elements
3. For complicated network (CMX case data)
    a. Linearly increase the number of orders until lost sales occur.
    b. Freeze orders at a high utilization level and add more network elements

Beyond comparative measure of for runtime, this analysis will give the version 0.1 of the runtime prediction algorithms.

Explanation of run-time of adaptive vs. batch-mode

The purpose of measuring run-time in adaptive and batch mode is to develop evidence of which mode of operation is more appropriate for a list of common changes that consultants perform in their development of network models. This means that the list of evaluations must be based on the common types of changes, which are presented in the overview below: (**Bold tests** are in the current report, others are for later):

1) NETWORK CHANGES:
    a) Add/remove channel anywhere in the network (f.x. ship directly from Source to Customer instead of via point X).
    b) Add/remove DC with 1-2 channels (storage)
    c) Add/remove fully connect DC (storage)
    d) Change lead-time on a single channel
    e) Change lead-time on all channels
    f) **Add/reduce bandwidth (shift from FTL to van's or pallets) ****
    g) Add single customer site (changes at the extremities of the network)
    h) Add small storage for a particular site
    i) **New alternative supplier ****
    j) Close channel for "Holiday Shut-down"

2) STORAGE CHANGES:
    a) Change storage capacity limit
    b) Change storage cost
    c) Close receiving / dispatch for holiday shut down

3) PRODUCT CHANGES:
    a) Add/remove product for full time horizon
    b) Add/remove product for promotion period (less than full horizon)
    c) Demand change
    d) Change in Target Service Level(TSL)

4) TRANSPORT CHANGES:
    a) **Change a single transport rate ****
    b) **Change all transport rates ****

5) PRODUCTION LINES CHANGES:
    a) New minimum order quantity
    b) New production line capacity
    c) Change MOQ/MINC

d) Engineering change – BoM change after a future date
e) Unscheduled Maintenance (3 day shutdown)

The time the model will use to incorporate a change is anticipated to be correlated to the relative size of a change in comparison to the number of objects in the network. This means that there will be situations where it is more feasible to rerun the model as a batch-job. They key result from this analysis is therefore to identify the strategy which can provide a reasonable detection of the propagated size of the changes and automatically select the method which has the shortest runtime.

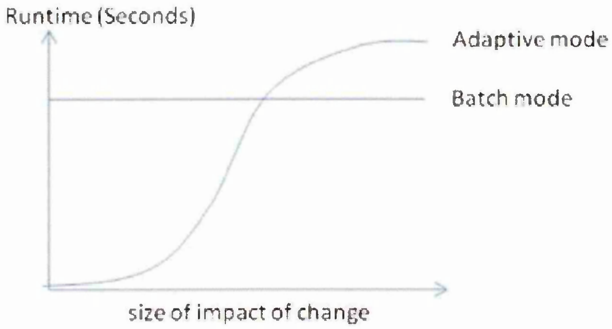
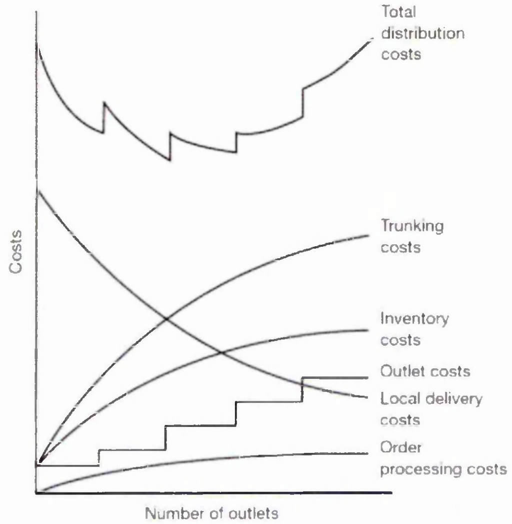
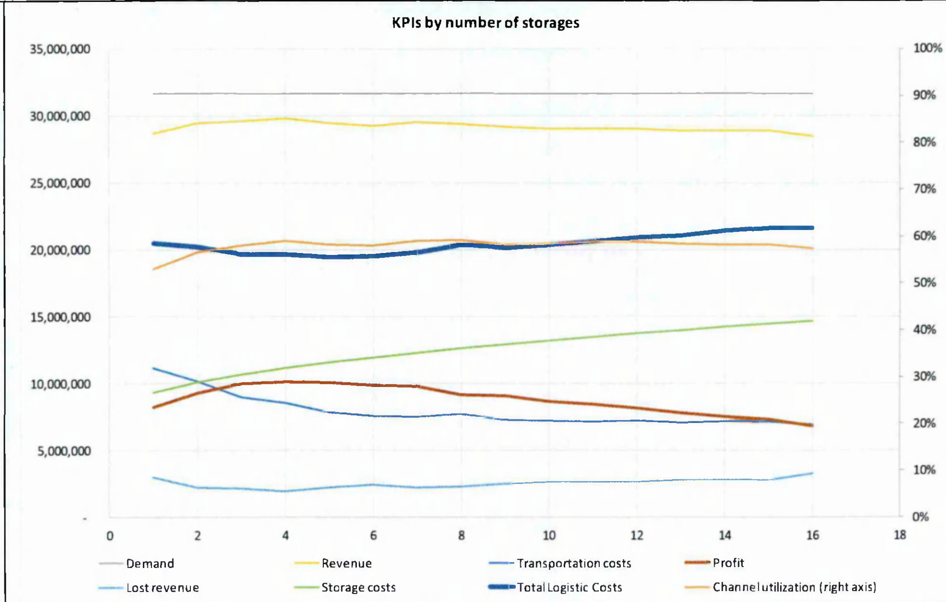


Figure 46 Expected performance of adaptive vs. batch-mode for incorporating a given size of change in a loaded context.

Increasing number of storages

Case ID	Increasing number of storage sites (SA4)
Category	Increasing problem complexity
Description	The test "demand analysis DA" produced transition zones at iteration 0,6 and 12. In this evaluation we keep the value of demand constant at € 31,680,000. The demand thereby reflects DA's iteration 4. The expectation is to re-create the chart below by varying the number of storages (outlets in the context of the figure below).  Figure 47 from Christopher, M. (2007) Logistics & SCM, p. 109, fig. 3.7
Network	Circular 1-N-16
Assertions	Assert that the variation (increase & reduction) of number of storages (outlets) produces a figure similar to the one in the description.

Results

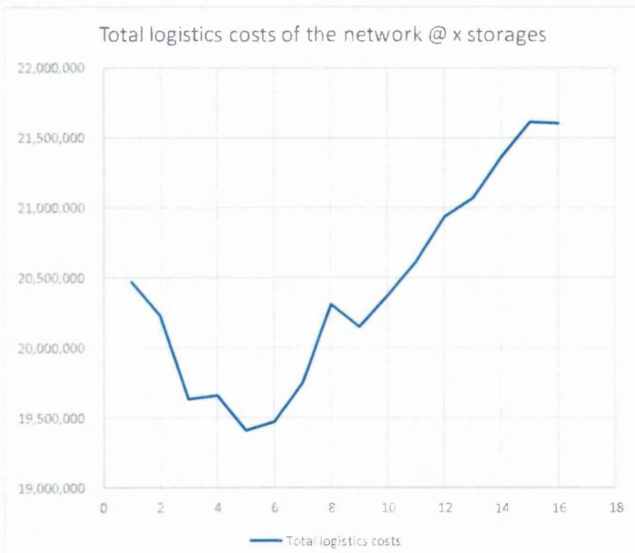


Results Data

Number of storage sites	Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	Total Logistic Costs
1	31,680,000	28,689,600	11,123,382	8,221,218	2,990,400	53%	9,345,000	20,468,382
2	31,680,000	29,481,600	10,147,474	9,251,826	2,198,400	57%	10,082,300	20,229,774
3	31,680,000	29,582,400	8,984,449	9,949,900	2,097,600	58%	10,648,050	19,632,500
4	31,680,000	29,787,200	8,533,814	10,128,386	1,892,800	59%	11,125,000	19,658,814
5	31,680,000	29,461,600	7,866,552	10,049,847	2,218,400	58%	11,545,201	19,411,753
6	31,680,000	29,282,400	7,543,959	9,813,349	2,397,600	58%	11,925,092	19,469,051
7	31,680,000	29,518,400	7,470,450	9,773,513	2,161,600	59%	12,274,437	19,744,887
8	31,680,000	29,415,600	7,712,629	9,103,371	2,264,400	59%	12,599,600	20,312,229
9	31,680,000	29,186,000	7,244,138	9,036,862	2,494,000	58%	12,905,000	20,149,138
10	31,680,000	29,041,600	7,179,939	8,667,806	2,638,400	58%	13,193,854	20,373,794
11	31,680,000	29,027,600	7,146,687	8,412,321	2,652,400	59%	13,468,592	20,615,279
12	31,680,000	29,068,000	7,204,088	8,132,811	2,612,000	59%	13,731,101	20,935,189
13	31,680,000	28,886,800	7,082,032	7,821,887	2,793,200	58%	13,982,881	21,064,913
14	31,680,000	28,901,200	7,137,861	7,538,189	2,778,800	58%	14,225,150	21,363,011
15	31,680,000	28,887,200	7,150,960	7,277,329	2,792,800	58%	14,458,911	21,609,871
16	31,680,000	28,448,800	6,922,707	6,841,093	3,231,200	57%	14,685,000	21,607,707

Conclusion

The assertion holds true. Because of the range of the KPIs, the chart above does not give the iconic depiction which the data reveals. In fact, when the total logistics costs are illustrated in isolation, the U-shaped curve is very visible.



Case ID	Increasing number of storage sites (SA8)

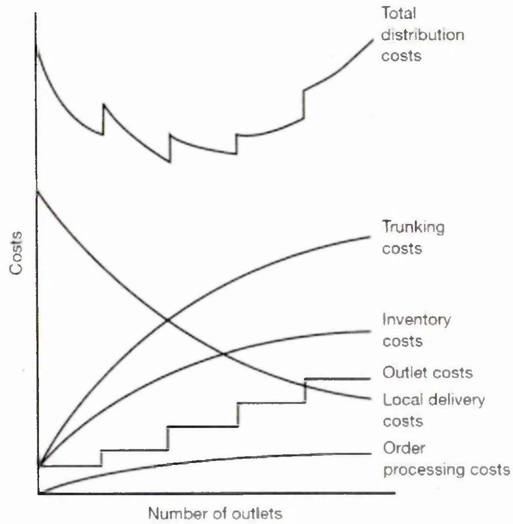
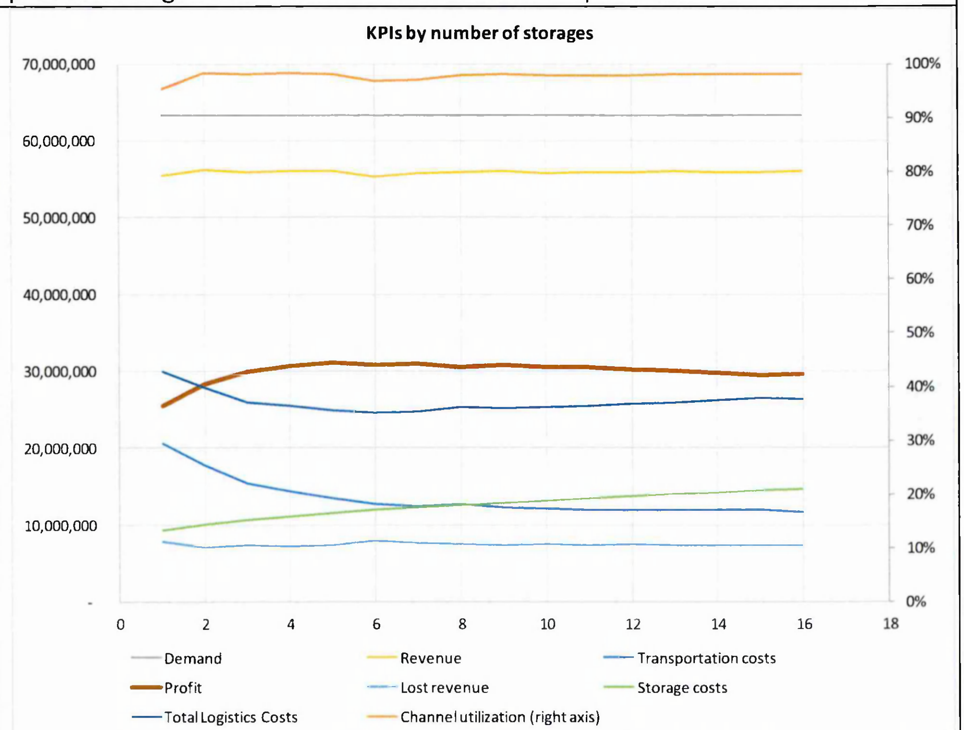
<table>
<tr><td>Category</td><td>Increasing problem complexity</td></tr>
<tr><td>Description</td><td>The test “demand analysis DA” produced transition zones at iteration 0,6 and 12.
In this evaluation we keep the value of demand constant at € 63,360,000. The demand thereby reflects DA’s iteration 8, where it should be noted that this is the point where channel capacity limit is reached.
The expectation is to re-create the chart below by varying the number of storages (outlets in the context of the figure below).


Figure 48 from Christopher, M. (2007) Logistics & SCM, p. 109, fig. 3.7</td></tr>
<tr><td>Network</td><td>Circular 1-N-16</td></tr>
<tr><td>Assertions</td><td>Assert that the variation (increase & reduction) of number of storages (outlets) produces a figure similar to the one in the description.</td></tr>
<tr><td>Results</td><td>


</td></tr>
</table>

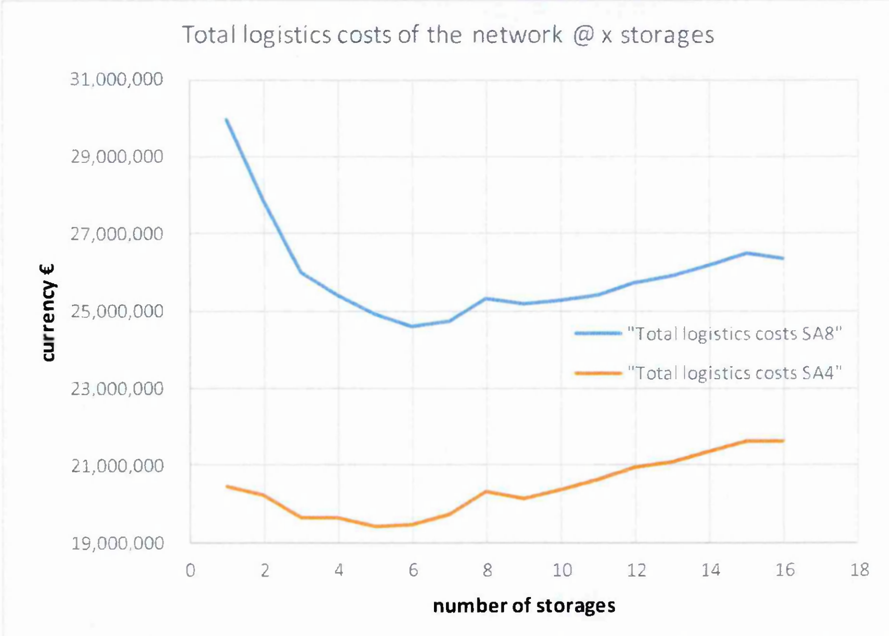
Results Data

Number of storage sites	Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	Total Logistics Costs
1	63,360,000	55,518,400	20,628,939	25,544,461	7,841,600	95%	9,345,000	29,973,939
2	63,360,000	56,240,000	17,805,084	28,352,616	7,120,000	98%	10,082,300	27,887,384
3	63,360,000	55,981,200	15,342,559	29,990,591	7,378,800	98%	10,648,050	25,990,609
4	63,360,000	56,140,800	14,289,873	30,725,927	7,219,200	98%	11,125,000	25,414,873
5	63,360,000	56,051,200	13,383,021	31,122,978	7,308,800	98%	11,545,201	24,928,222
6	63,360,000	55,420,000	12,662,099	30,832,809	7,940,000	97%	11,925,092	24,587,191
7	63,360,000	55,774,400	12,462,698	31,037,264	7,585,600	97%	12,274,437	24,737,136
8	63,360,000	55,891,200	12,739,124	30,552,475	7,468,800	98%	12,599,600	25,338,725
9	63,360,000	56,046,400	12,274,953	30,866,447	7,313,600	98%	12,905,000	25,179,953
10	63,360,000	55,869,600	12,096,563	30,579,183	7,490,400	98%	13,193,854	25,290,417
11	63,360,000	55,958,400	11,947,990	30,541,818	7,401,600	98%	13,468,592	25,416,582
12	63,360,000	55,928,000	11,984,224	30,212,675	7,432,000	98%	13,731,101	25,715,325
13	63,360,000	56,041,600	11,945,238	30,113,481	7,318,400	98%	13,982,881	25,928,119
14	63,360,000	56,011,200	11,958,895	29,827,155	7,348,800	98%	14,225,150	26,184,045
15	63,360,000	55,980,800	12,023,000	29,498,889	7,379,200	98%	14,458,911	26,481,911
16	63,360,000	56,033,600	11,665,535	29,683,065	7,326,400	98%	14,685,000	26,350,535

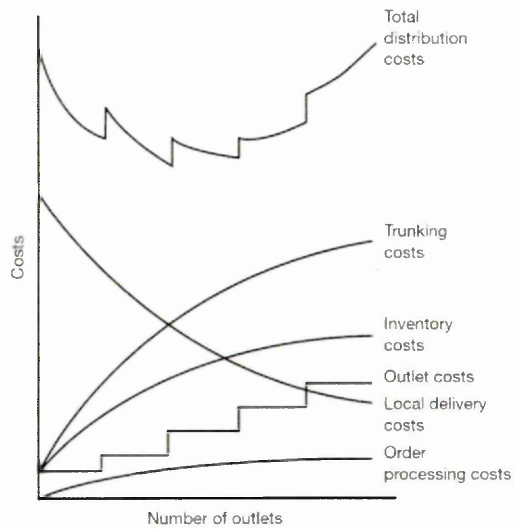
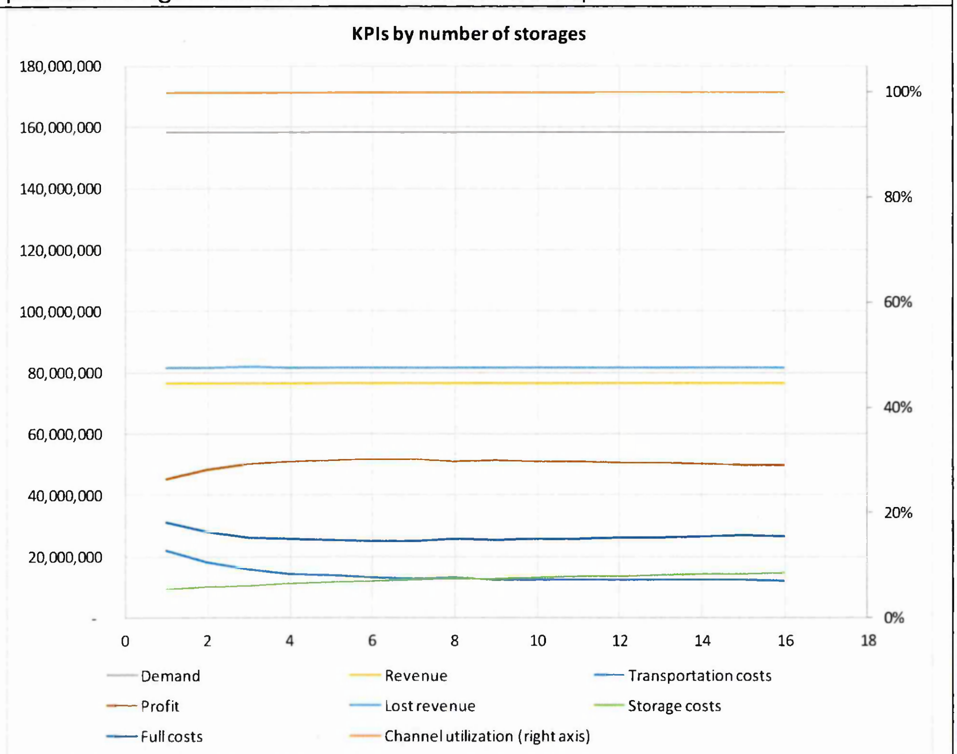
Conclusion

The assertion holds true. Because of the range of the KPIs, the chart above does not give the iconic depiction which the data reveals.

A particular notable element is that this is the point where demand equals total channel capacity. Hereby the cost in SA8 increase more than in SA4 as more volume needs to be moved in non-ideal transport channels.



Case ID	Increasing number of storage sites (SA20)
Category	Increasing problem complexity
Description	The test “demand analysis DA” produced transition zones at iteration 0,6 and 12. In this evaluation we keep the value of demand constant at € 158,400,000. The demand thereby reflects DA’s iteration 20, where it should be noted that this is the point where it by far exceeds the networks capacity. The expectation is to re-create the chart below by varying the number of storages (outlets in the context of the figure below).

	 Figure 49 from Christopher, M. (2007) Logistics & SCM, p. 109, fig. 3.7
Network	Circular 1-N-16
Assertions	Assert that the variation (increase & reduction) of number of storages (outlets) produces a figure similar to the one in the description.
Results	

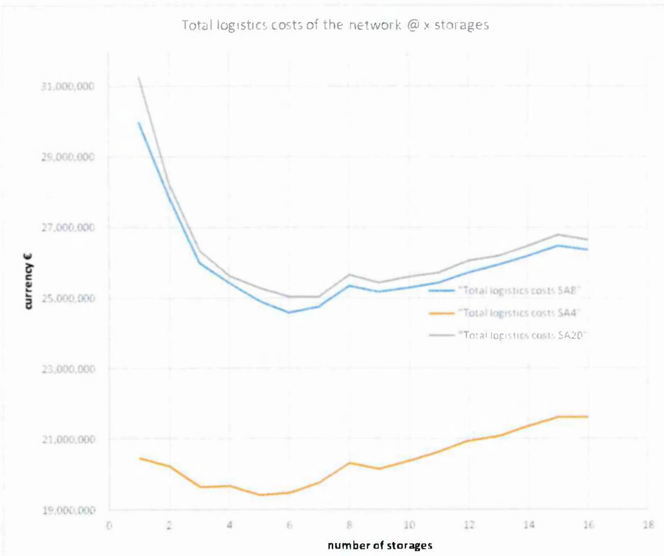
Results Data

Number of storage sites	Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	Full costs
1	158,400,000	76,672,000	21,902,402	45,424,598	81,728,000	100%	9,345,000	31,247,402
2	158,400,000	76,672,000	18,163,690	48,426,010	81,728,000	100%	10,082,300	28,245,990
3	158,400,000	76,640,000	15,684,287	50,307,663	81,760,000	100%	10,648,050	26,332,337
4	158,400,000	76,672,000	14,489,197	51,057,803	81,728,000	100%	11,125,000	25,614,197
5	158,400,000	76,672,000	13,741,882	51,384,917	81,728,000	100%	11,545,201	25,287,083
6	158,400,000	76,672,000	13,117,117	51,629,791	81,728,000	100%	11,925,092	25,042,209
7	158,400,000	76,672,000	12,757,773	51,639,790	81,728,000	100%	12,274,437	25,032,210
8	158,400,000	76,672,000	13,063,070	51,009,330	81,728,000	100%	12,599,600	25,662,670
9	158,400,000	76,672,000	12,524,036	51,242,964	81,728,000	100%	12,905,000	25,429,036
10	158,400,000	76,672,000	12,400,349	51,077,796	81,728,000	100%	13,193,854	25,594,204
11	158,400,000	76,672,000	12,241,862	50,961,546	81,728,000	100%	13,468,592	25,710,454
12	158,400,000	76,672,000	12,302,110	50,638,789	81,728,000	100%	13,731,101	26,033,211
13	158,400,000	76,672,000	12,208,315	50,480,804	81,728,000	100%	13,982,881	26,191,196
14	158,400,000	76,672,000	12,241,518	50,205,332	81,728,000	100%	14,225,150	26,466,668
15	158,400,000	76,672,000	12,334,230	49,878,859	81,728,000	100%	14,458,911	26,793,141
16	158,400,000	76,672,000	11,958,149	50,028,851	81,728,000	100%	14,685,000	26,643,149

Conclusion

The assertion holds true.

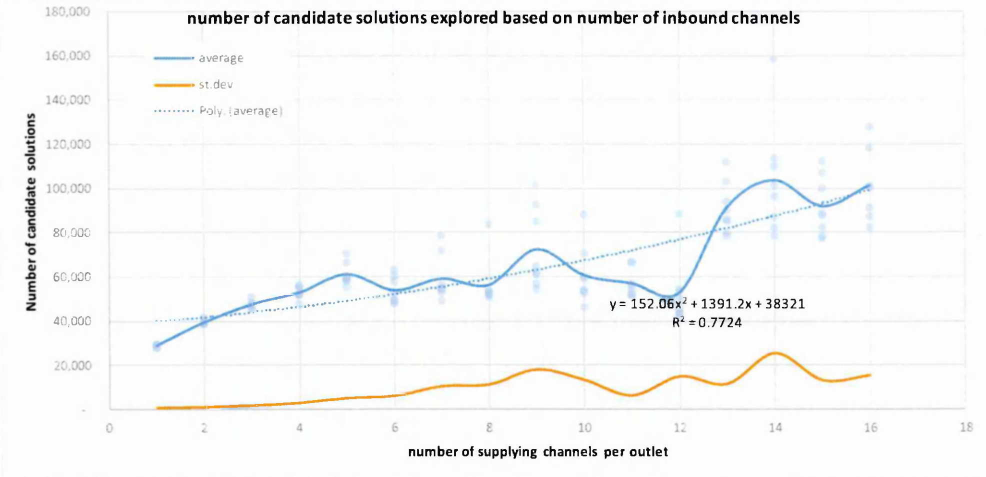
As the products have different prices the sole objective is to minimize costs in order to optimally profit from the system. The chart below shows the minor improvement possible be excluded lesser-than-maximally profitable products in comparison to SA8 where the channel capacity limit was reached.

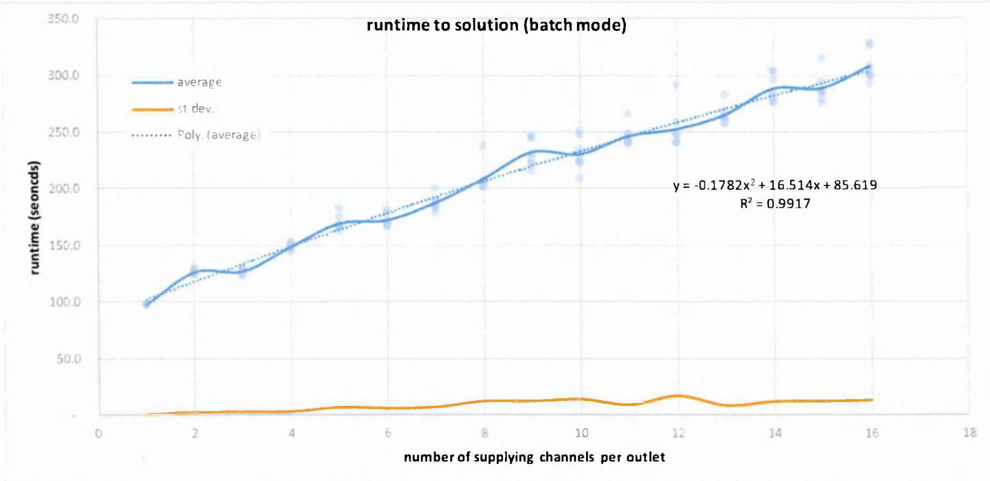


Increasing number of channels

Case ID	Excess channel capacity (PCh)
Category	Increasing problem complexity
Description	In a network with high product price where everything is profitable, sufficient channel capacity the effect of increasing the number of channels into each outlet is assessed.
Network	Starts as Linear 1-16-16 with a single channel, ends as Linear 1-16-16 fully connected.
Assertions	N/A

Results



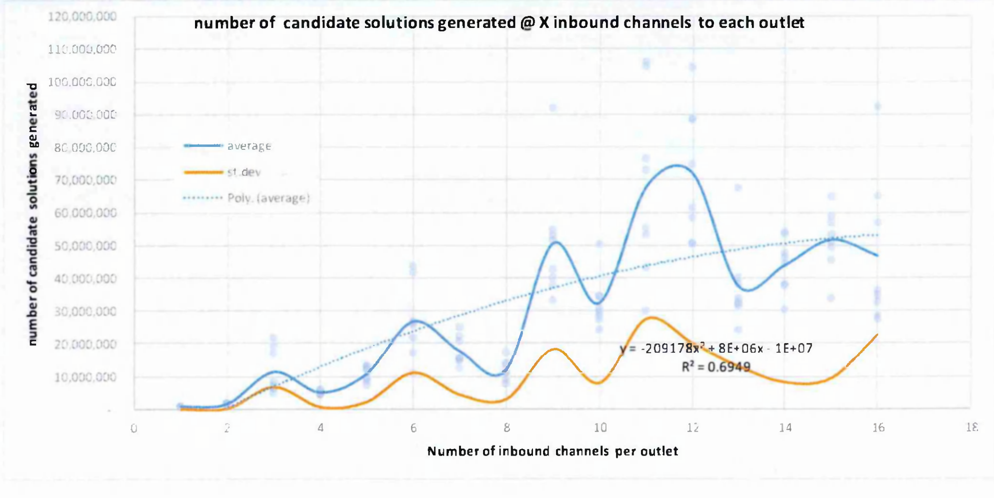


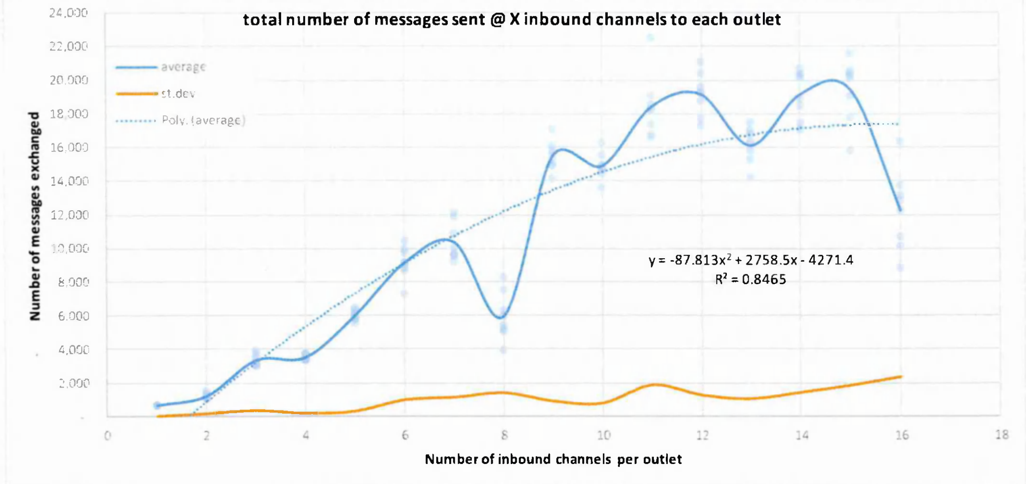
Results Data

number of	number of options		Experiment No.							
channels	average	st.dev	1	2	3	4	5	6	7	8
1	28,976	598	28,007	28,007	29,299	29,299	29,299	29,299	29,299	29,299
2	39,527	940	39,263	39,620	38,974	41,686	39,227	39,301	38,533	39,611
3	47,564	1,603	46,811	45,869	50,981	47,672	47,273	46,034	48,215	47,656
4	52,855	2,751	56,248	55,066	55,657	51,600	52,182	52,097	47,786	52,207
5	61,106	4,926	60,362	58,991	70,508	60,082	55,347	58,214	66,475	58,869
6	53,817	5,884	60,434	63,080	57,970	48,949	53,082	49,367	49,758	47,898
7	59,064	10,395	54,294	54,421	49,089	71,899	55,051	78,691	52,879	56,185
8	56,236	11,192	52,237	83,839	51,474	52,196	53,552	53,558	50,814	52,214
9	72,333	18,065	62,061	101,667	61,141	85,192	54,425	64,833	56,782	92,559
10	60,621	13,311	88,409	46,309	59,363	54,034	53,451	70,730	52,595	60,076
11	57,024	6,212	66,253	52,473	56,666	66,932	56,231	51,561	54,404	51,673
12	52,910	15,021	88,441	54,268	43,913	43,594	43,336	50,600	45,828	53,298
13	91,201	11,501	93,955	103,275	80,184	90,158	78,660	86,085	85,304	111,983
14	103,541	25,578	96,535	110,027	101,194	87,144	82,305	113,734	78,692	158,693
15	91,821	13,210	88,920	107,128	77,466	78,379	87,850	100,030	82,454	112,341
16	101,227	15,423	100,501	91,417	87,400	82,170	100,885	100,947	127,945	118,548

number of channels	time (seconds) average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
1	98.00	0.00	98.0	98.0	98.0	98.0	98.0	98.0	98.0	98.0
2	126.63	1.92	130.0	127.0	125.0	129.0	125.0	126.0	126.0	125.0
3	127.25	2.55	127.0	123.0	129.0	130.0	129.0	124.0	127.0	129.0
4	149.00	2.78	153.0	149.0	152.0	149.0	149.0	147.0	144.0	149.0
5	169.38	6.50	168.0	166.0	183.0	168.0	163.0	166.0	175.0	166.0
6	172.25	5.47	179.0	182.0	172.0	167.0	171.0	167.0	171.0	169.0
7	187.50	6.57	187.0	184.0	179.0	190.0	186.0	201.0	183.0	190.0
8	208.63	12.12	207.0	238.0	201.0	202.0	206.0	207.0	202.0	206.0
9	232.13	12.05	223.0	245.0	231.0	247.0	216.0	228.0	222.0	245.0
10	230.38	13.90	248.0	209.0	235.0	224.0	224.0	251.0	223.0	229.0
11	246.25	8.71	248.0	241.0	241.0	266.0	249.0	242.0	244.0	239.0
12	252.38	17.12	292.0	248.0	241.0	241.0	241.0	259.0	248.0	249.0
13	265.00	8.11	264.0	283.0	258.0	263.0	258.0	264.0	261.0	269.0
14	288.00	11.86	282.0	297.0	285.0	279.0	276.0	304.0	277.0	304.0
15	288.50	12.11	279.0	286.0	286.0	276.0	295.0	287.0	284.0	315.0
16	308.00	13.17	327.0	293.0	309.0	300.0	300.0	301.0	329.0	305.0

Conclusion	The results indicate that the growth in number of channels result in linear growth of runtime. This contradicts the classical theory which assumes that with increased number of options the runtime should increase exponentially $O(m^n)$. However as the multi agent systems RDN does not perform single sided search the runtime is reduced to $O(n)$.

Case ID	Limited channel capacity (PCh2)
Category	Increasing problem complexity
Description	In a network with high product price where everything is profitable, sufficient channel capacity, except between the factory and storages, the effect of increasing the number of channels into each outlet is assessed. This is a more complex case than PCh as the constraints aggregate at the source and not the storages.
Network	Starts as Linear 1-16-16 with a single channel, ends as Linear 1-16-16 fully connected.
Assertions	N/A
Results	



Results Data

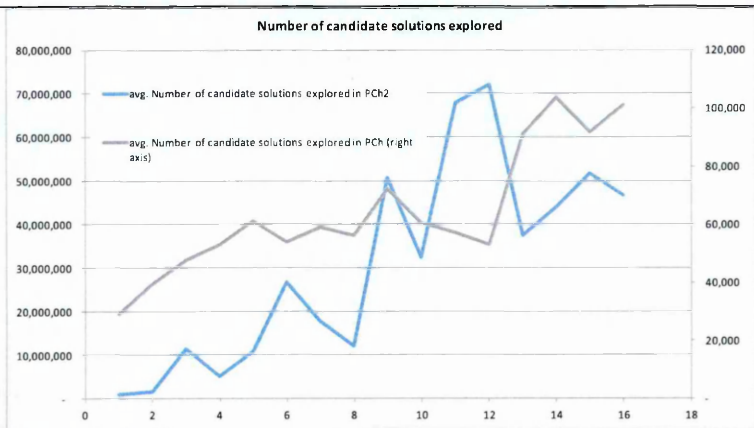
number of channels	number of options		Experiment No.							
	average	st.dev	1	2	3	4	5	6	7	8
1	828,022	1,892	827,243	828,976	828,976	828,976	828,976	828,976	828,472	823,583
2	1,632,649	168,214	1,772,768	1,366,760	1,542,746	1,697,059	1,859,070	1,775,612	1,526,419	1,520,755
3	11,460,404	6,753,782	21,786,548	5,071,800	19,294,584	17,127,612	6,507,268	6,417,879	7,147,923	8,329,621
4	5,109,183	587,617	4,737,937	5,093,670	4,724,936	5,977,991	4,574,057	5,456,925	4,459,270	5,848,674
5	10,860,138	2,195,345	13,307,335	9,926,748	12,161,464	9,449,475	12,521,314	7,506,522	8,865,707	13,142,541
6	26,806,979	11,183,189	43,677,110	26,534,127	17,207,586	41,712,128	10,993,205	21,797,190	25,674,623	26,859,865
7	17,700,382	4,389,428	25,027,212	14,715,676	20,787,770	22,240,988	15,294,009	12,537,574	15,488,700	15,511,130
8	12,053,738	3,018,416	7,760,800	11,056,585	14,145,437	12,145,396	13,566,702	11,350,241	9,046,298	17,358,446
9	50,646,523	18,390,655	33,240,028	42,748,752	54,738,232	40,236,657	51,477,699	91,973,723	52,892,321	37,864,772
10	32,438,969	7,982,710	30,744,771	28,506,222	50,263,849	34,538,096	34,321,627	27,346,520	24,188,410	29,602,260
11	67,780,356	27,599,486	53,211,215	30,062,282	76,514,951	43,226,275	55,337,360	106,061,474	104,795,829	73,033,459
12	72,149,823	20,066,713	74,742,113	88,625,841	50,700,219	50,350,944	58,439,022	61,502,069	104,352,976	88,485,403
13	37,498,669	13,037,905	33,707,804	40,130,934	32,153,139	67,511,547	32,663,586	31,500,788	38,216,672	24,104,880
14	43,951,172	8,172,495	44,323,025	47,431,234	37,742,714	53,513,455	53,937,252	30,323,541	38,075,755	46,262,397
15	51,721,619	9,427,934	53,325,301	56,829,191	49,546,905	58,911,526	64,917,947	51,038,839	45,463,375	33,739,864
16	46,711,784	22,936,024	27,469,700	28,457,244	32,597,667	34,677,116	65,019,629	92,425,180	56,769,313	36,278,419

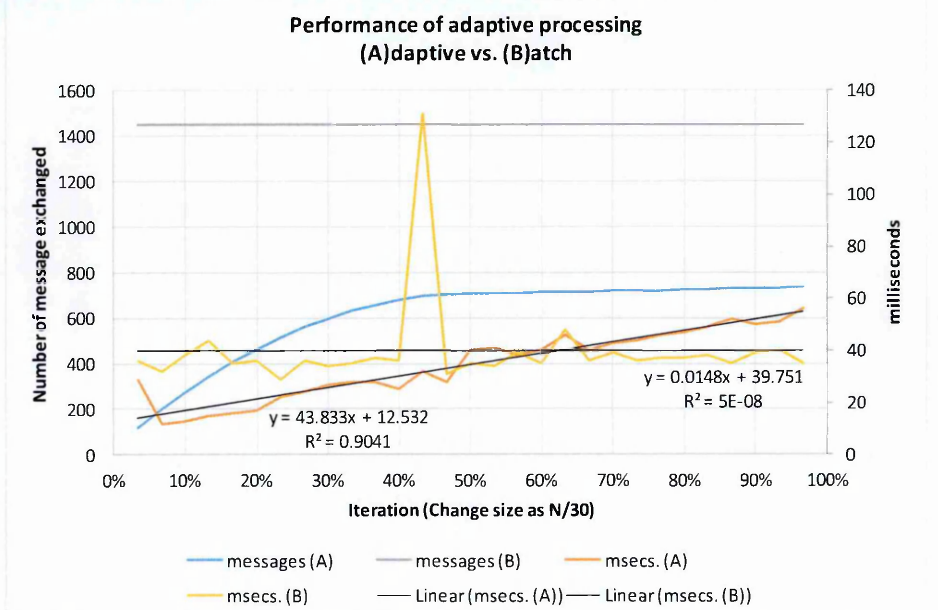
number of channels	number of messages		Experiment No.							
	average	st.dev	1	2	3	4	5	6	7	8
1	674	-	674	674	674	674	674	674	674	674
2	1,221	150	1,113	1,079	1,373	1,186	1,089	1,508	1,182	1,238
3	3,357	329	3,505	3,008	3,933	3,258	3,211	3,177	3,049	3,711
4	3,527	170	3,633	3,786	3,491	3,406	3,361	3,735	3,350	3,450
5	6,027	295	5,566	6,071	6,325	5,854	6,481	5,930	6,166	5,822
6	9,162	976	9,978	8,835	8,679	10,488	7,283	9,165	9,101	9,769
7	10,384	1,120	11,874	9,572	10,880	9,221	12,154	9,638	9,665	10,066
8	5,966	1,394	3,957	5,976	7,521	5,200	6,304	5,418	5,073	8,280
9	15,507	888	15,611	14,960	17,111	14,156	15,836	16,107	15,314	14,957
10	14,908	775	13,619	14,952	16,283	14,881	14,704	15,542	14,459	14,822
11	18,439	1,866	16,605	18,327	18,232	16,782	17,373	22,516	19,105	18,572
12	19,193	1,252	17,326	21,095	19,275	19,586	17,768	18,779	19,298	20,414
13	16,147	1,040	16,781	16,165	15,308	17,501	16,335	17,036	15,830	14,222
14	19,196	1,421	17,057	18,915	20,739	20,384	20,300	20,201	18,506	17,466
15	19,472	1,849	17,798	20,437	19,087	21,591	20,590	20,118	20,315	15,843
16	12,269	2,361	12,268	10,717	8,843	12,842	13,180	16,384	10,146	13,773

Conclusion

The complexity is easily visualised in number of candidate solutions explored as the upstream bottleneck results in the generation of a large set of useless candidate solutions further downstream.

The chart below shows the difference in absolute numbers, where PCh2 ranges from 0 to 80m and PCh only to 120,000. This gives an increase in number of candidate-solutions of approximately a factor of 600!



Case ID	Adaptive change of transport rates (AC)
Category	Increasing problem complexity
Description	In a network with 30 storages and 30 shops in a fully connected graph, the supply to the shops is scheduled in batch mode from the storages. In N=30 subsequent iterations, the transport costs on channels outbound from (N=iteration) storages is/are increased. After each iteration the scheduler is reset to the original context, so that the size of change increases linearly with N. The test case inspects the number of messages exchanged in adaptive and batch mode and the time required to incorporate the change Adaptive: number of messages & scheduling time Batch: number of messages & scheduling time
Network	10 distribution centres and 30 outlets fully connected. Thereby there are 300 channels. A change of 10% thereby affects 30 of the 300 channels.
Assertions	N/A
Results	

Results Data

Iteration	Size of change	Adaptive mode messages (A)	Adaptive mode msecs. (A)	Batch mode messages (B)	Batch mode msecs. (B)
1	3.3%	117	29	1455	36
2	6.7%	198	12	1455	32
3	10.0%	273	13	1455	38
4	13.3%	342	15	1455	44
5	16.7%	405	16	1455	35
6	20.0%	462	17	1455	36
7	23.3%	513	22	1455	29
8	26.7%	558	24	1455	36
9	30.0%	597	27	1455	34
10	33.3%	630	28	1455	35
11	36.7%	657	28	1455	37
12	40.0%	678	25	1455	36
13	43.3%	693	32	1455	131
14	46.7%	702	28	1455	31
15	50.0%	705	40	1455	35
16	53.3%	707	41	1455	34
17	56.7%	709	38	1455	39
18	60.0%	711	40	1455	35
19	63.3%	713	46	1455	48
20	66.7%	715	40	1455	36
21	70.0%	717	43	1455	39
22	73.3%	721	44	1455	36
23	76.7%	721	46	1455	37
24	80.0%	723	47	1455	37
25	83.3%	727	49	1455	38
26	86.7%	729	52	1455	35
27	90.0%	729	50	1455	39
28	93.3%	733	51	1455	40
29	96.7%	735	56	1455	35

Conclusion	The transition from adaptive to batch mode processing occurs, later than the expected 50%, at 57% change, which may well be due to the network topology.

Increasing number of cost functions

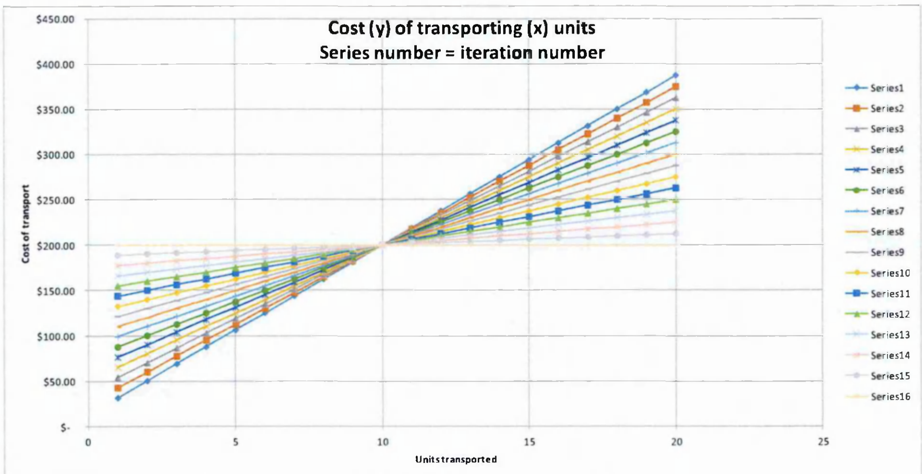
Case ID	Different transport cost models (TCM)
Category	Increasing problem complexity
Description	Under a transport cost model, which in iterations is changing from Pallet price to Full Truck Load price, it is expected that the transition will result in larger shipments with subsequent higher vehicle utilization, of the simple reason that the additional pallet in a FTL is already paid for, whilst the additional pallet in the pallet-price model comes at an individual cost.

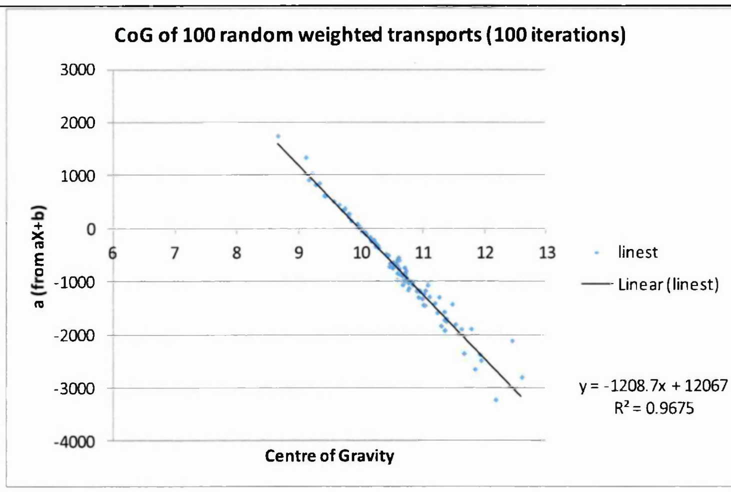
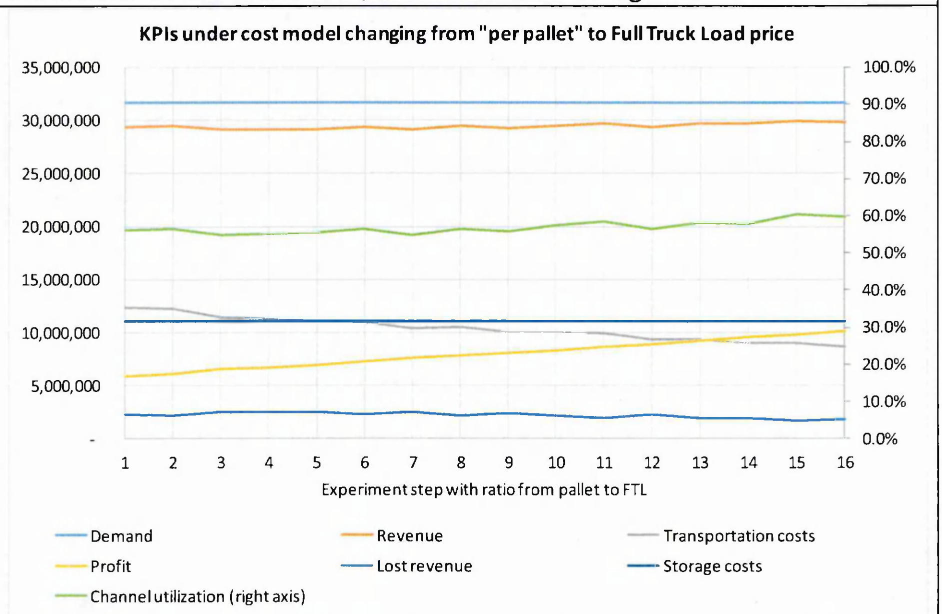
	LinearCostFunction(y0,y1,x1)		
iteration	Y0	Y1	X1
1	$ 12.50	$ 387.50	20
2	$ 25.00	$ 375.00	20
3	$ 37.50	$ 362.50	20
4	$ 50.00	$ 350.00	20
5	$ 62.50	$ 337.50	20
6	$ 75.00	$ 325.00	20
7	$ 87.50	$ 312.50	20
8	$ 100.00	$ 300.00	20
9	$ 112.50	$ 287.50	20
10	$ 125.00	$ 275.00	20
11	$ 137.50	$ 262.50	20
12	$ 150.00	$ 250.00	20
13	$ 162.50	$ 237.50	20
14	$ 175.00	$ 225.00	20
15	$ 187.50	$ 212.50	20
16	$ 200.00	$ 200.00	20

Transport costs @ N units

iteration	1	2	3	4	5	6	7	8	9	10
1	$ 31.25	$ 50.00	$ 68.75	$ 87.50	$ 106.25	$ 125.00	$ 143.75	$ 162.50	$ 181.25	$ 200.00
2	$ 42.50	$ 60.00	$ 77.50	$ 95.00	$ 112.50	$ 130.00	$ 147.50	$ 165.00	$ 182.50	$ 200.00
3	$ 53.75	$ 70.00	$ 86.25	$ 102.50	$ 118.75	$ 135.00	$ 151.25	$ 167.50	$ 183.75	$ 200.00
4	$ 65.00	$ 80.00	$ 95.00	$ 110.00	$ 125.00	$ 140.00	$ 155.00	$ 170.00	$ 185.00	$ 200.00
5	$ 76.25	$ 90.00	$ 103.75	$ 117.50	$ 131.25	$ 145.00	$ 158.75	$ 172.50	$ 186.25	$ 200.00
6	$ 87.50	$ 100.00	$ 112.50	$ 125.00	$ 137.50	$ 150.00	$ 162.50	$ 175.00	$ 187.50	$ 200.00
7	$ 98.75	$ 110.00	$ 121.25	$ 132.50	$ 143.75	$ 155.00	$ 166.25	$ 177.50	$ 188.75	$ 200.00
8	$ 110.00	$ 120.00	$ 130.00	$ 140.00	$ 150.00	$ 160.00	$ 170.00	$ 180.00	$ 190.00	$ 200.00
9	$ 121.25	$ 130.00	$ 138.75	$ 147.50	$ 156.25	$ 165.00	$ 173.75	$ 182.50	$ 191.25	$ 200.00
10	$ 132.50	$ 140.00	$ 147.50	$ 155.00	$ 162.50	$ 170.00	$ 177.50	$ 185.00	$ 192.50	$ 200.00
11	$ 143.75	$ 150.00	$ 156.25	$ 162.50	$ 168.75	$ 175.00	$ 181.25	$ 187.50	$ 193.75	$ 200.00
12	$ 155.00	$ 160.00	$ 165.00	$ 170.00	$ 175.00	$ 180.00	$ 185.00	$ 190.00	$ 195.00	$ 200.00
13	$ 166.25	$ 170.00	$ 173.75	$ 177.50	$ 181.25	$ 185.00	$ 188.75	$ 192.50	$ 196.25	$ 200.00
14	$ 177.50	$ 180.00	$ 182.50	$ 185.00	$ 187.50	$ 190.00	$ 192.50	$ 195.00	$ 197.50	$ 200.00
15	$ 188.75	$ 190.00	$ 191.25	$ 192.50	$ 193.75	$ 195.00	$ 196.25	$ 197.50	$ 198.75	$ 200.00
16	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00

iteration	11	12	13	14	15	16	17	18	19	20
1	$ 218.75	$ 237.50	$ 256.25	$ 275.00	$ 293.75	$ 312.50	$ 331.25	$ 350.00	$ 368.75	$ 387.50
2	$ 217.50	$ 235.00	$ 252.50	$ 270.00	$ 287.50	$ 305.00	$ 322.50	$ 340.00	$ 357.50	$ 375.00
3	$ 216.25	$ 232.50	$ 248.75	$ 265.00	$ 281.25	$ 297.50	$ 313.75	$ 330.00	$ 346.25	$ 362.50
4	$ 215.00	$ 230.00	$ 245.00	$ 260.00	$ 275.00	$ 290.00	$ 305.00	$ 320.00	$ 335.00	$ 350.00
5	$ 213.75	$ 227.50	$ 241.25	$ 255.00	$ 268.75	$ 282.50	$ 296.25	$ 310.00	$ 323.75	$ 337.50
6	$ 212.50	$ 225.00	$ 237.50	$ 250.00	$ 262.50	$ 275.00	$ 287.50	$ 300.00	$ 312.50	$ 325.00
7	$ 211.25	$ 222.50	$ 233.75	$ 245.00	$ 256.25	$ 267.50	$ 278.75	$ 290.00	$ 301.25	$ 312.50
8	$ 210.00	$ 220.00	$ 230.00	$ 240.00	$ 250.00	$ 260.00	$ 270.00	$ 280.00	$ 290.00	$ 300.00
9	$ 208.75	$ 217.50	$ 226.25	$ 235.00	$ 243.75	$ 252.50	$ 261.25	$ 270.00	$ 278.75	$ 287.50
10	$ 207.50	$ 215.00	$ 222.50	$ 230.00	$ 237.50	$ 245.00	$ 252.50	$ 260.00	$ 267.50	$ 275.00
11	$ 206.25	$ 212.50	$ 218.75	$ 225.00	$ 231.25	$ 237.50	$ 243.75	$ 250.00	$ 256.25	$ 262.50
12	$ 205.00	$ 210.00	$ 215.00	$ 220.00	$ 225.00	$ 230.00	$ 235.00	$ 240.00	$ 245.00	$ 250.00
13	$ 203.75	$ 207.50	$ 211.25	$ 215.00	$ 218.75	$ 222.50	$ 226.25	$ 230.00	$ 233.75	$ 237.50
14	$ 202.50	$ 205.00	$ 207.50	$ 210.00	$ 212.50	$ 215.00	$ 217.50	$ 220.00	$ 222.50	$ 225.00
15	$ 201.25	$ 202.50	$ 203.75	$ 205.00	$ 206.25	$ 207.50	$ 208.75	$ 210.00	$ 211.25	$ 212.50
16	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00	$ 200.00



	
Network	Circular 1-4-16
Assertions	Assert that: The variation in the schedulers' usage of the number of units should be consistent over the tests. Series close to iteration 1, the deliveries should be frequent and small. At iterations close to 16 the deliveries should be few and many. Differences at less than 1000 $ are not considered significant.
Results	

Results Data

step from Pallet to FTL price	Demand	Revenue	Transportation costs	Profit	Lost revenue	Channel utilization (right axis)	Storage costs	
1	31,680,000	29,388,800	12,383,570	5,880,230	2,291,200	56.1%	11,125,000	Pallet price
2	31,680,000	29,462,800	12,219,429	6,118,371	2,217,200	56.6%	11,125,000	
3	31,680,000	29,128,400	11,459,610	6,543,790	2,551,600	54.8%	11,125,000	
4	31,682,400	29,159,600	11,290,990	6,743,610	2,522,800	55.1%	11,125,000	
5	31,680,000	29,126,000	11,037,018	6,963,982	2,554,000	55.6%	11,125,000	
6	31,680,000	29,442,800	11,026,406	7,291,394	2,237,200	56.4%	11,125,000	
7	31,680,000	29,180,400	10,425,980	7,629,420	2,499,600	54.8%	11,125,000	
8	31,680,000	29,460,000	10,519,480	7,815,520	2,220,000	56.4%	11,125,000	
9	31,680,000	29,299,200	10,079,257	8,094,943	2,380,800	55.7%	11,125,000	
10	31,680,000	29,485,600	10,006,355	8,354,245	2,194,400	57.5%	11,125,000	
11	31,680,000	29,764,000	9,943,586	8,695,414	1,916,000	58.4%	11,125,000	
12	31,680,000	29,387,600	9,395,203	8,867,397	2,292,400	56.7%	11,125,000	
13	31,680,000	29,742,400	9,332,561	9,284,839	1,937,600	58.2%	11,125,000	
14	31,680,000	29,702,000	8,984,486	9,592,514	1,978,000	57.8%	11,125,000	
15	31,680,000	29,943,600	9,035,116	9,783,484	1,736,400	60.6%	11,125,000	
16	31,680,000	29,891,600	8,649,331	10,117,269	1,788,400	59.9%	11,125,000	FTL Price

Conclusion: The assertions hold true.

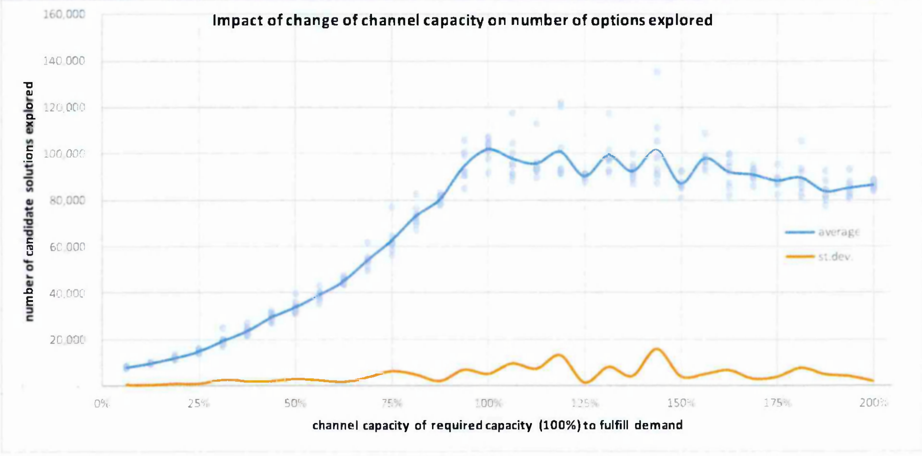
	Under constant high profitable demand, the transport costs are gradually reduced as the channel utilization increases with the usage of FTL price model.

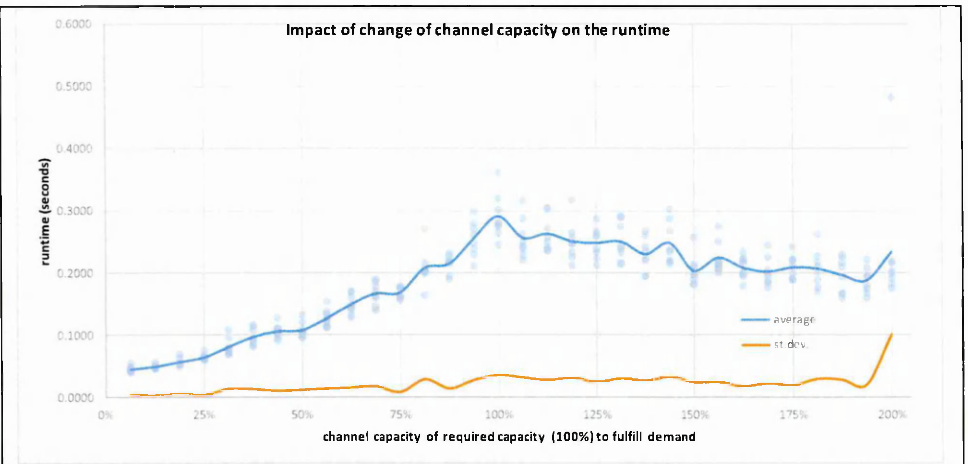
Increasing channel capacity

"(Performance, capacity)" shows three cases, where the capacity changes from 0 to 200% of the capacity required to fulfil all demand.

The cases differ as follows:

- ***PC*** is constrained on the capacity between **storage** and **outlet**
- ***PC2*** is constrained on the capacity between **factory** and **storage**
- ***PC3*** is identical to PC2, but runs in batch-mode where all other runs in adaptive mode. PC3 therefore compares to PC2 in absolute numbers.
- ***PC4*** has **two storage layers (1-4-4-16)** which are connected 1:1. Hereby the complexity of choices increases quadratic.

Case ID	Change on downstream channel - adaptive mode (PC)
Category	Increasing problem complexity
Description	In a network with profitable highly priced products, the capacity of the channels from storage to outlet is changed in iterations from 0 to 200% of what is required to fulfil all demand.
Network	1-4-16
Assertions	Assert that signal-to-noise-ratio (SNR) is constant as complexity is relax (by having more channel capacity)
Results	



Results Data

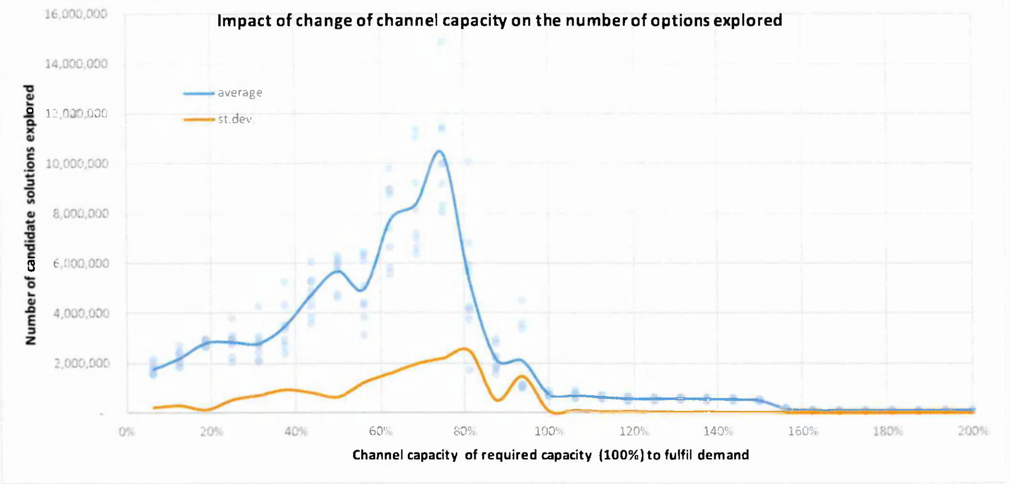
required channel capacity	number of options average	number of options st.dev.	Experiment No. 1	2	3	4	5	6	7	8
200%	86,255	1,782	87,841	83,780	85,897	84,331	85,424	88,614	88,112	86,044
194%	85,071	4,019	84,847	81,524	86,294	80,532	87,706	83,654	82,901	93,112
188%	83,608	4,772	81,606	84,159	92,324	79,924	81,254	77,355	88,154	84,091
181%	89,524	7,620	80,982	105,218	85,093	83,201	88,730	93,471	92,121	87,374
175%	88,201	3,649	95,262	88,309	82,145	87,346	89,153	87,328	86,581	89,481
169%	90,804	2,840	94,932	85,675	90,316	92,982	92,518	88,915	91,406	89,684
163%	91,965	6,602	94,172	87,208	81,833	92,083	95,945	99,355	99,764	85,356
156%	97,833	4,863	98,400	96,588	108,674	96,528	92,512	98,955	94,084	96,922
150%	86,918	3,868	80,532	92,462	86,641	85,917	85,434	87,305	85,020	92,032
144%	101,322	15,901	91,236	86,931	135,071	111,092	89,961	98,121	92,866	105,296
138%	92,132	4,001	94,991	92,080	88,811	86,932	89,777	99,677	93,658	91,128
131%	99,116	8,019	98,033	101,301	117,305	91,321	97,376	98,234	92,508	96,852
125%	89,999	1,112	90,464	89,513	90,989	90,364	87,630	90,845	89,484	90,703
119%	100,581	13,139	91,653	120,200	102,373	121,865	92,799	91,384	91,282	93,093
113%	95,488	7,192	93,156	89,459	95,916	92,324	93,994	92,665	93,678	112,711
106%	97,714	9,571	117,449	90,254	95,653	104,372	91,253	99,973	88,185	94,575
100%	101,639	4,999	101,874	107,106	103,686	101,018	91,692	97,620	104,076	106,037
94%	94,380	6,801	90,470	91,945	100,340	105,846	99,188	91,959	84,828	90,461
88%	80,190	1,835	81,480	81,459	80,414	78,817	82,770	80,924	77,935	77,723
81%	72,956	4,675	66,806	82,332	70,659	73,318	69,585	72,567	76,097	72,284
75%	62,499	6,184	58,532	56,099	61,052	64,343	61,789	60,882	76,609	60,687
69%	54,071	3,593	49,231	55,181	50,892	61,493	53,920	53,426	54,055	54,366
63%	44,886	1,413	46,336	44,739	43,662	43,766	44,095	45,679	47,316	43,493
56%	39,052	2,171	38,476	39,543	37,326	42,993	39,772	35,503	38,869	39,930
50%	33,715	2,772	35,723	31,683	33,399	32,407	33,882	32,070	31,025	39,534
44%	29,415	1,871	30,441	26,917	29,064	30,599	28,121	27,224	32,080	30,871
38%	23,440	1,803	21,672	21,504	21,956	26,890	24,037	23,286	24,620	23,556
31%	19,308	2,375	19,760	18,011	24,678	16,902	19,609	17,872	19,225	18,408
25%	14,718	688	14,847	15,392	13,744	13,956	14,578	14,930	14,484	15,812
19%	11,941	680	13,294	12,071	11,196	11,597	11,932	11,556	12,475	11,406
13%	9,640	130	9,431	9,588	9,508	9,639	9,711	9,832	9,744	9,670
6%	7,950	261	8,294	8,076	8,070	7,897	7,519	8,070	8,057	7,614

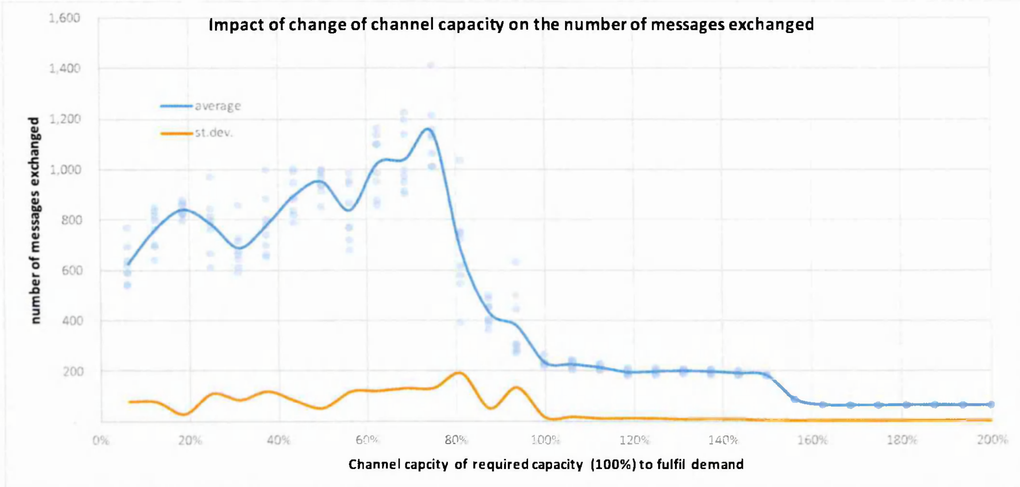
required channel capacity	runtime (seconds) average	runtime (seconds) st.dev.	Experiment No. 1	2	3	4	5	6	7	8
200%	0.2336	0.1016	0.4821	0.1754	0.1995	0.2025	0.2191	0.2168	0.1912	0.1822
194%	0.1878	0.0210	0.2211	0.1679	0.1747	0.1595	0.1801	0.1955	0.1955	0.2081
188%	0.1965	0.0286	0.2167	0.2266	0.1945	0.1600	0.1666	0.1670	0.2106	0.2299
181%	0.2066	0.0302	0.1883	0.2016	0.1689	0.2104	0.2271	0.2173	0.2628	0.1766
175%	0.2085	0.0194	0.2220	0.2137	0.2205	0.1887	0.1914	0.2004	0.1892	0.2426
169%	0.2018	0.0230	0.2448	0.1860	0.1951	0.1994	0.2043	0.1759	0.2252	0.1835
163%	0.2084	0.0180	0.2337	0.1792	0.1948	0.2163	0.2209	0.2227	0.1984	0.2009
156%	0.2245	0.0253	0.2148	0.2101	0.2505	0.2179	0.2005	0.2200	0.2070	0.2750
150%	0.2037	0.0247	0.1819	0.2091	0.1986	0.1891	0.1821	0.2577	0.1967	0.2141
144%	0.2479	0.0336	0.2343	0.2185	0.2873	0.3023	0.2156	0.2377	0.2201	0.2679
138%	0.2299	0.0271	0.2414	0.2607	0.1951	0.2241	0.2202	0.2343	0.1948	0.2682
131%	0.2502	0.0309	0.2329	0.2744	0.2426	0.2912	0.2146	0.2896	0.2395	0.2166
125%	0.2479	0.0249	0.2596	0.2612	0.2233	0.2342	0.2390	0.2670	0.2121	0.2867
119%	0.2505	0.0319	0.2471	0.2650	0.2336	0.3173	0.2116	0.2561	0.2257	0.2474
113%	0.2627	0.0283	0.3048	0.2357	0.2575	0.2370	0.2718	0.2410	0.2500	0.3038
106%	0.2559	0.0321	0.2627	0.2389	0.2205	0.3166	0.2426	0.2873	0.2287	0.2495
100%	0.2906	0.0362	0.2809	0.3199	0.2779	0.2746	0.2453	0.2641	0.3015	0.3607
94%	0.2550	0.0285	0.2297	0.2716	0.2390	0.2804	0.2482	0.2981	0.2111	0.2617
88%	0.2145	0.0148	0.2168	0.2182	0.2160	0.2320	0.1952	0.2286	0.2196	0.1899
81%	0.2077	0.0294	0.1644	0.2705	0.2013	0.2051	0.2037	0.2157	0.1988	0.2019
75%	0.1681	0.0090	0.1743	0.1626	0.1603	0.1787	0.1758	0.1604	0.1763	0.1563
69%	0.1666	0.0180	0.1593	0.1461	0.1401	0.1700	0.1628	0.1894	0.1868	0.1784
63%	0.1495	0.0159	0.1420	0.1706	0.1695	0.1604	0.1481	0.1330	0.1439	0.1282
56%	0.1267	0.0147	0.1146	0.1264	0.1378	0.1334	0.1545	0.1123	0.1130	0.1218
50%	0.1080	0.0126	0.1088	0.0958	0.1019	0.1205	0.1052	0.0969	0.1023	0.1326
44%	0.1061	0.0111	0.1278	0.0961	0.1073	0.1033	0.0912	0.1114	0.1099	0.1020
38%	0.0972	0.0137	0.1069	0.1143	0.0814	0.1166	0.0945	0.0840	0.0922	0.0878
31%	0.0811	0.0144	0.0746	0.0693	0.1093	0.0721	0.0804	0.0700	0.0967	0.0768
25%	0.0644	0.0046	0.0615	0.0632	0.0613	0.0748	0.0644	0.0638	0.0605	0.0657
19%	0.0570	0.0064	0.0631	0.0549	0.0500	0.0532	0.0538	0.0699	0.0558	0.0552
13%	0.0495	0.0035	0.0519	0.0480	0.0468	0.0466	0.0489	0.0498	0.0568	0.0472
6%	0.0453	0.0049	0.0488	0.0418	0.0467	0.0402	0.0427	0.0427	0.0440	0.0552

Conclusion	The assertion holds true. The number of candidate solutions level out (though with a slightly increase variation) at 100% of required capacity. The runtime increases up to the peak 100% capacity, which requires fully coordinated exploitation of the capacity. Beyond 100% the runtime decreases the constraints are relax without influence on the quality of the solution: *"Whether a parcel comes on truck A or B doesn't matter if you pay for full truck loads and they are profitable anyway."*

Case ID	Change on upstream channel - adaptive mode (PC2)
Category	Increasing problem complexity
Description	- The capacity of the factory channels changes in the same way as in the previous study (PC). The main difference is that this situation is more difficult for the engine because demand is exclusively driven by the shops, whereby the storage first adapt to demand, and then needs to incorporate the constraints of supply from the factory.
Network	1-4-16
Assertions	The assertion is that the number of impossible/redundant options should increase significantly as the bottleneck moves away (upstream) from the demand signal.

Results





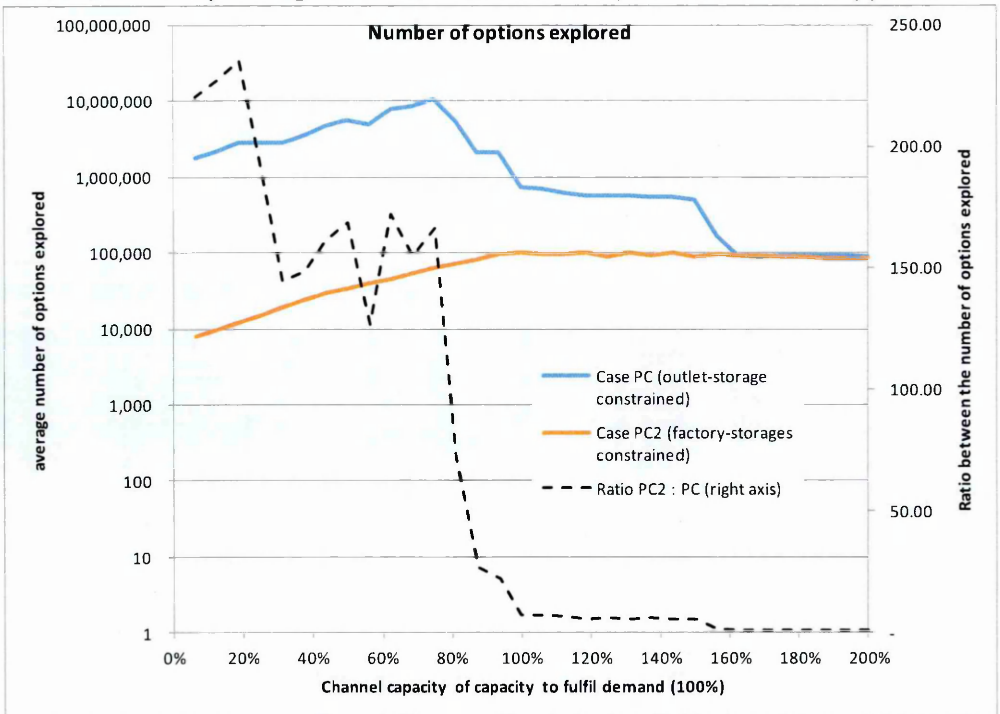
Results Data

required channel capacity	number of options		Experiment No.							
	average	st.dev.	1	2	3	4	5	6	7	8
200%	88,146	2,131	85,014	89,641	89,498	89,970	85,885	87,011	87,275	90,871
194%	90,909	6,576	99,006	102,630	88,170	87,632	89,774	87,892	89,590	82,577
188%	91,155	4,109	90,610	89,016	96,715	89,645	83,414	91,881	95,343	92,618
181%	92,040	4,759	87,162	100,088	94,105	90,950	91,600	88,954	86,507	96,957
175%	92,568	4,552	98,738	99,752	93,160	92,597	87,372	89,024	90,899	89,005
169%	89,151	2,433	91,995	87,458	88,574	91,014	92,612	87,054	86,011	88,492
163%	92,606	8,126	84,970	106,930	97,897	97,372	90,268	90,817	91,761	80,832
156%	163,717	7,914	178,012	174,498	161,086	161,840	157,897	158,015	159,022	159,365
150%	496,069	18,112	494,270	483,844	470,335	492,616	513,262	509,772	523,562	480,894
144%	538,114	26,913	501,315	523,903	510,708	544,602	537,915	579,713	537,191	569,568
138%	551,384	35,941	531,616	537,031	484,746	598,614	549,853	575,800	547,919	585,495
131%	575,756	31,798	540,601	595,717	586,979	518,867	586,485	587,149	571,827	618,426
125%	561,820	39,295	571,742	496,220	623,574	572,190	598,906	533,808	548,883	549,238
119%	567,740	61,856	520,567	650,487	541,982	609,534	482,096	643,111	573,819	520,326
113%	628,694	45,041	581,034	618,162	619,539	636,559	570,420	686,088	698,629	619,118
106%	710,451	108,095	694,585	628,980	716,499	603,384	921,011	743,436	779,937	595,777
100%	737,392	69,672	764,483	658,762	691,435	738,925	705,357	890,977	726,316	722,878
94%	2,091,028	1,468,880	3,356,668	1,103,654	1,122,020	3,587,969	1,019,713	1,058,969	965,846	4,513,384
88%	2,146,769	505,660	1,552,883	1,848,711	2,265,380	1,845,560	2,799,705	2,948,438	1,730,588	2,182,890
81%	5,287,538	2,485,436	1,713,186	5,638,793	3,762,014	6,817,099	10,064,678	5,934,710	4,230,290	4,139,535
75%	10,405,942	2,195,795	9,165,994	9,988,827	10,007,970	14,871,483	8,053,296	11,483,564	8,321,819	11,354,586
69%	8,388,068	1,976,882	9,237,267	6,376,919	6,989,769	6,637,206	11,391,174	8,179,138	11,064,474	7,228,600
63%	7,734,714	1,589,115	5,588,889	8,925,236	6,637,987	7,375,232	8,738,106	5,847,380	8,979,802	9,785,080
56%	4,960,509	1,237,354	3,126,190	4,420,925	5,085,881	6,321,649	4,324,180	6,110,218	3,844,379	6,450,652
50%	5,681,953	642,902	6,308,886	4,741,959	4,620,480	6,142,690	5,934,000	5,713,850	6,082,667	5,911,091
44%	4,739,878	810,318	4,897,118	3,862,704	3,598,750	6,045,545	5,295,661	4,646,385	4,299,997	5,272,867
38%	3,488,220	933,985	4,347,439	2,923,302	3,348,179	5,256,881	2,668,260	3,469,275	3,512,296	2,380,130
31%	2,787,180	709,516	2,092,580	3,063,565	2,051,766	2,714,913	2,875,732	4,298,734	2,432,264	2,767,889
25%	2,839,860	539,088	2,877,934	2,036,422	2,235,549	2,950,035	2,963,963	3,800,939	3,093,982	2,760,057
19%	2,809,433	134,492	2,789,128	2,901,047	2,873,752	2,668,761	2,661,815	2,653,236	2,963,630	2,964,094
13%	2,192,816	304,687	2,702,929	2,199,732	2,039,053	2,443,467	1,821,416	1,866,133	2,069,209	2,400,585
6%	1,752,873	223,794	2,147,383	1,806,703	1,521,842	1,596,147	1,783,890	1,992,323	1,564,811	1,609,882

required channel capacity	number of messages average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
200%	66	2	63	68	64	67	64	66	66	66
194%	66	2	68	68	65	65	67	63	65	65
188%	66	1	66	64	68	66	65	67	68	67
181%	67	1	66	69	66	67	66	66	65	67
175%	66	1	67	66	65	66	64	66	66	65
169%	65	1	66	65	65	67	66	64	63	67
163%	67	1	67	68	67	67	66	65	68	66
156%	88	1	87	90	89	89	89	88	89	86
150%	184	2	183	181	181	183	187	186	186	182
144%	193	7	184	191	184	194	190	201	192	204
138%	199	9	192	193	183	207	201	208	199	205
131%	202	7	195	204	206	189	203	206	201	210
125%	199	10	202	185	216	198	209	188	200	195
119%	196	11	188	200	191	206	180	215	194	192
113%	215	10	206	213	207	218	204	228	230	212
106%	227	16	225	215	232	210	248	240	242	205
100%	234	14	237	219	227	232	228	266	232	231
94%	379	133	502	308	304	445	279	289	271	632
88%	432	49	364	404	455	401	501	488	389	454
81%	676	190	393	749	581	753	1,037	726	547	618
75%	1,143	131	1,064	1,129	1,139	1,414	1,013	1,160	1,011	1,215
69%	1,038	130	1,140	903	950	915	1,227	997	1,198	973
63%	1,023	119	860	1,138	953	987	1,100	878	1,101	1,165
56%	836	117	679	769	867	955	772	943	721	985
50%	950	51	988	852	915	1,001	938	939	999	971
44%	896	80	891	820	789	1,003	949	882	838	993
38%	778	118	885	700	742	999	654	780	803	662
31%	688	84	613	729	593	666	678	859	645	717
25%	782	109	783	612	668	846	814	971	800	761
19%	840	27	859	867	828	822	798	827	838	880
13%	764	76	850	801	703	819	696	642	764	835
6%	625	77	770	623	546	539	639	696	594	591

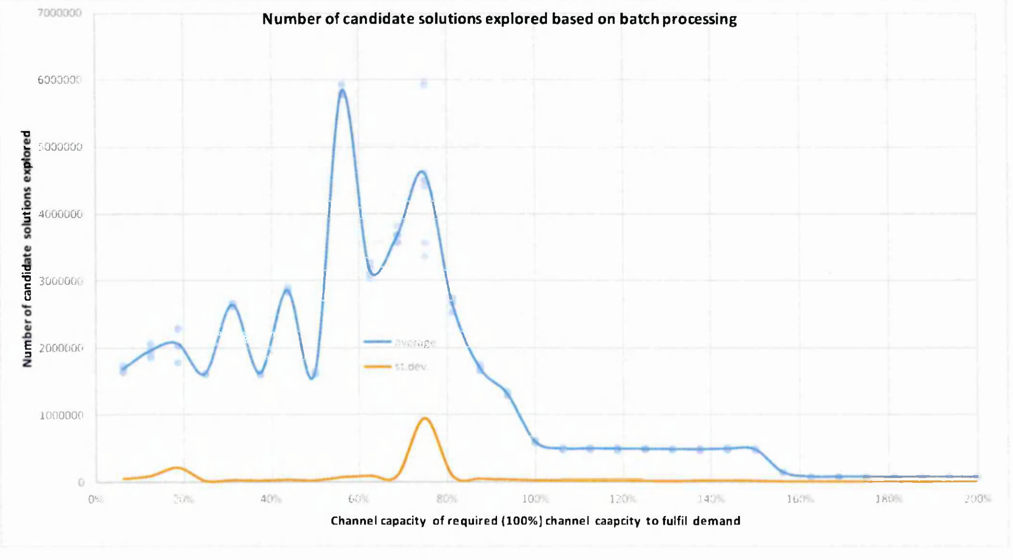
Conclusion

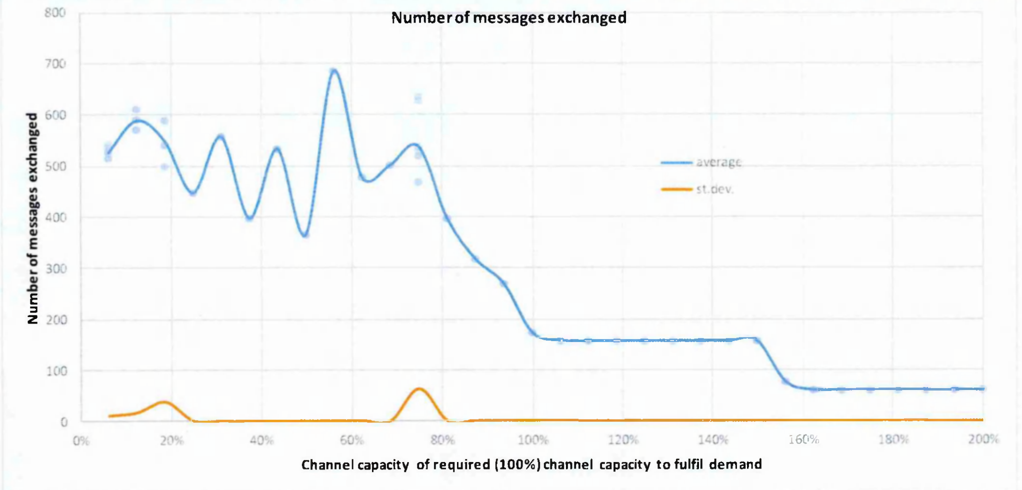
The assertion holds true. The number of options explored increase by 1-2 orders of magnitude as the bottleneck moves upstream. However as capacity is increased and the constraint stops being a bottleneck (at C > 100%) the difference disappears.



This observation is well aligned with the scalability tests on 10k & 20k SKUs where it is computationally more efficient to generate options by following the waves of demand upstream in waves. *A strategy for handling this can be put in place in the industrial version. (PC3)*

PC2 - Constrained on channel from Factory to Storages			PC - Constrained on channel from Storage to Outlets			Ratio PC2:PC	
required channel capacity	number of options		required channel capacity	number of options			
	average	st.dev.		average	st.dev.	average	st.dev
200%	88,146	2,131	200%	86,255	1,782	1.02	0.836107
194%	90,909	6,576	194%	85,071	4,019	1.07	0.611079
188%	91,155	4,109	188%	83,608	4,772	1.09	1.16127
181%	92,040	4,759	181%	89,524	7,620	1.03	1.601246
175%	92,568	4,552	175%	88,201	3,649	1.05	0.801561
169%	89,151	2,433	169%	90,804	2,840	0.98	1.167324
163%	92,606	8,126	163%	91,965	6,602	1.01	0.812501
156%	163,717	7,914	156%	97,833	4,863	1.67	0.614533
150%	496,069	18,112	150%	86,918	3,868	5.71	0.21353
144%	538,114	26,913	144%	101,322	15,901	5.31	0.590838
138%	551,384	35,941	138%	92,132	4,001	5.98	0.111322
131%	575,756	31,798	131%	99,116	8,019	5.81	0.252191
125%	561,820	39,295	125%	89,999	1,112	6.24	0.028297
119%	567,740	61,856	119%	100,581	13,139	5.64	0.21242
113%	628,694	45,041	113%	95,488	7,192	6.58	0.159682
106%	710,451	108,095	106%	97,714	9,571	7.27	0.088541
100%	737,392	69,672	100%	101,639	4,999	7.26	0.071744
94%	2,091,028	1,468,880	94%	94,380	6,801	22.16	0.00463
88%	2,146,769	505,660	88%	80,190	1,835	26.77	0.003629
81%	5,287,538	2,485,436	81%	72,956	4,675	72.48	0.001881
75%	10,405,942	2,195,795	75%	62,499	6,184	166.50	0.002816
69%	8,388,068	1,976,882	69%	54,071	3,593	155.13	0.001818
63%	7,734,714	1,589,115	63%	44,886	1,413	172.32	0.000889
56%	4,960,509	1,237,354	56%	39,052	2,171	127.02	0.001755
50%	5,681,953	642,902	50%	33,715	2,772	168.53	0.004312
44%	4,739,878	810,318	44%	29,415	1,871	161.14	0.002309
38%	3,488,220	933,985	38%	23,440	1,803	148.81	0.001931
31%	2,787,180	709,516	31%	19,308	2,375	144.35	0.003347
25%	2,839,860	539,088	25%	14,718	688	192.95	0.001277
19%	2,809,433	134,492	19%	11,941	680	235.28	0.005059
13%	2,192,816	304,687	13%	9,640	130	227.46	0.000425
6%	1,752,873	223,794	6%	7,950	261	220.50	0.001165

Case ID	Change on upstream channel - batch mode (PC3)
Category	Increasing problem complexity
Description	This test is the same as the previous with one modification: A new option that follows the waves of computations (learning from PC2) so that new messages are not processed until the wave of negotiations are completed at each upstream tier. This is only expected to work for tree-like graphs and will not work well for real-time usage, however it provides a faster mode of batch scheduling this class of networks.
Network	1-4-16
Assertions	Assert that usage of "batch"-mode provides more efficient message parsing mechanism than interactive.
Results	



Observation: The core utilization is less than 100% as cores start waiting for a wave of negotiations to complete.

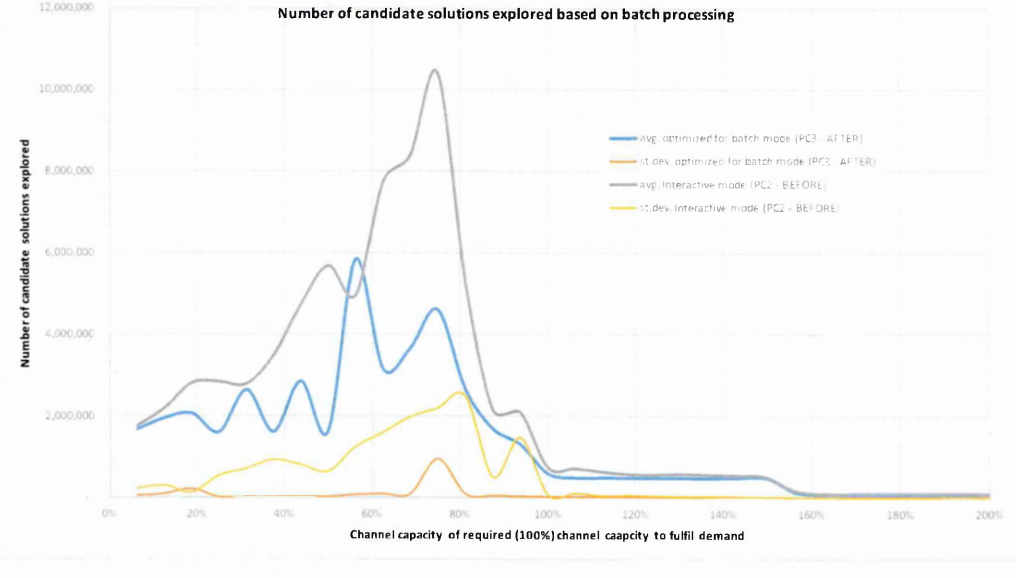
Results Data

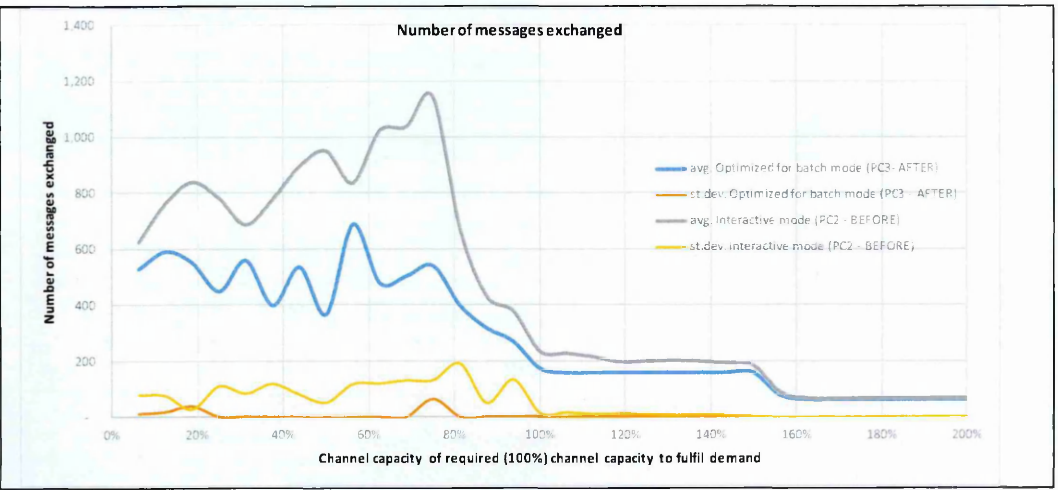
required channel	number of options		Experiment No.							
capacity	average	st.dev.	1	2	3	4	5	6	7	8
200%	80,414	2,185	80136	77934	82344	82344	82344	77934	82344	77934
194%	81,239	2,282	82893	78483	78483	82893	82893	78483	82893	82893
188%	83,040	-	83040	83040	83040	83040	83040	83040	83040	83040
181%	80,120	2,646	83169	78759	83169	78759	76416	78759	83169	78759
175%	82,550	1,559	83101	83101	83101	78691	83101	83101	83101	83101
169%	81,237	2,357	79032	83442	83442	79032	79032	83442	79032	83442
163%	82,632	2,041	83734	79324	83734	83734	83734	83734	83734	79324
156%	146,644	2,549	143089	147836	147823	143049	147863	147823	150223	145449
150%	487,045	11,612	470791	486384	502388	472817	484835	501721	491495	485926
144%	496,766	16,948	505847	493693	480520	501759	479109	478828	526194	508181
138%	488,628	14,158	493968	480576	502046	484904	497182	474176	508497	467674
131%	491,730	8,784	484851	496617	483043	498954	482174	506535	487017	494646
125%	494,683	11,797	500101	485829	513430	502308	485704	482961	504435	482695
119%	496,502	17,593	493092	503957	466347	475356	517250	510998	506391	498625
113%	501,704	12,685	508031	512328	494301	513614	490180	481050	498122	516009
106%	498,500	19,201	493623	508319	500595	502961	524377	467306	515091	475731
100%	604,701	18,375	608923	602449	600748	591681	631874	626083	573836	602017
94%	1,317,679	30,753	1337358	1297113	1289077	1315040	1278183	1351630	1308566	1364468
88%	1,700,918	41,721	1667148	1768512	1674472	1721564	1656769	1664643	1708248	1745984
81%	2,669,541	86,722	2526322	2752475	2553421	2723019	2692171	2690147	2660403	2758373
75%	4,613,134	949,893	5925393	4508999	3571685	4416919	4620767	4501952	3368417	5990942
69%	3,661,557	85,646	3578872	3700234	3687180	3674222	3576829	3825944	3675878	3573294
63%	3,169,723	88,534	3222256	3131578	3128889	3279704	3089482	3191578	3035376	3278921
56%	5,835,561	66,611	5801919	5779808	5952198	5807431	5927831	5786584	5831856	5796863
50%	1,626,799	16,931	1614477	1642981	1621648	1621932	1608362	1621467	1660823	1622699
44%	2,862,579	27,549	2851418	2889661	2915393	2854036	2867796	2831962	2848539	2841825
38%	1,620,269	15,412	1633158	1634218	1615875	1633434	1604366	1616634	1631127	1593338
31%	2,647,514	22,839	2610480	2678132	2645449	2660473	2675562	2635241	2641102	2633670
25%	1,615,069	8,919	1626415	1612550	1623885	1614057	1618461	1597228	1611655	1616298
19%	2,070,230	212,758	2310024	2030627	2030869	1794366	2276174	2297568	1778054	2044155
13%	1,959,250	85,564	1961785	1970186	1990117	2082135	2059343	1897211	1864738	1848485
6%	1,683,138	43,815	1674877	1635869	1635116	1647680	1731391	1697815	1688511	1753845

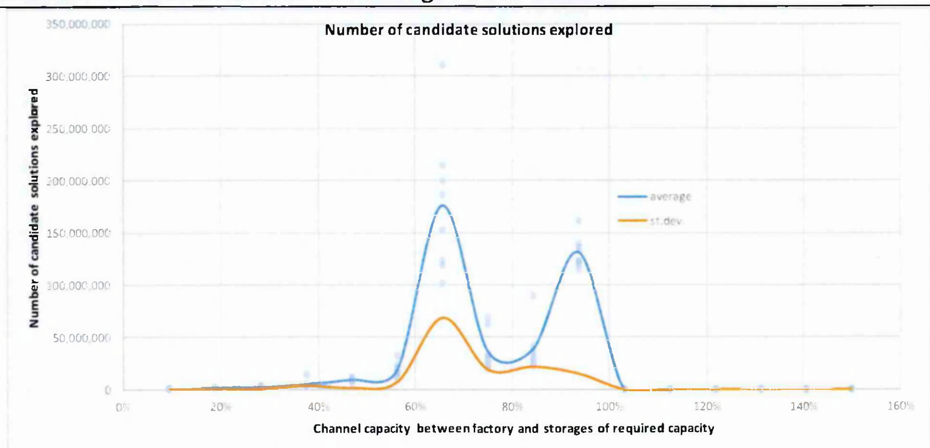
required channel capacity	number of messages average	number of messages st.dev.	Experiment No. 1	2	3	4	5	6	7	8
200%	62	-	62	62	62	62	62	62	62	62
194%	62	-	62	62	62	62	62	62	62	62
188%	62	-	62	62	62	62	62	62	62	62
181%	62	-	62	62	62	62	62	62	62	62
175%	62	-	62	62	62	62	62	62	62	62
169%	62	-	62	62	62	62	62	62	62	62
163%	62	-	62	62	62	62	62	62	62	62
156%	78	-	78	78	78	78	78	78	78	78
150%	158	-	158	158	158	158	158	158	158	158
144%	158	-	158	158	158	158	158	158	158	158
138%	158	-	158	158	158	158	158	158	158	158
131%	158	-	158	158	158	158	158	158	158	158
125%	158	-	158	158	158	158	158	158	158	158
119%	158	-	158	158	158	158	158	158	158	158
113%	158	-	158	158	158	158	158	158	158	158
106%	158	-	158	158	158	158	158	158	158	158
100%	174	-	174	174	174	174	174	174	174	174
94%	270	-	270	270	270	270	270	270	270	270
88%	318	-	318	318	318	318	318	318	318	318
81%	398	-	398	398	398	398	398	398	398	398
75%	539	63	628	530	470	520	538	520	468	636
69%	502	-	502	502	502	502	502	502	502	502
63%	478	-	478	478	478	478	478	478	478	478
56%	686	-	686	686	686	686	686	686	686	686
50%	366	-	366	366	366	366	366	366	366	366
44%	534	-	534	534	534	534	534	534	534	534
38%	398	-	398	398	398	398	398	398	398	398
31%	558	-	558	558	558	558	558	558	558	558
25%	448	-	448	448	448	448	448	448	448	448
19%	550	38	590	542	542	500	590	590	500	542
13%	590	17	592	592	592	612	612	572	572	572
6%	527	11	526	516	516	516	544	530	530	538

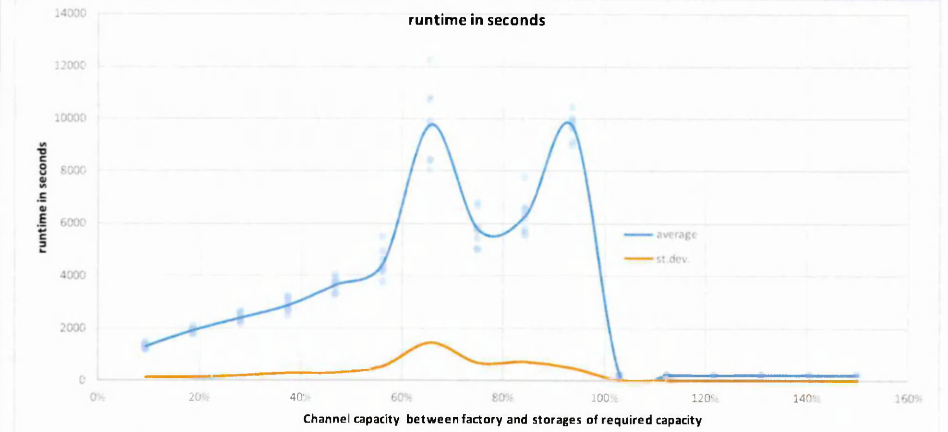
Conclusion

The assertion holds True. See the 2 charts below: Both the number of candidate solutions and number of messages are significantly reduced AFTER the message parsing is optimised further.





Case ID	Change on upstream channel in multi-tier network (PC4)
Category	Increasing problem complexity
Description	The assertion is that limiting the number of options (learning from PC3) has very limited effect when the network is multi-tiered. The network has 4 Tiers: In Tier 1 there are 16 outlets which each are connected to 2 storages in Tier 2. To fulfil demand the outlets have to utilize both channels. Each of the Tier 2 storages are connected to 1 storages each in the 3rd Tier on a 1:1 basis. Finally all of the Tier 3 storages are connected to the 1 source (Tier 4). Because of the two storage Tiers (2, and 3) the complexity of choices increases quadratic. All products are profitable.
Network	1-4-4-16
Assertions	Assert whether change of capacity of channels in Tier 1-2 (sources to storages) result in reduced number of messages and runtime
Results	



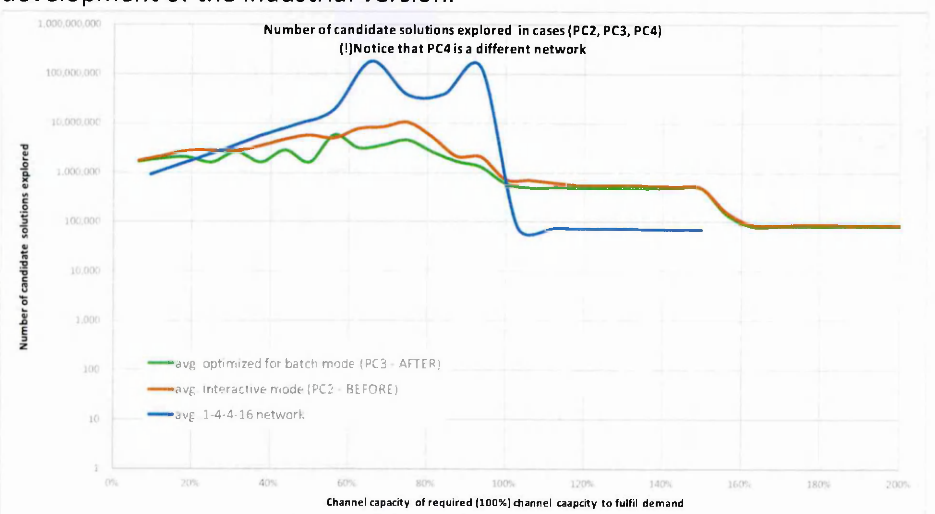
Results Data

required channel capacity	number of options average	number of options st.dev.	Experiment No. 1	2	3	4	5	6	7	8
9%	893,317	159,176	1133724	699700	1100460	786282	973431	850701	836303	765931
19%	1,605,462	195,560	1614703	1960272	1757114	1461985	1649207	1606872	1464319	1329221
28%	2,846,459	388,226	2654216	3413284	2536035	2690628	2274338	3122153	3261178	2819942
38%	5,463,332	3,610,853	5181108	14165071	2729129	4409984	4597603	5061087	3385743	4176928
47%	9,313,514	1,491,107	11748246	8157317	10768915	7243275	9046171	8847328	8452948	10243909
56%	18,440,658	7,216,242	21166781	16939569	15718408	32353192	16846982	23441802	9776516	11282014
66%	176,009,671	68,179,144	123250829	118966253	214876731	152350819	186438496	101467987	310913860	199812391
75%	36,487,830	18,689,515	68744450	24400681	27968026	62709064	31909168	22810991	33678512	19681746
84%	38,443,862	21,620,826	22299442	37119060	89465598	34637640	28017045	30305330	24059712	41647065
94%	131,285,350	14,958,723	123302174	132671274	115294159	119568325	121590560	139901432	161496208	136458668
103%	77,116	1,954	78475	73379	75632	77061	76446	78023	78379	79535
113%	73,266	2,961	78079	71816	70533	71827	75749	76155	70081	71885
122%	71,198	1,623	70803	69275	72409	73844	72173	69228	70173	71679
131%	71,886	1,260	70382	73213	72164	73194	71317	72452	72537	69828
141%	69,155	2,064	65963	71800	68617	71733	70487	67347	68907	68386
150%	69,014	2,236	72921	68772	67990	68332	71341	68387	68869	65501

required channel capacity	time in seconds average	time in seconds st.dev.	Experiment No. 1	2	3	4	5	6	7	8
9%	1,296	98	1290	1165	1454	1314	1358	1370	1226	1194
19%	1,914	105	1793	2096	1950	1749	1923	1942	1941	1914
28%	2,389	167	2302	2630	2342	2238	2362	2510	2571	2158
38%	2,873	264	2800	3172	2457	2924	3041	3220	2671	2702
47%	3,651	273	3703	3254	4046	3528	3744	3738	3305	3887
56%	4,481	532	4614	4314	4153	5489	4362	4946	3771	4198
66%	9,795	1,450	8428	8448	10825	9918	9717	8049	12255	10716
75%	5,793	667	6804	5438	5782	6674	5876	5005	5714	5050
84%	6,305	709	5786	6529	7755	6331	5657	6626	5555	6203
94%	9,730	477	9684	10039	8994	9140	9666	9923	10450	9940
103%	217	4	220	211	214	214	215	222	216	222
113%	215	5	224	217	212	214	219	216	211	208
122%	211	3	214	210	207	213	214	210	209	213
131%	215	4	211	216	218	216	214	214	222	212
141%	215	4	212	221	212	218	219	210	210	214
150%	218	4	224	220	218	214	218	218	216	212

Conclusion

The assertion is undetermined. The number of candidate solutions explored in this (more complex) network, exceed the previous by two orders of magnitude (see chart below). This case should be subject for further research during the development of the industrial version.

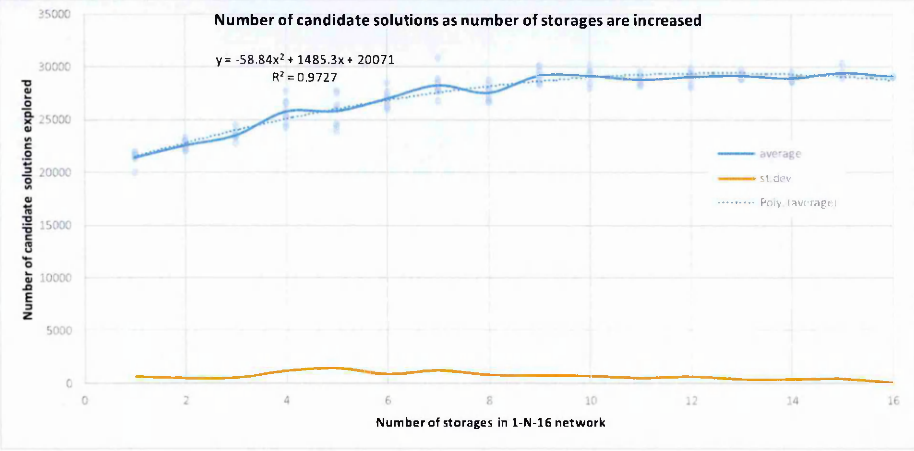


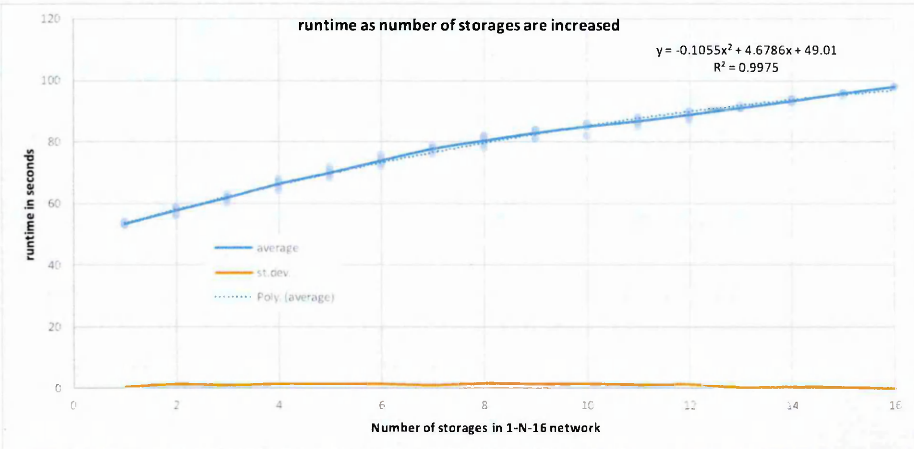
	Notice the absolute number of options evaluated under competing constraints. This difference is produced by the increase of complexity of the network topology.

Increasing number of routes with parallel storages

Case ID	Excess channel capacity (PPar)
Category	Increasing problem complexity
Description	This takes an existing network and schedules in batch mode. In each step the number of storages (1-N-16) between a factory and 16 retail outlets are increased. The retail outlets are thereby reassigned to the storage with shortest distance. There are no channel bottlenecks (excess capacity)
Network	1-N-16, starting as 1-1-16, ends as1-16-16 linear.
Assertions	Assert that the runtime does not increase exponentially.

Results





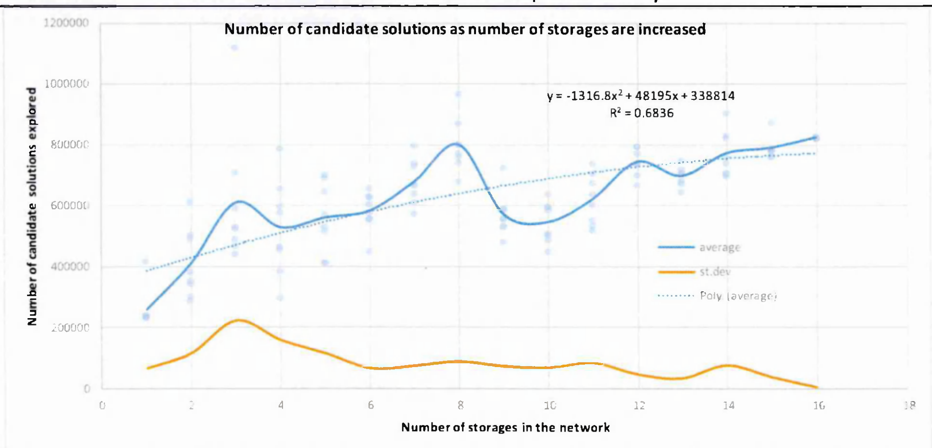
Results Data

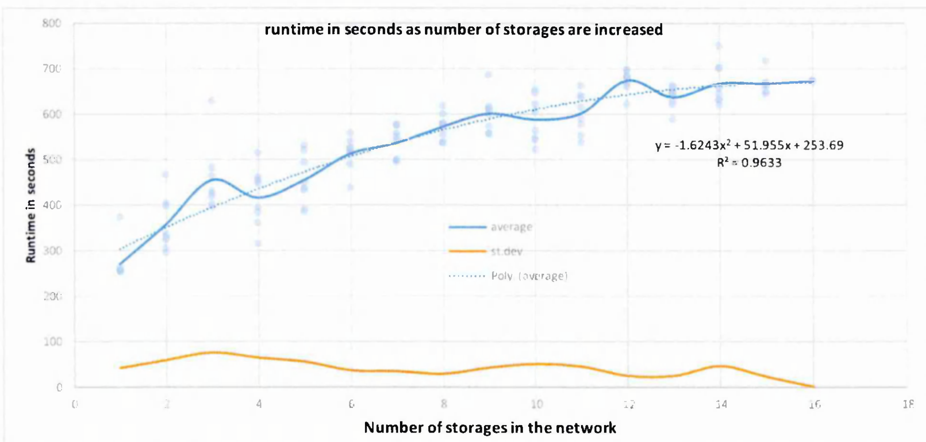
number of storages	number of candidate solutions average	st.dev	Experiment No. 1	2	3	4	5	6	7	8
1	21,456	613	20013	21506	21373	21604	21626	21967	21883	21672
2	22,607	457	22476	22939	22096	23350	22086	23040	22412	22457
3	23,532	480	24476	23512	23373	23486	23826	23373	22773	23440
4	25,796	1,163	25397	25950	24526	26686	25234	26612	24275	27688
5	25,842	1,423	27760	24602	23963	24412	26027	26379	27581	26010
6	27,026	832	27041	27674	26856	26483	26261	25993	27335	28562
7	28,296	1,217	28575	28887	27796	27858	27704	27831	26821	30895
8	27,580	764	27386	27910	26922	26668	26792	28269	28793	27901
9	29,192	708	29026	30146	29620	28529	28517	28363	29248	30086
10	29,185	675	29162	27996	29200	29614	30167	29459	29404	28478
11	28,812	457	29171	29566	29084	28967	28523	28411	28551	28219
12	29,085	617	28940	29437	29299	28305	28067	29904	29308	29419
13	29,189	341	28945	29326	28892	29326	28892	29416	28892	29823
14	28,944	348	28661	28661	29086	29098	28661	28661	29622	29098
15	29,473	400	29371	30363	28964	29605	29371	29371	29371	29371
16	29,092	-	29092	29092	29092	29092	29092	29092	29092	29092

number of storages	time in seconds average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
1	53	0	53	53	53	54	53	54	53	53
2	58	1	58	59	56	58	56	59	57	58
3	62	1	62	62	62	61	63	62	60	62
4	66	1	67	65	66	67	67	66	64	68
5	70	1	72	68	69	69	70	71	70	70
6	74	1	74	76	74	72	74	73	75	73
7	78	1	78	76	78	78	78	78	77	79
8	80	1	81	81	78	79	80	82	82	80
9	83	1	84	84	83	83	81	81	83	84
10	85	1	85	82	86	85	86	85	86	85
11	87	1	87	88	87	88	86	85	87	86
12	89	1	88	89	90	88	87	90	90	89
13	91	0	91	91	91	91	91	91	91	92
14	93	1	93	93	93	94	93	93	94	94
15	96	0	96	96	95	96	96	96	96	96
16	98	-	98	98	98	98	98	98	98	98

Conclusion	The assertion holds True. The runtime increases sublinearly.

Case ID	Limited channel capacity (PPar2)
Category	Increasing problem complexity
Description	This takes an existing network and schedules in batch mode. In each step the number of storages (1-N-16) between a factory and 16 retail outlets are increased. The retail outlets are thereby reassigned to the storage with shortest distance. In contrast to the previous test (PPar) this scenario has limited capacity on the channel from the source to the storages.
Network	1-N-16, starting as 1-1-16, ends as1-16-16 linear.
Assertions	Assert that the runtime does not increase exponentially.
Results	





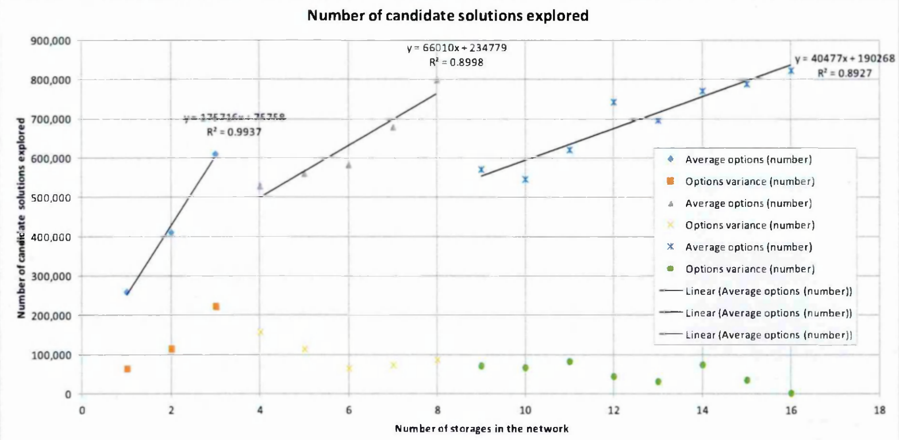
Results Data

number of storages	number of candidate solutions average	st.dev	Experiment No. 1	2	3	4	5	6	7	8
1	259,525	64,236	231179	242046	238567	236996	236282	418326	235764	237039
2	411,089	114,647	615153	492760	302763	289573	354561	384280	505136	344484
3	610,958	222,552	1121387	592771	527205	491965	442666	711091	530444	470132
4	529,401	157,510	458089	599205	299248	656905	789528	466860	578963	386407
5	561,154	113,938	701838	692702	531206	413122	414225	517413	571833	646890
6	583,718	65,031	577849	450791	632404	628697	658562	556112	606533	558794
7	679,784	73,360	730771	798284	641311	667578	741184	672091	612107	574942
8	800,135	87,104	871672	967646	802184	773184	680325	742166	802267	761638
9	571,445	71,344	725261	530928	559293	591998	588283	558265	481901	535628
10	546,520	66,773	586904	509768	598167	595900	504064	639149	488509	449696
11	621,788	82,457	638793	553218	676397	519703	604570	739086	534251	708282
12	744,175	44,357	739433	773279	795690	795745	668480	742212	701719	736842
13	698,077	31,443	707962	698791	749964	681353	706971	646055	673721	719801
14	772,363	74,243	698439	823414	709392	905301	741016	706725	763103	831513
15	790,584	35,497	789131	761916	783856	785480	779475	763922	874677	786213
16	824,889	2,283	823778	821540	826470	822668	826284	828472	825901	824001

number of storages	time in seconds average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
1	271.25	41.62	254.00	263.00	257.00	257.00	257.00	374.00	254.00	254.00
2	357.63	59.08	467.00	403.00	306.00	296.00	330.00	337.00	398.00	324.00
3	456.00	76.04	630.00	463.00	425.00	419.00	402.00	482.00	431.00	396.00
4	416.88	64.53	384.00	462.00	315.00	455.00	515.00	395.00	448.00	361.00
5	455.50	55.69	521.00	532.00	450.00	385.00	391.00	433.00	437.00	495.00
6	513.38	36.30	518.00	438.00	540.00	513.00	558.00	526.00	524.00	490.00
7	537.13	34.49	558.00	577.00	496.00	550.00	576.00	542.00	499.00	499.00
8	573.00	28.60	601.00	619.00	580.00	556.00	539.00	537.00	579.00	573.00
9	601.63	42.27	687.00	557.00	610.00	617.00	612.00	573.00	557.00	600.00
10	588.38	50.40	606.00	548.00	647.00	622.00	564.00	654.00	544.00	522.00
11	602.25	44.77	615.00	578.00	637.00	538.00	591.00	664.00	553.00	642.00
12	674.50	24.17	683.00	680.00	697.00	698.00	623.00	684.00	662.00	669.00
13	637.75	24.01	643.00	648.00	660.00	622.00	663.00	589.00	630.00	647.00
14	668.00	45.91	620.00	704.00	633.00	751.00	646.00	631.00	658.00	701.00
15	667.75	21.89	670.00	645.00	668.00	667.00	665.00	649.00	717.00	661.00
16	674.00	-	674.00	674.00	674.00	674.00	674.00	674.00	674.00	674.00

Conclusion

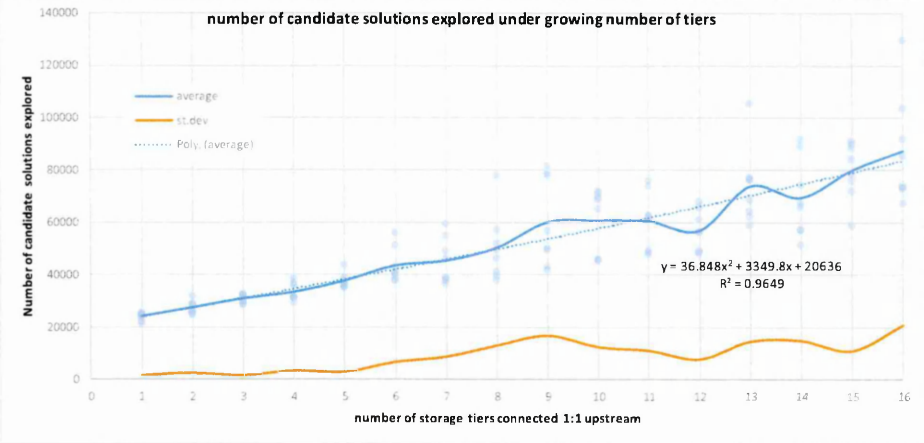
The assertion holds True, but not with a linear extensible model.
The chart below shows how the solution-space folds around the trend line defined in the first chart, but is non-linear in each segment.

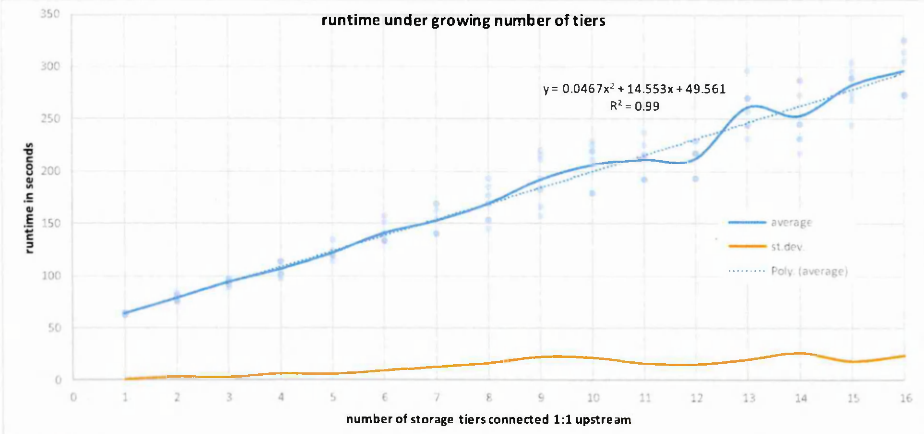


Increasing route length with longer chain of storages

Case ID	Excess channel capacity (PSS)
Category	Increasing problem complexity
Description	In a basic network 1-4(N)-16 network additional Tiers of storages are added with 1:1 channels. Hereby the length of the supply chain increases with N.
Network	1-4-16 at start, 1-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-16 at the end.
Assertions	Assert that the runtime does not increase exponentially.

Results





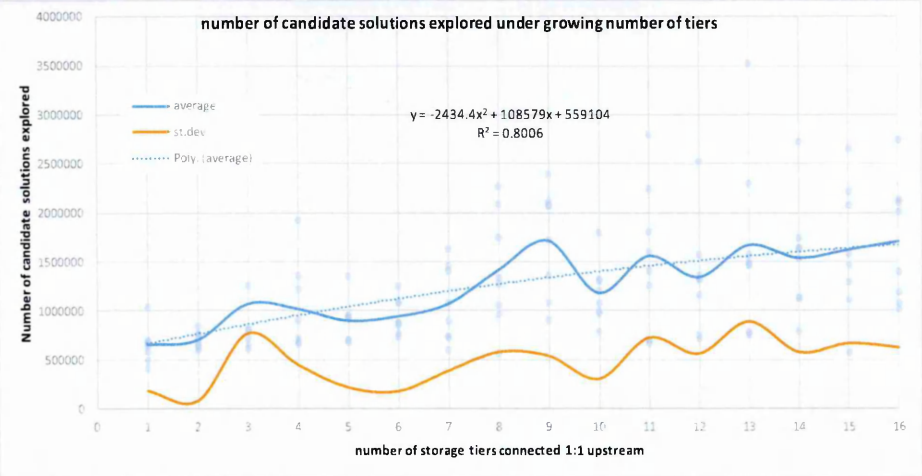
Results Data

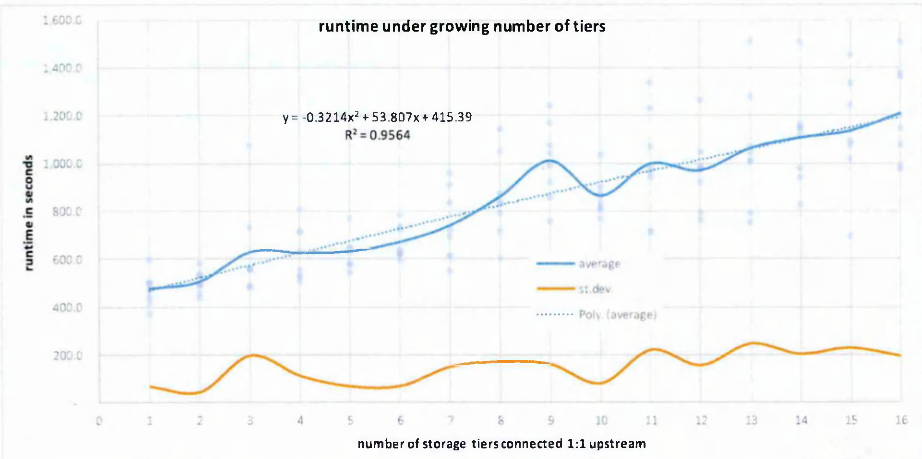
number of storage tiers	number of candidate solutions		Experiment No.							
	average	st.dev	1	2	3	4	5	6	7	8
1	23,987	1,445	22267	23943	24725	24789	24202	25127	21363	25480
2	27,331	2,424	28951	25871	28749	27232	24966	31929	26085	24866
3	30,792	1,584	31300	32618	29792	31990	32511	28628	28824	30674
4	33,285	3,296	31521	31303	35998	38553	29404	31526	31272	36700
5	37,568	2,820	37816	36311	36810	38994	43830	35392	35212	36176
6	43,440	6,641	56089	51336	38094	40055	38071	42833	40335	40708
7	45,291	8,685	59570	38551	46239	37953	36661	55026	49271	39059
8	50,285	13,005	52015	78037	57347	41229	37975	49591	39739	46343
9	60,113	16,866	41819	49951	56989	78639	49845	43289	81932	78439
10	60,625	12,404	71930	68855	45578	65477	45998	71364	69886	45912
11	60,477	11,080	74231	49482	63162	47921	48779	62461	61742	76034
12	56,862	7,770	56511	68143	48616	56511	61198	65986	49318	48609
13	73,859	14,592	76896	62461	77279	76361	64463	105525	59159	68727
14	69,492	14,896	57407	66148	89308	57303	74558	92013	51657	67539
15	80,052	10,966	72215	90196	59033	84326	91395	78606	76144	88501
16	87,424	20,914	92130	103743	85586	67484	73892	73510	73257	129790

number of storage tiers	time in seconds		Experiment No.							
	average	st.dev.	1	2	3	4	5	6	7	8
1	64	1	63	63	65	64	64	65	63	65
2	79	3	82	76	82	78	76	84	76	76
3	94	3	95	98	92	98	95	89	92	95
4	107	7	102	102	114	114	98	102	106	114
5	122	6	120	120	120	125	135	115	120	120
6	141	9	152	158	134	141	134	140	134	134
7	153	13	170	141	155	141	141	169	163	141
8	169	17	170	194	186	154	146	178	154	170
9	192	23	158	183	194	212	185	167	216	221
10	206	22	230	220	180	212	180	226	220	180
11	211	17	238	193	215	193	193	217	215	226
12	212	16	218	230	194	218	218	230	194	194
13	261	21	271	245	271	271	245	297	232	258
14	253	27	232	246	288	232	274	288	218	246
15	283	19	290	290	245	290	296	275	269	305
16	296	24	306	327	315	274	274	274	274	325

Conclusion	The assertion holds True. The signal propagation solution implemented for a chain evidently works effectively for multiple parallel chains.

Case ID	Limited channel capacity (PSS2)
Category	Increasing problem complexity
Description	In a basic network 1-4(N)-16 network additional Tiers of storages are added with 1:1 channels. Hereby the length of the supply chain increases with N. In contrast to the previous test the channel capacity is limited at the tier between the source and the first storage tier (worst case scenario)
Network	1-4-16 at start, 1-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-16 at the end.
Assertions	Assert that the runtime does not increase exponentially.
Results	



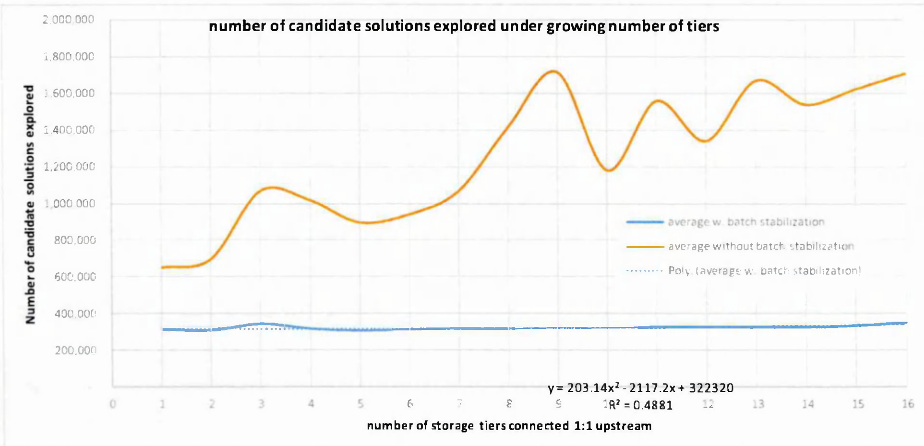


Results Data

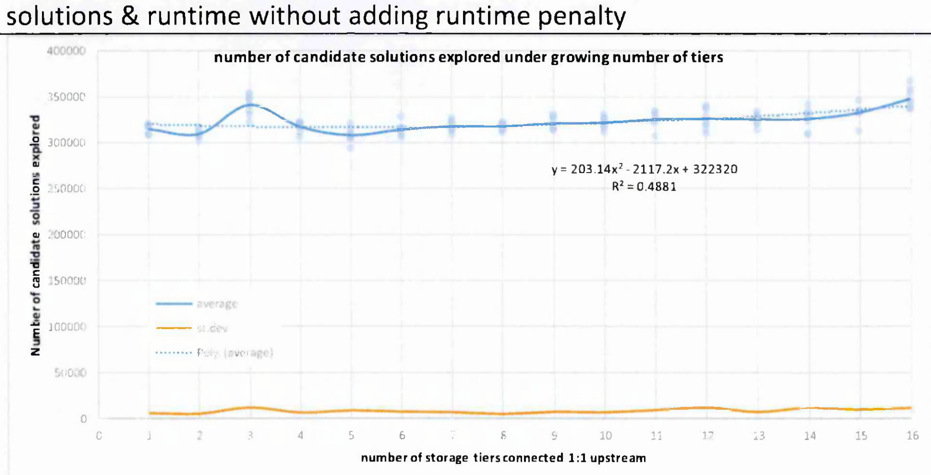
number of storage tiers	number of candidate solution		Experiment No.							
	average	st.dev	1	2	3	4	5	6	7	8
1	651,185	185,842	644155	662223	491110	688142	586460	1038446	408497	690444
2	697,860	81,897	593819	621777	839690	643488	745102	704288	763351	671364
3	1,067,957	773,857	815306	2918507	1254783	636934	716857	805835	606084	789346
4	1,017,385	452,374	897047	683865	1218571	661058	722114	676663	1923655	1356107
5	895,674	220,918	1352548	951007	697388	943717	699309	925740	919510	676171
6	939,342	180,952	732800	870637	849204	1083338	765229	1252191	877324	1084015
7	1,064,982	389,384	1406034	729103	733423	1631397	894542	1454989	595726	1074644
8	1,411,929	582,882	1746804	2268583	1334408	563928	1275281	961694	2089225	1055508
9	1,714,910	542,349	2389513	1360599	1723335	2069254	1082283	2077728	2114710	901860
10	1,179,769	307,359	1796353	1198255	1318425	1305043	994407	1072057	967115	786493
11	1,557,164	728,224	1257779	1804187	2794561	1598009	1402815	668050	692099	2239814
12	1,339,418	564,733	1161602	1569029	2522714	709587	1358889	1327104	747562	1318857
13	1,668,092	893,434	1466976	778706	3522841	1571304	1490381	2297084	752597	1464848
14	1,537,451	581,261	1739724	787155	1627793	1132329	2722103	1641446	1529487	1119569
15	1,621,102	670,185	570549	2660069	2076592	1467536	1288635	2216181	1107626	1581629
16	1,706,262	627,153	1072790	2742379	2133042	1395531	1186816	2005508	1011209	2102818

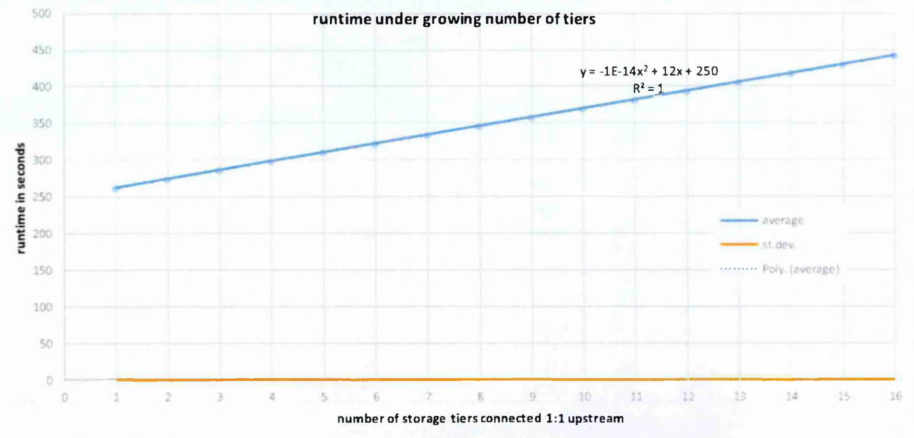
number of storage tiers	time in seconds average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
1	475.9	68.3	506.0	471.0	417.0	501.0	442.0	600.0	373.0	497.0
2	505.0	45.7	440.0	458.0	583.0	499.0	539.0	499.0	531.0	491.0
3	628.0	196.9	559.0	1,077.0	735.0	491.0	557.0	561.0	484.0	560.0
4	625.4	112.0	634.0	508.0	715.0	558.0	527.0	532.0	811.0	718.0
5	631.1	69.1	774.0	639.0	585.0	649.0	546.0	649.0	630.0	577.0
6	670.0	70.5	594.0	624.0	618.0	737.0	630.0	789.0	642.0	726.0
7	739.1	150.0	838.0	614.0	615.0	961.0	696.0	913.0	552.0	724.0
8	858.8	172.6	876.0	1,146.0	832.0	607.0	841.0	796.0	1,053.0	719.0
9	1,009.8	160.4	1,245.0	924.0	996.0	1,077.0	859.0	1,045.0	1,172.0	760.0
10	863.4	81.0	1,036.0	873.0	904.0	817.0	808.0	863.0	834.0	772.0
11	997.5	220.6	969.0	1,072.0	1,340.0	990.0	942.0	715.0	718.0	1,234.0
12	969.4	156.2	922.0	1,051.0	1,268.0	765.0	977.0	993.0	796.0	983.0
13	1,061.6	246.3	1,048.0	796.0	1,515.0	1,014.0	1,075.0	1,283.0	755.0	1,007.0
14	1,106.4	203.4	1,149.0	831.0	1,148.0	983.0	1,512.0	1,167.0	1,118.0	943.0
15	1,134.4	228.6	698.0	1,456.0	1,338.0	1,097.0	1,078.0	1,248.0	1,021.0	1,139.0
16	1,207.5	194.6	994.0	1,511.0	1,366.0	1,149.0	1,080.0	1,204.0	977.0	1,379.0

Conclusion	The assertion holds True.



Case ID	Limited channel capacity (PSS3)
Category	Increasing problem complexity
Description	In a basic network 1-4(N)-16 network additional Tiers of storages are added with 1:1 channels. Hereby the length of the supply chain increases with N. In contrast to the previous test the channel capacity is limited at the tier between the source and the first storage tier (worst case scenario)
Network	1-4-16 at start, 1-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-16 at the end.
Assertions	Verify that usage of batch-stabilized mode removes the variance in candidate solutions & runtime without adding runtime penalty
Results	



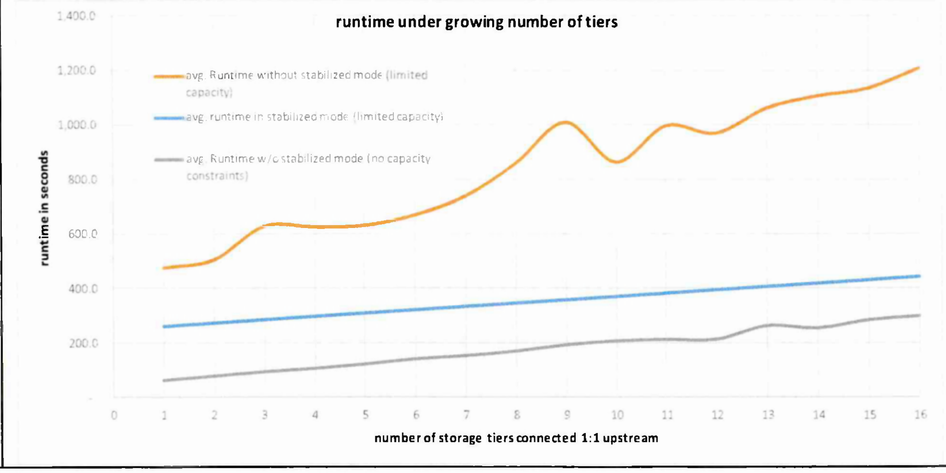
Results Data

number of storage tiers	number of candidate solutions average	st.dev	Experiment No. 1	2	3	4	5	6	7	8
1	315,182	5,309	319486	308879	308592	318236	309583	316004	319597	321076
2	309,777	4,658	317193	310621	314410	307421	307232	310166	309291	301883
3	341,585	11,354	346971	353081	341348	354370	330515	348798	334971	322627
4	317,518	5,811	317229	322971	311493	320549	317899	306381	321947	321673
5	308,205	7,960	308675	306445	306796	315023	321075	309959	303699	293965
6	314,447	6,702	314953	305960	316669	312667	328531	313848	314443	308505
7	318,167	6,144	320561	309249	327129	314798	313549	323338	313929	322783
8	318,143	4,244	320520	320263	311325	320619	320605	323309	314650	313854
9	321,511	6,624	317917	330997	317282	314107	314430	328592	327102	321663
10	322,219	6,223	327026	326950	324298	311888	318234	330848	321071	317437
11	326,122	8,707	334653	325638	321987	330053	329874	334341	324791	307636
12	326,662	11,565	337286	340856	320486	321101	310272	338338	314965	329993
13	326,003	6,585	314330	321050	329706	321508	331120	334736	327929	327641
14	326,339	11,272	328929	342002	331607	334792	324241	328902	310857	309383
15	333,097	9,371	313010	333370	347174	335266	335748	333438	332688	334083
16	348,092	10,916	343707	358568	353804	339788	337829	337179	346372	367487

number of storage tiers	time in seconds average	st.dev.	Experiment No. 1	2	3	4	5	6	7	8
1	262	-	262	262	262	262	262	262	262	262
2	274	-	274	274	274	274	274	274	274	274
3	286	-	286	286	286	286	286	286	286	286
4	298	-	298	298	298	298	298	298	298	298
5	310	-	310	310	310	310	310	310	310	310
6	322	-	322	322	322	322	322	322	322	322
7	334	-	334	334	334	334	334	334	334	334
8	346	-	346	346	346	346	346	346	346	346
9	358	-	358	358	358	358	358	358	358	358
10	370	-	370	370	370	370	370	370	370	370
11	382	-	382	382	382	382	382	382	382	382
12	394	-	394	394	394	394	394	394	394	394
13	406	-	406	406	406	406	406	406	406	406
14	418	-	418	418	418	418	418	418	418	418
15	430	-	430	430	430	430	430	430	430	430
16	442	-	442	442	442	442	442	442	442	442

Conclusion

The signal propagation solution implemented for a chain evidently works effectively for multiple parallel chains.


runtime under growing number of tiers
avg. Runtime without stabilized mode (limited capacity)
avg. runtime in stabilized mode (limited capacity)
avg. Runtime w/o stabilized mode (no capacity constraints)
runtime in seconds
1,400.0
1,200.0
1,000.0
800.0
600.0
400.0
200.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
number of storage tiers connected 1:1 upstream

A.2.4 considerations for implementation

To verify whether New Supply Chain Model reliably could handle a large supply network a scalability test was conceived, where the experiment was based on the 1-5-80 (synthetic) network with a randomized demand with 100% density for the time period. The experiment was performed on an 80 core 1.0 Ghz Nehalem (160 hyper threads) High Performance Cluster with 2TB of DRAM provided by Microsoft Research in Cambridge.

Test objective	10k SKU	20k SKU	Goal
Factories	1	1	1
Distribution Centres	5	5	5
Retail outlets	80	80	80
SKUs	10,000	20,000	100,000
Time Horizons	365 days	365 days	365 days
Demand type	Daily uniform random		
Order density per day	1.00	1.00	1.00
Service level pursued	100%	100%	100%
Runtime (hh:mm:ss)	02:10:21	07:44:11	See test report
RAM usage	220Gb	436Gb	2180Gb
Output file size	4.93 Gb	9.88 Gb	49.40
Share of goal	10%	20%	100%

Table 31 configuration & test results

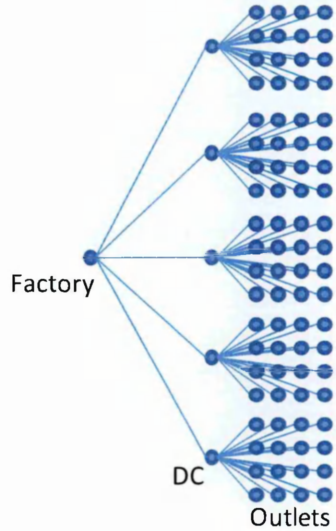


Figure 50 Network topology for 1-5-80

The experiment showed that the MATmodeller could schedule the demand without any form of optimisation and without any runtime errors. The following summarizes the observations concerning the output:

+ The "saw tooth" curve of supply and consumption from DC's was optically correct.
+ Randomness of demand is verifiable.
+ Deliveries utilize the channel 100%.
- Un-anticipated lost sales in the beginning and during the year < 0.01% of demand.

The following observations raise awareness for the industrial version:

Observation	Solution
Only 60 cores of the 80 cores were used effectively	Pin threads to cores
HPC socket topology blocks at N>2 CPU Sockets	Use message parsing between hyper threads instead of locking.
Large amount (30%) of time is spent on garbage collection	To not drop memory objects to garbage collection. Update instead.

Application must be written to use pools of objects in memory which is allocated to each hyper thread so that memory access across sockets is prevented by using message parsing instead.

A.2.4.1 Performance considerations for Multi-Agent Architecture

A classic argument in design of applications is first to get the logic right, then to tune the performance to achieve the desired runtime. In my observations this process is the root-caused of scalability failures. The best approach is to design the architecture to be scalable on jungle architecture from the first beginning, as it makes the developer actively think about how to avoid the common pitfalls of scalability and performance compromises. However it also requires a rigorous decomposition of the logic.

In the following I will present some key lessons[34] from the development of the New Supply Chain Model and some recommendations to operate with a logic as close to functional programming paradigm as possible. The latter is because functional programming requires that each functions does one thing only, but it does it well.

K-groups

A lesson learned from the scalability test with New Supply Chain Model was the scalability of running 584 million objects was that hardware utilization dropped radically on a multi-core architecture.

This was caused by our application not being K-group aware[35]. K-Groups are a construct that appear once a machine has more than 64 cores. The Windows kernel is then partitioned into multi K-group, each with their own kernel structures, memory and processors. If an application is not K-Group aware Windows will "pin" the app to a single K-group, which in the case of this machine, probably is either 30 or 60 processors[36]. If you are familiar with NUMA memory, this is very similar, but at a higher level in the hierarchy and with much stricter rules on which cores you can run on.

The solution is that the application must be specifically coded to support K-groups or alternatively, to split the application into multiple processes. Each of them will be able to run in their own K-group (the boundary is per process), but of course, once the application is split into multiple processes, the problem of inter process communication arises.

Small datasets take longer on large scale homogeneous architectures

On larger machines, the clock rate of the CPU cores is much lower than on smaller machines (about two thirds). This happens because this is the only way to dissipate the heat from the cores and synchronise memory access (laws of physics constraints).

On the NUMA structure of such a many core machine, such as the Nehalem CPU series used in the experiment, has four memory controllers on each die. It needs to balance the access to NUMA local memory with remote memory across these controllers. This happens because a single CPU cannot manage a lot of memory, so multi CPU sockets are needed to manage the 2TB of DRAM.

---

[34] By Thomas Kejser, Fri, Jul 19, 2013 at 9:23 AM

[35] http://msdn.microsoft.com/en-us/library/windows/desktop/dd405503(v=vs.85).aspx

[36] http://msdn.microsoft.com/en-us/windows/hardware/gg463349.aspx

Consider the topology of a one socket first, a non NUMA system: CPU - 4 connectors - DRAM

On a two socket system the resulting architecture changes: DRAM = CPU = CPU = DRAM

Since there are two connectors to memory and between the two CPU, the difference from a single socket to a two socket appears negligible. So for less than 16 cores, most standard applications work fine.

But consider what happens when 4 sockets are required, as on high memory machines:

```
DRAM - CPU - CPU - DRAM
         |  X  |
DRAM - CPU - CPU - DRAM
```

The result is that more time is spend on coordination between cores to synchronise the access to DRAM as x86/x64 has strict memory ordering.
On the test machine used for the large supply chain test the machine has 8 sockets, so the performance penalty is quadratic:

Every time a memory access is required, there the penalty applies if the target memory object is located on a CPU that is not directly accessible for the hyper thread requiring the memory object. The penalty is significant, typically in the range of 200-300ns. In addition while a CPU is waiting for memory access it is stalled - and it will report 100% CPU load, even though it is not doing anything other than waiting. This effect is replicable on all systems, but it worse on large (>64 core) systems.

If the application contains additional exclusivity to memory objects using locking/latching/spinning, the resolution of the locks will take longer to acquire and longer to release. This is where the software design itself limits scalability.

Likewise programmers in the current era rarely worry about garbage collection. So software developers unfortunately make heavy use of garbage collection, which will be significantly slower of large systems. It is therefore considered poor coding practice to design applications which repeatedly creates and destroys memory objects. A much better approach is to reuse objects instead of creating new ones using object pools which may act as general memory containers.

On large NUMA machines, object pools must be partitions. Windows lookaside lists in C++ (and Linux Slab allocators) are examples of how to implement such a pool in a scalable manner for each NUMA node.

Some things to consider for data structures

To run well on large systems, the code must use data structures that are scalable and carefully design the way it does the threading.

- Partition data structure per core. For example, if the application has a counter that keeps track of something, make that counter an array of counters (once per CPU core) and let each thread always update the "local" one. Windows will registers which core the application managers' thread is on.
- Pin threads to cores

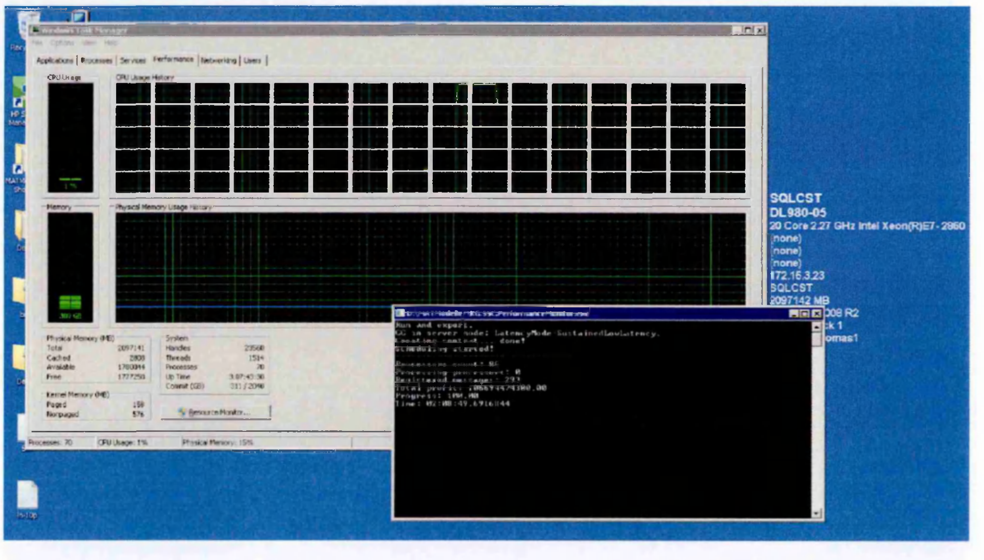
- Make use of scalable locking structures[37]. This is a tough discipline and generally not possible on the .NET platform.
- Make the code aware of where objects are in memory[38]. For example, if a thread allocates an object and modifies it, prefer to use this thread to access the object in the future[39]. This requires significant scheduling considerations and may require that a custom CPU scheduler is written for this.
- Use Lock/wait free data structures. This is an area of a lot of research, but there are good code samples available.

Best practice for development is hard to point out. Troubleshooting the scale problems in the code starts with a CPU level profiling (because the memory stalls show up as CPU busy). Then compare a CPU profile of a workload on a 2 socket with a larger N socket machine. If you are lucky, you will see that some small parts of the code (on the same workload) takes longer on the 8 socket. Typically, this is synchronisation code: Spinlocks/Locks/Latches or access to some central data structures. You can then look into replacing those lock and data structures with more efficient versions. If things go well, you may be able to make some "surgical" fixes that will make it scale better.

A word of warning on scalable data structures: Make sure you check that the results are OK... Really counter intuitive things happen when you venture into lock free data structures.

A.2.4.2 Large scale testing - 10k SKU

The experiment runs the 1-5-80 synthetic network with 10,000 SKUs and a uniform random demand generator which creates daily demand.



Scheduling results

As the scheduling results cover 10,000 SKUs at 86 sites, providing an complete overview is infeasible, due to amount of information. Instead a sample is provided:

---

[37] There are ways to write your own data structure that are faster. I would recommend this book : http://www.amazon.co.uk/The-Multiprocessor-Programming-Maurice-Herlihy/dp/0123705916. It uses JAVA for the examples, but he idea is the same in .NET

[38] http://www.akkadia.org/drepper/cpumemory.pdf "What every programmer should know about memory"

[39] http://www.1024cores.net/ - good Russian site.

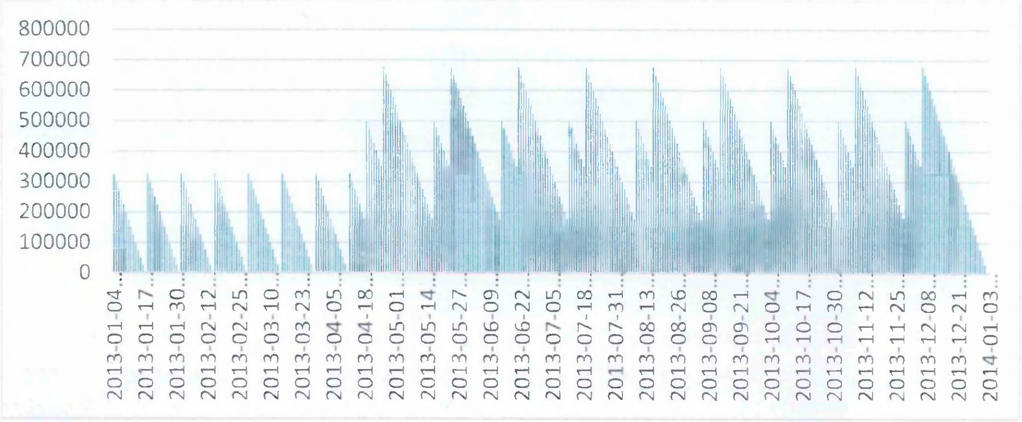


Figure 51 Demand Site #1's total inventory profile.

Storage site #1 stock states

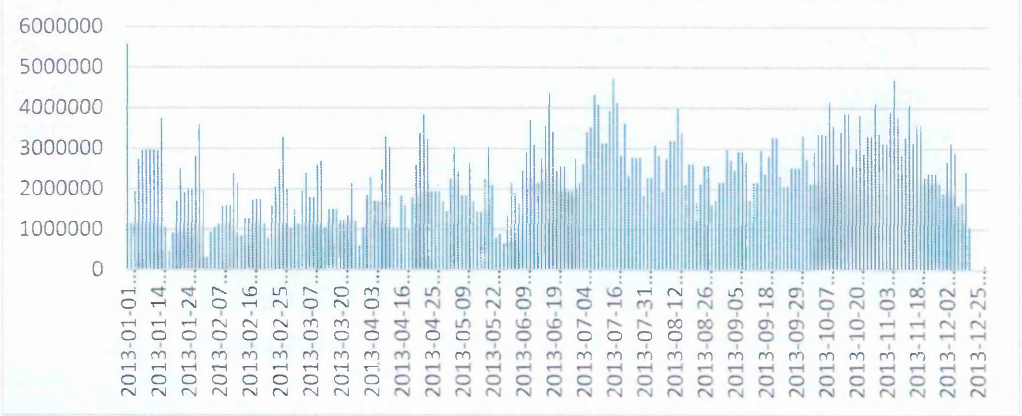


Figure 52 Storage site #1's inventory profile (distribution centre).

Demand site #1 deliveries are characterised by full utilization of the supply chain until the end of the schedule where it is more parsimonious to empty the inventory.

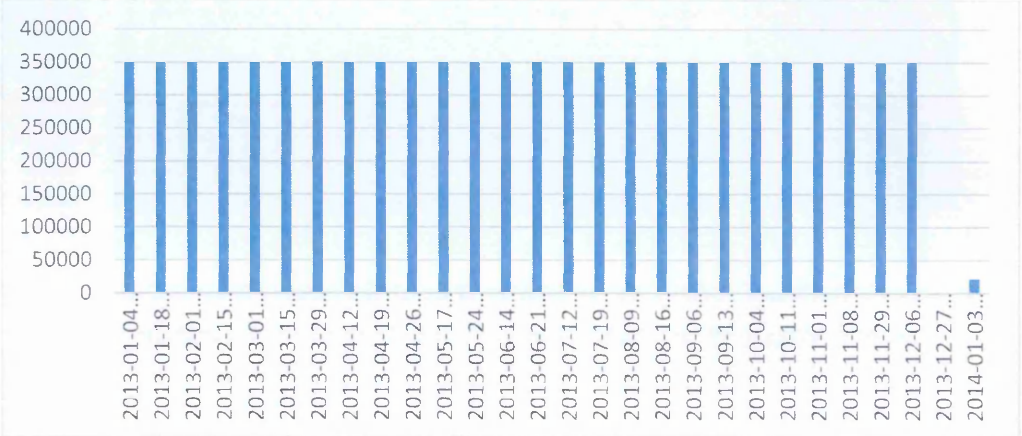


Figure 53 Demand Site #1's inbound deliveries.

Issues detected

Garbage collector performance problems

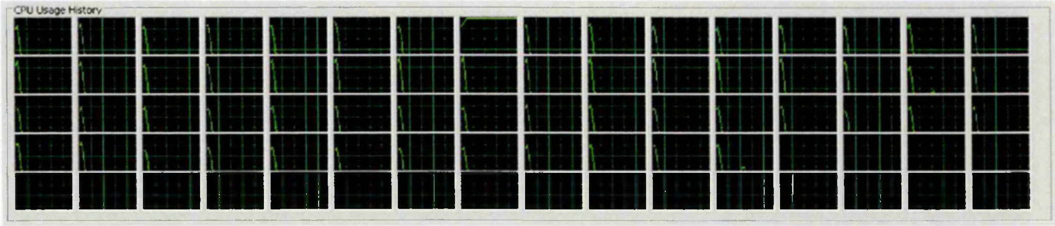


Figure 54 Screenshot of the CPU utilization at the end of the scheduling. Notably only 60 cores where utilized.

Lost a lot of time for collecting unused objects.

Strange lost sales on demand sites

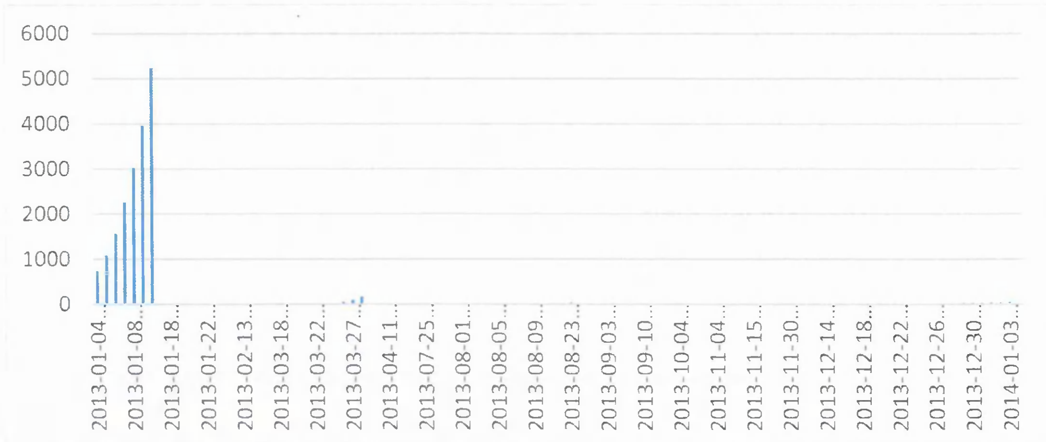
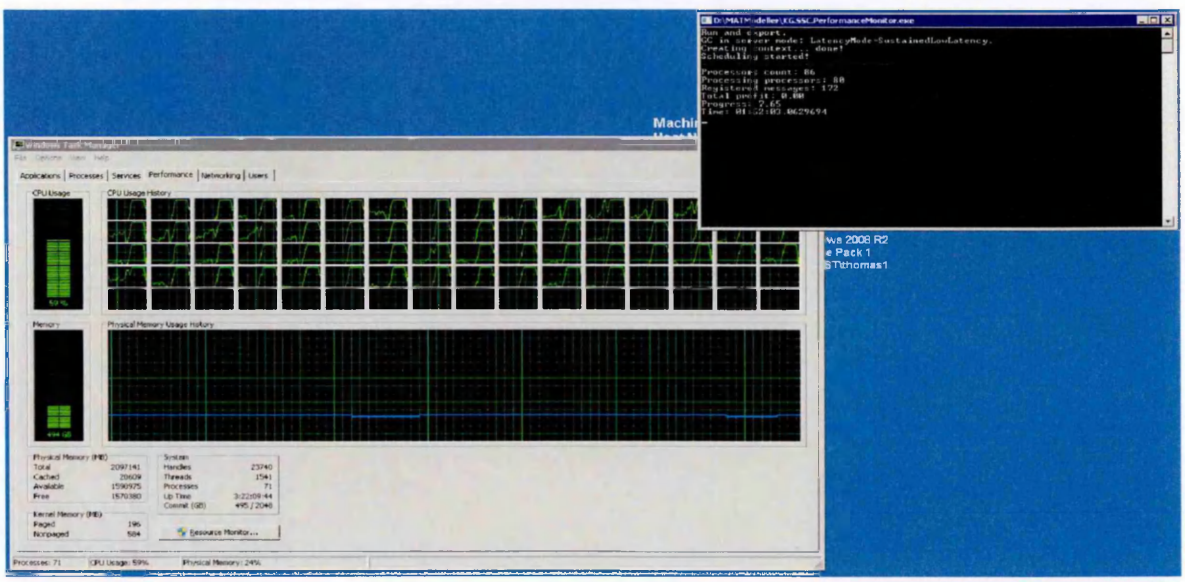


Figure 55 Lost sales (up to 5000 units) at the beginning and ranging 1-20 units throughout the 365 days.

Up to 5000 at the beginning; 1-20 in the middle of the year.

A.2.4.3 Large scale testing - 20k SKU

The test was performed in the same manner as with the 10,000 SKU test, expect the number of SKUs generated for the synthetic test covered 20k.



Scheduling results

Demand site #1 stock states

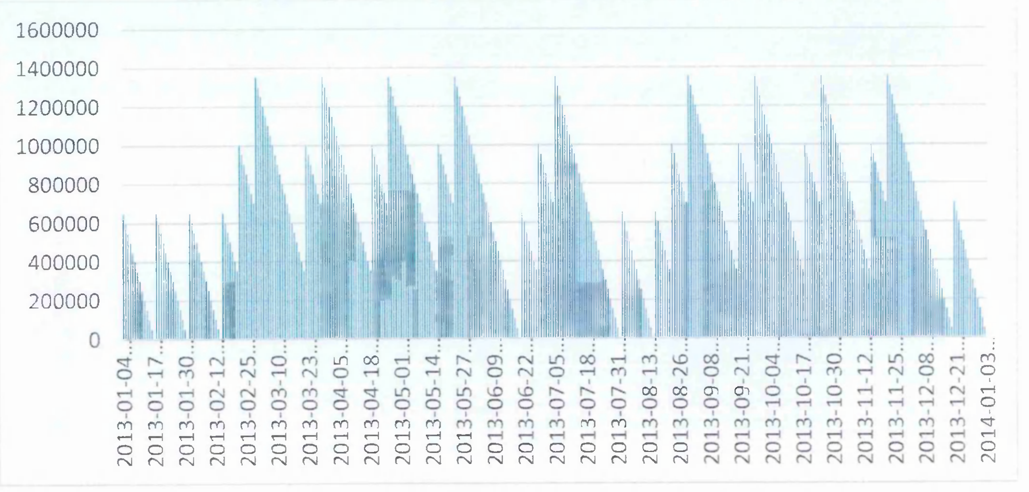


Figure 56 Demand site #1's inventory profile.

Storage site #1 stock states

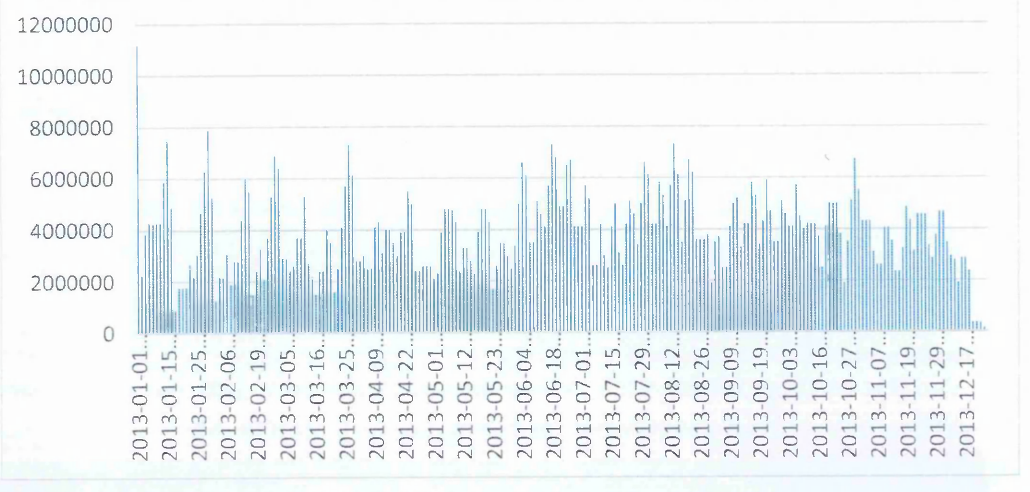


Figure 57 Storage site 1#'s stock state (distribution center).

Demand site #1 deliveries

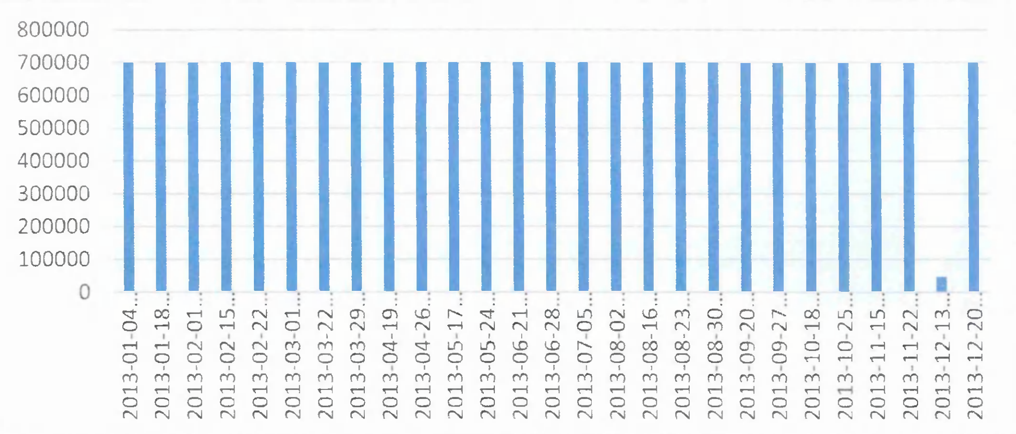


Figure 58 Demand Site #1's inbound deliveries

Issues detected

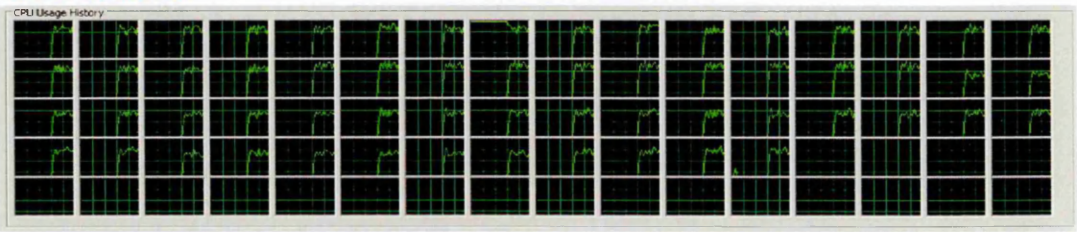


Figure 59 Windows Task Managers presentation of the CPU utilization.

Garbage collector performance problems

Strange lost sales on demand sites

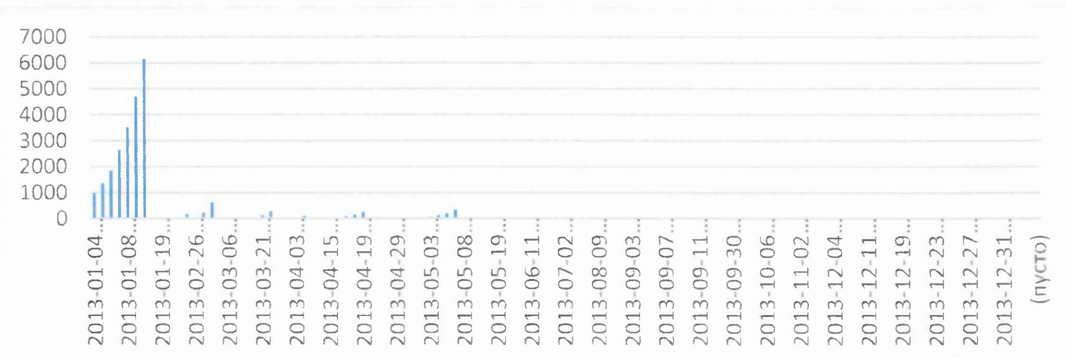


Figure 60 Unexplained lost sales in the range from 1-6050 units

A.3 Literature review on supply chain management (extension)

A.3.1 Review Question

To inform a business manager on how to make the most productive intervention in the complex economic system he or she may exert influence on, the following literature review seeks to clarify:

What is the critique of current intervention methods for SCM?

This literature review is not intended to be exhaustive, but rather intended to identify key aspects of the process and concepts which enable the decision maker to transform information about the state of the supply chain into a schedule of resource allocations, which result in the most productive intervention.

A.3.1.1 Review Scope

The field of enquiry is viewed in the context of supply chains as a complex economic system, where three fields of research overlap (as illustrated in Figure 61, below):

- The relationship amongst agents in control of resources, which is dominated by the thinking within supply chain management literature and influenced by organisation theory and complex- and general system theory.
- The methods for transforming information into decisions, which is dominated by applications of management science, operations research and lately significantly influenced by methods of artificial or augmented intelligence.
- The models of information exchange, which covers the interpretation and propagation of information.

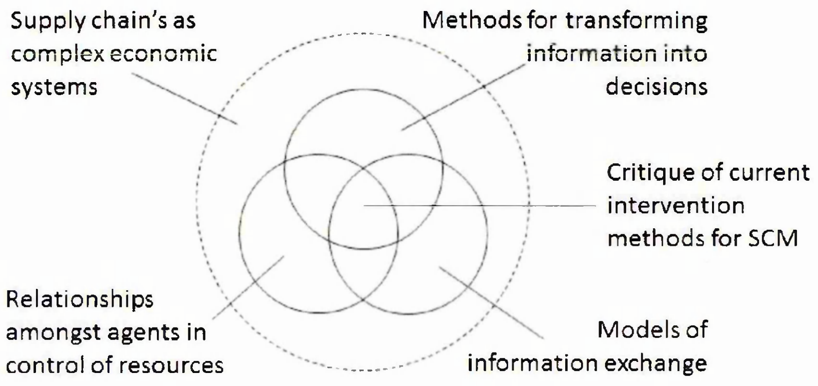


Figure 61 positioning the field of enquiry

Other fields of research that influence the complex systems perspective, such as microeconomics, operations management and theories of meta-interventions in management are to some degree included implicitly through examples, as presentation of the literature otherwise may appear to depart from its intended pragmatic and applied perspective.

A.3.1.2 Review method

Based on a search using the search string "supply chain" and "literature review" for a period from 1984-2014, used in Google Scholar's[40] search engine, the top 15 newest and publicly available literature reviews are obtained. Though the top 15 cited articles are subject to the superstar effect (Rosen 1981) due to the Google Scholar's usage of the PageRank-algorithm (Chen et al. 2007), this entry point for the literature review is followed by a back-tracking of articles and books, which totals approximately 2200 references. Of the top 15 articles, key authors cited are extracted, based on the following appraisal criteria (Table 32, below).

#	Criteria	Score
1	Recognise the discrete choice in management	{0,1}
2	Provide a critical stance to the conditions for which decisions may be made	{0,1}
3	Present a theoretical model that is consistent with practice	{0,1}
4	Have methods which extend beyond a single functional problem	{0,1}

Table 32 Appraisal criteria of literature

The back-tracking of literature is supported with supplementary searches in google scholar and using the search string "supply chain" and "model".

#	Source	Type	Score
1	(Mentzer et al. 2001)	Journal	4
2	(Cooper et al. 1997)	Journal	4
3	(Lambert & Cooper 2000)	Journal	4
4	(Lambert et al. 1998)	Journal	4
5	(Srivastava 2007)	Journal	4
6	(Da Silveria et al. 2001)	Journal	4
7	(Croom et al. 2000)	Journal	4
8	(Vidal & Goetschalckx 1997)	Journal	4
9	(Stadtler 2005)	Journal	4
10	(Melo et al. 2009)	Journal	4
11	(Gunasekaran & Ngai 2004)	Journal	4
12	(Meixell & Gargeya 2005)	Journal	4
13	(Gunasekaran & Ngai 2005)	Journal	4
14	(Huang et al. 2003)	Journal	4
15	(Burgess et al. 2006)	Journal	4

Table 33 top-15 most cited literature reviews on SCM

In addition a set of high quality books and papers were added which provide valuable contrasts to the literature review:

#	Source	Type	Score
1	(Christopher 2005)	Book	4
2	(Vollmann et al. 2005)	Book	4
3	(Shapiro 2007)	Book	4
4	(Harrison & Hoek 2008)	Book	3
5	(Oliveira & Gimeno 2014)	Book	3
6	(Swaminathan et al. 1998)	Journal	4

[40] Includes Elsevier, IEEE, arXiv, Association for Computing Machinery Digital Libray, Citebase, CiteSeer, DBLP, IEEE Xplore, Microsoft Academic Research, Science.gov, ScienceDirect, SpringerLink.

7	(Angerhofer & Angelides 2000)	Journal	3
8	(Dooley 2009)	Journal	4
9	(Ellram & Cooper 2014)	Journal	4
10	(Stadtler & Kilger 2005)	Book	3
11	(Leitao & Vrba 2011)	Journal	4

Table 34 Selected high quality books

A point of critique of the review method is that it does not present statistics or charts of the journal papers, book chapters and secondary sources, as the focal search on critique of the subject "how to make the most productive intervention in a complex economic system" returns an empty set. The sources presented thereby position the problems around the focal subject, and will collectively provide an overview from the 3 perspectives (Figure 61, above). The literature review that follows will reveal why, this specific case is a pending question that has not found existing answers.

A.3.2 Relationships amongst Agents

A.3.2.1 Perspectives

The relationships amongst agents in the supply chains are in typically described from a predetermined perspective, with either focus on (i) network design, the (ii) transactions and the information system which supports the transactions, or the (iii) managerial function (Vidal & Goetschalckx 1997; Cooper et al. 1997; Lambert & Cooper 2000; Lambert et al. 1998; Angerhofer & Angelides 2000; Huang et al. 2003; Huang & Zhang 2007; Gunasekaran & Ngai 2004; Gunasekaran & Ngai 2005; Meixell & Gargeya 2005; Stadtler & Kilger 2005; Stadtler 2005; Burgess et al. 2006; Melo et al. 2009; Christopher 2001; Christopher 2005; Christopher & Rutherford 2004; Allesina et al. 2010; Oliveira & Gimeno 2014). These perspectives are outlined in the three tables below. Models for supply chain management, tend to include some of the listed perspectives and factors, though so far no framework have covered them all, nor has had sufficiently abstract framework to construct all aspects.

Network design	1. Number, location, capacity and type of facilities (plants and warehouses) 2. Sources of supply and demand, and contractual terms 3. Transportation modes and possible channels 4. Macroeconomic conditions (stability, security, transparency and trade culture)

Transactions	1. Microeconomic decisions made by members of the chain of peers exchanging information (including information systems), which trigger logistic activities. 2. Organisation of business processes which balance supply with demand, such as planning and control of: 2.1. Procurement of raw materials, 2.2. Inventory, including coordination of production and shipping between facilities (routing), work-in-progress and finished goods. 2.3. Allocation of available stock to confirmed demand. 3. National interests such as customs declaration and operations associated with cross-border/cross-trade zone transactions (taxes, duties, exchange rates, trade barriers, transfer prices).

Managerial function	Strategic capabilities 1. Financial planning, such as evaluation of investments in 1st and 2nd tier suppliers 2. Comparative studies of alternative service model- and supply network-designs.
	Tactical allocations (connecting capacities to need for transactions) 1. Planning of material and resource requirements based on demand, bills-of-materials, lead time and available production and delivery methods. 2. Allocation of resources to committed portfolio of activities
	Operational continuity (completion of pending transactions) 1. Completion of order management, pick, pack & dispatch including administrative obligations 2. Preparation and complementation of production and maintenance 3. Resolution of conflicts caused by unexpected events.

Vidal & Goetschalckx's (1997) summarise their critique as follows:

> "The [...] considerations allow us to claim that there exist many research opportunities for developing more comprehensive global supply chain models that include BOM constraints, more stochastic factors, and qualitative aspects that are very important within a global environment. Specific opportunities for research are the following:
>
> – explicit inclusion of more stochastic features in modeling international supply chains;
> – consideration of vendor and transportation channel reliability in the selection of vendors and transportation channels;
> – inclusion of customer service level as part of the set of constraints;
> – explicit modeling of potential economies of scale existing in interactional supply chains;

- simulation of qualitative factors, such as the general infrastructure of a country;
- differentiation of products by country;
- determination of adequate excess capacity in different countries;
- coordination of commodity flows, cash flow, and information flow within an international environment;
- modeling of alliances and multi-company network configurations; and
- development of specialized methods of solution."

(Vidal & Goetschalckx 1997, pp.14–15)

> "it is easy to conclude that there exists a lack of features in the existing strategic models for the design of supply chains [...]. The main drawback of these models is the fact that most uncertainties are not considered in the formulations. In addition, there does not exist a formal and consistent way to represent BOM constraints. Moreover, some international factors, such as exchange rates, taxes, and duties are not fully described by the existing models."
>
> (Vidal & Goetschalckx 1997, p.15).

They highlight that "[m]any authors do not present specific models to manage the supply chain, but describe important additional aspects to consider in any formulation." (Vidal & Goetschalckx 1997, p.9) and that the research has been focused on discovery and classification of concepts to be included in the models of the supply chain, which precedes the development of a coherent theoretical framework.

Swaminathan et al. (1998) attempt to "provide a modular and reusable framework with primitives that allow development of realistic supply chain models" (Swaminathan et al. 1998, p.612) which is departs from the dominant usage of mixed integer and linear programming using manufacturing, distribution and transportation ***agents*** where the supply chain is defined as a network of autonomous business entities which collectively are responsible for procurement, manufacturing and distribution activities associated with one or more families of products. In this attempt Swaminathan et al. (1998) conclude that an approach which does not exploit discrete event simulation, but instead pursues a reductionist approach is bound to provide answers of limited application:

> "One of the prime concerns while managing a large supply chain is how to control the inventory within the supply chain while providing the required service to customers. It is impossible to have tractable analytical models for these problems under realistic assumptions." (Swaminathan et al. 1998, p.626).

This observation speaks in advance of approaches in which the model represents the requisite complexity of the real the world on a one-to-one scale which addresses Vidal & Goetschalckx's (1997) critique, by going into a detailed description of how to construct a multi-agent based discrete event simulation and address the point that a major flaw in the discrete event simulation, is its lack of support to the operational levels. The paper also provides a critique of the usage of statistics in decision support for managers who need to understand the consequence of their chosen alternatives. This absence of simulation in the moment of choice, is not addressed in literature until critique of accounting methods appear in 2000-2005 (Kaplan & Anderson 2003; Goldratt & Cox 2004; Kaplan & Anderson 2004; Geri & Ronen 2005).

Angerhofer & Angelides (2000) review the usage of system dynamics modelling in supply chain management where their critique highlights:

> "Current research on System Dynamics Modelling in supply chain management focuses on inventory decision and policy development, time compression, demand amplification, supply chain design and integration, and international supply chain management." (Angerhofer & Angelides 2000, p.342).

Their concern however is that even though system dynamics has provided answers to specific questions, dating back to Forrester's work in 1958, the authors argue that it is odd, that system dynamics do not find presence consistently in all supply chain models. From the literature around year 2000 it becomes clear that research communities occupied with operations research have coexisted with limited interaction, though have many lateral contributions to show.

A.3.2.2 The fragmented set of theoretical models

Tako & Robinson (2011) provide a summary of 127 journal articles over a period from 1996-2006, to categorise the usage of different approaches to optimisation in SCM, challenging the assumption regarding discrete event simulation (68% of articles) and system dynamics modelling (30% of articles, 2% hybrids), that the former is more suitable for operational/tactical problems, whilst the latter is suitable for strategic issues. Tako & Robinson (2011, pp.23–24) asked the critical question: "Did the SD models address an issue better than the DES models, or vice versa?" and conclude that "[t]his would be difficult to establish because detailed information about the models and their impact is not always made readily available in the papers." As a reference to the critique this must be considered a weakness that the community of SCM does pride itself of the simulations, but publish work that is neither replicable, nor transparent. One can only speculate why this is, though Katz (2013) provide the general critique that the citation impact system does not provide sufficient incentive for researchers to publish software which permits reproducible research. Rather, the citation impact system provides incentive for the researcher to produce as many low-quality papers which barely pass the threshold for the journals, so that volume - not parsimony, rigour or novelty - is represented. (Katz 2013) write:

> Issues of motivation are of particular concern today[41] as science becomes more collaborative (aka team science), and as collaboration leads to more — and better — science.[42] The average number of authors per paper is increasing, and collaborative projects are becoming common, which is part of the cause for the increasing number of paper authors.

Croom et al. (2000) provides more evidence for this observation with a wide ranging literature review created for classification purpose same year as

---

[41] Howison, J., Herbsleb, J. D. Incentives and integration in scientific software production. In Proceedings of the 2013 conference on Computer supported cooperative work, pp. 459–470 (2013).

[42] Wuchty, S., Jones, B.F., Uzzi, B. The Increasing Dominance of Teams in Production of Knowledge. Science 316(5827). pp. 1036-1039 (2008).

Angerhofer & Angelides (2000). In the process they discover that only 17% of the literature contains theoretical contributions:

> "One of the most significant findings from our literature analysis has been the relative lack of theoretical work in the field when compared to empirical based studies. Our concern with the finding that the literature is primarily empirical-descriptive is that any development of a cognate supply chain management discipline requires more rigorous and structured research in the topic." (Croom et al. 2000, pp.74–75).

The year 2000 is interesting as it coincides with the growth of public access to internet search engines and simultaneously brings a new generation of enterprise database systems online. For practice this means that a lot of research shifts from solving specific research problems, towards experimentation with hybrid planning methods. Stadtler & Kilger (2005) summarize this in in the preface of their third edition of *Supply Chain Management and Advanced Planning*:

> "The field of Supply Chain Management (SCM) and Advanced Planning has evolved tremendously since the first edition was published in 2000. SCM concepts have conquered industry – most industry firms appointed supply chain managers and are "managing their supply chain". Impressive improvements have resulted from the application of SCM concepts and the implementation of Advanced Planning Systems (APS). **However, in the last years many SCM projects and APS implementations failed or at least did not fully meet expectations.** Many firms are just "floating with the current" and are applying SCM concepts without considering all aspects and fully understanding the preconditions and consequences."

From Stadtler & Kilger's description it is shown that little consideration have been given to the problem that *execution systems* must respond to higher-ranked planning systems in case of disruption, even though most systems are designed for hierarchical top-down propagation of events (Chapter "4.1 What is Planning" (Stadtler & Kilger 2005, p.86). This problem still haunts many planning systems – in contrast to management systems - today. An example hereof is Ivanov & Sokolov (2013) who claim that the supply chain coordination problem has not received sufficient attention when one or more processes are modernised and thereby exposes the supply chain to disruption of continuity of output, even though the disruption can be planned.

In 2007 the critique of SCM and IT becomes stronger as Shapiro (2007) presents the view that supply chain management needs to distinguish explicitly about the discrete steps in decision making. Shapiro makes it clear that in order to explore and exploit the economic system the supply chain is a part of it is essential to differentiate what is transactional in contrast to what is analytical. He reminds his readers that transactional IT has the role of capturing the company's discrete events, whilst analytical IT is responsible for evaluating the information to make fact based decisions about future commitments. With this distinction in mind, Shapiro (2007) divides the decision models into descriptive (forecasting, data mining, activity-based costing, performance metrics, simulation, systems dynamics) and prescriptive modelling (optimisation models) well knowing that there is a problem: Decision makers are not rational and have limitations in attention time, they have deteriorating recall rate of events from memory, have limited ability to deal with complex mental models and are physically limited in the ability to communicate dense information. Shapiro (2007) therefore arrives to the conclusion that the state-of-art organisation is conscious about three different foci to the role of optimisation in supply chain management which IT systems must support:

- The role of modelling the supply chain in order to explore potential opportunities.
- The role of IT to enable exploitation of candidate solutions in the solution set available through optimisation methods
- The role of supply chain managers and analysts to perform operational interpretation of candidate solutions.

Christopher's and Gattorna's work is well aligned with Shapiro's critique as they attempt different approaches towards operational supply chain alignment through collaboration between people (Christopher 2005; Gattorna 2006; Shapiro 2007). However they do not present a solution to the coordination problem between the rigid approaches to planning and the planning systems lack of capability to effectively incorporate disruptive events that they all (Stadtler & Kilger 2005; Gattorna 2006; Christopher 2005; Shapiro 2007) are concerned about.

A.3.3 Transformation of Information into Decisions

A.3.3.1 Overview

The methods used for transforming information into decisions are emphasised by their focus on some more or less explicit set of objective functions. Whilst the rigorous treatment of this subject is multi-objective optimisation (Coello 2006; Fu 2002), the more soft or inspirational is presented as leading the agents of the supply chain towards joint coordinated efforts of delivering the customer value proposition (Porter 2008; Christopher 2005; Gattorna 2006). Across the literature performance indicators are used to indicate the relative ability of agents to work towards the set objective functions. The dimensions of the objective functions typically belong to the classical MBA/M.Sc. SCM curriculum and range across customer experience, profit (revenue, costs) and the ability to execute at strategic, tactical and operational levels:

1. Factors which influence customer expectations:
   - 1.1. Brand expectations, reputation.
   - 1.2. Value-proposition means of product/service/image differentiation
2. Factors which influence revenue:
   - 2.1. Trade enabling factors, such as availability of service and efforts/cost of trade, i.e. being visible in the market.
   - 2.2. Order fulfilment rate: right time, right product, right location, right quantity and to right terms & conditions.
3. Factors which influence costs:
   - 3.1. Fixed and variable -production, -facility, -vendor/order, -transport and –production line costs; including costs associated with hedging, volume contracts and loans.
   - 3.2. Cost of capital from work-in-progress, inventory (pipeline-, cycle- and safety-stock) and excess inventory caused by lack of influence to coordinate/forecast demand, including lack of supplier reliability.
   - 3.3. Taxes, duties and other regulatory fees including licensing fees of IP-rights.
   - 3.4. Depreciation of obsolete and overdue products

3.5. Government subsidies (cost reduction)
4. Factors which influence ability to execute at all levels:
   4.1. Human resources, talent
   4.2. Information systems
   4.3. Human/computer interaction

Table 35 Dimensions used to characterise objective functions of the supply chain

Badole et al. (2012) provide the latest and most extensive a review of 690 papers[43] with detailed insight in the publications of papers which concern supply chain models, with focus on particular problems and the method used for solving the problem. Badole et al. (2012) find papers of supply chain models in 24 journals with 53.97% (of 302 papers) in the International Journal of Production Economics and the European Journal of Operational Research. The diversity of methods used is an extensive mix of 17 methods (Genetic algorithm, system dynamics, mathematics, linear programming, game theory, simulation, Taguchi methods, dynamic sequencing, fuzzy sets, mixed integer and linear programming, sensitivity analysis, Markov chains, petri net, agent based simulation, Lagrangian mechanics, ant colony optimisation, artificial neural network) for 3 key problem categories (planning supply and demand, operational planning/scheduling and network design). The two tables below provides a summary of applied quantitative methods in literature.

#	Method	Supply & Demand Planning[44]	Scheduling[45]	Supply Network Design
1	Stochastic approximation[46]	2		
2	Ranking and selection	2		
3	Game Theory	4		1
4	Markov chain	3	2	
5	Petri net	1	4	1
6	Fuzzy Logic	3	3	2
7	Combinatorial optimisation	1	1	
8	Simulated annealing		3	
9	Dynamic Programming (divide and conquer)		2	2
10	Artificial Neural Network	1		1
11	Lagrangian relaxation	2		1

[43] In (Badole et al. 2012, p.78) citations [59] and [64] are duplicates with errors in the authors title, so whilst the work covers a lot of papers, there is still opportunity for improvement of rigour.

[44] Including forecasting

[45] Including travelling salesman's problem and its derived routing problems

[46] Including iterative attempts to identify extrema which can only be estimated, not computed

12	Mixed integer and linear programming	9	5	17
13	Monte Carlo simulation	1		1
14	Discrete event simulations (DES)	7	4	3
15	DES with system dynamics	1	6	1
16	Genetic algorithms	2	15	3
17	Tabu Search	1	1	
18	Particle Swarm optimisers		6	
19	Ant Colony Optimisers	1	2	1
20	Agent Based Models	1	43	1

Table 36 Methods and focus

#	**Author**
1	(Robbins & Monro 1951; Nemirovski et al. 2009)
2	(Runarsson & Yao 2000; Chan & Chung 2013; Giovannucci et al. 2007)
3	(Neumann et al. 1944; Huang & Zhang 2007; Shoham & Leyton-Brown 2008; Caro & Martinez-de-Albeniz 2010)
4	(Srivastava 2007; Shoham & Leyton-Brown 2008; De Boer & Boer 2000)
5	(Viswanadham & Raghaven 2000; Badole et al. 2012; Van der Aalst 1998; Biswas & Narahari 2004)
6	(Chan & Chung 2013; Bollen et al. 2010)
7	(Bidot et al. 2008; Giovannucci et al. 2007)
8	(Chan & Chung 2013; Kirkpartick et al. 1983; Iridia et al. 1997)
9	(Johnson 1954; Bellman 1986; Wu et al. 1999; Vidal & Goetschalckx 1997)
10	(Grljevic & Bosnjak 2011; Bollen et al. 2010; Astor & Adami 2000)
11	(Badole et al. 2012; Lidestam & Ronnqvist 2011)
12	(Badole et al. 2012; Shapiro 2007; Vidal & Goetschalckx 1997)
13	(Badole et al. 2012; Shapiro 2007; Vidal & Goetschalckx 1997)
14	(Chan & Chung 2013; Tako & Robinson 2011; Moon & Phatak 2005; Mönch et al. 2011; Shapiro 2007)
15	(Chan & Chung 2013; Angerhofer & Angelides 2000; Tako & Robinson 2011; Pathak et al. 2007)
14	(Power & Sharda 2007; Moon & Phatak 2005; Matuszek & Mleczko 2009; Klemmt et al. 2009; Mönch et al. 2011)
15	(Siebers et al. 2010)
16	(Konak et al. 2006; Horn et al. 1994; Coello 2000; Ghosh & Dehuri 2004; Poli et al. 2008; Slak et al. 2011; Chan & Chung 2013; Dimitrov & Baumann 2011)
17	(Badole et al. 2012; Chan & Chung 2013)
18	(Martinez & Coello 2011; Engelbrecht 2005; Zhang et al. 2011; Mohemmed et al. 2007; Fidanova 2005; Chan & Chung 2004; Silva et al. 2002)
19	(Meuleau & Dorigo 2002; Dorigo 1992; Ilie et al. 2010; Iridia et al. 1996; Bakhouya & Gaber 2007)
20	(Anosike & Zhang 2007; Max Gath, Stefan Edelkamp 2013; Siebers et al. 2010; Andreev et al. 2007; Madsen et al. 2012; Akanle & Zhang 2008; Allan 2009; Leitao & Vrba 2011; Chatfield et al. 2007; Ivanov et al. 2010; Leitão 2009; Turgay 2009; Zhang et al. 2006; Gath et al. 2013; Brintrup 2010; Skobelev 2011; Lau et al. 2006; Holmgren 2008; Chan & Chung 2013; Smith 2010; Neagu et al. 2006; Fox et al. 2000)

Table 37 Authors and methods. Some authors compare several methods.

From the overview it should be noted that the only method which may incorporate the rest is Agent Based Modelling (ABM) which combines discrete event processing through message exchange amongst agents, with internal rules of achieving state updates. The internal methods can thereby use all 20 methods, including ABM within or integrated with other ABM. Several authors therefore conclude that ABM is the way forward, with statements such as:

> "Agent technology has been recognized as a promising paradigm for next generation manufacturing systems." (Shen et al. 2006, p.415)

The mixture could indicate that the SCM community still is experimenting with methods. Evidence of this hypothesis is that, for example, that few authors are publishing papers on more than one problem solving method, and none publish for more than four methods, and that several authors spend sections to argue in order to obtain peer-acceptance of the notion of "optimality" when dealing with multi-objective optimisation problems, as optimality does not constitute the classical mathematical optimum (Coello 2006, p.29).

The casual reader could speculate which approach is more productive: To adopt the most widely distributed techniques or to focus novel methods which are not widely deployed? The answer which is more likely to yield the most productive outcome, is that the niche problems which were attempted to be solved were prompted by specific conditions, and are as such not supply chain problems. In other words: The problems are selected for anticipated success of usage of the tools and not necessarily the productivity of the supply chain as a competing entity. Mönch et al. (2011) raise the issue that the operations research and production oriented journals who approach optimisation from a modelling perspective receive harder critique, than if the author selected a known method and constrain the problem to suit the method.

A.3.3.2 Academic bias in choice of objective functions

Melo et al. (2009) attempts to systematically explore what supply chain models have been focused on and discover that 75% of the literature is mainly focused on costs, compared to 9% multiple objectives and 16% on profit (Melo et al. 2009, p.408). Despite the critique of transparency of the models which were used, which was raised by Vidal & Goetschalckx in 1997, Melo et al. (2009) argue that a clear and specific algorithm can be traced in 75% of the articles when associated with facility location problems, though they declare that in "most of them the structure of the supply chain network is considerably simplified" (Melo et al. 2009, p.409). They add:

> "In addition to these findings, we note that the large majority of location models within SCM is mostly cost-oriented. This somewhat contradicts the fact that SCND[47] decisions involve large monetary sums and investments are usually evaluated based on their return rate."... "...Moreover, substantial investments lead to a period of time without profit. Companies may wish to invest under the constraint that a minimum return will be gradually achieved." ... "By considering profit-oriented objective functions, it also makes sense to understand, anticipate and react to customer behaviour in order to maximize profit or

[47] Supply Chain Network Design

revenue. This means bringing revenue management ideas into strategic supply chain planning." Melo et al. (2009, p.410)

The last statement cannot be emphasised enough: Revenue management has been left out of consideration of supply chain design for most of its history, disregarding the fact that cost reduction is a question of minimizing the cost-driving activities even though some cost-driving activities also may be highly profitable. Combined with the observation that 75% of the articles are associated with facility location problems, this observation should raise alerts with the critical reader, as facility location problems are the most prominent business investments and influence many jobs. But if they only are evaluated from a cost-perspective, the models will favour facilities, which combines economies of scale, forcing a centralization into the planning approach. This is a serious problem when the models are used to inform management decisions!

A.3.4 Information Exchange amongst Agents

A.3.4.1 Overview

The critique of choice of methods is repeated for studies of practical applications. Badole et al. (2012), for example, raise attention towards the absence of interactively negotiated compromise between optimisation methods. This is an important critique as information exchanged by systems between businesses often is a naïve propagation of demand (see Figure 62, below, where there is no feedback loop from purchasing and MRP).

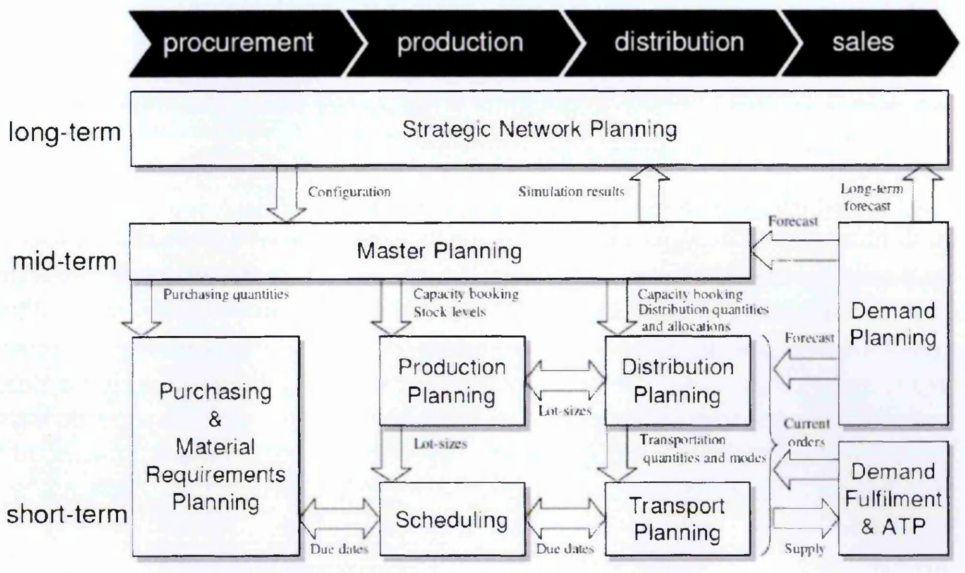


Fig. 13.1. Coordination and data flows of APS modules

Figure 62 From Stadtler & Kilger (2005, p.246) *Illustration of data flow amongst modules.*

Combined with classic optimisation techniques each business will perform sub optimisation of what it expects its suppliers to do disregarding what the supplier might be capable of delivering (Snapp 2009). Badole et al. (2012)'s critique summarizes both the asynchronous human-computer interaction and the system-to-system interaction, which the industry and academia fails to recognise in their implemented models:

"While there is an abundance of SC management literature, it is realized that research at the inter-organizational level is less prevalent. However, the objective of SCM is to integrate all the firms in the value chain and treat them as a single entity (global supply chain). **Notwithstanding, the current research has failed to look at that perspective of the SCM.**" (Badole et al. 2012, p.75)

Fowler & Rose (2004) attempts to synthesise the key challenges for practical exploitation of the modelling and simulation methods as:

1. *An order of magnitude reduction in problem solving cycles*
2. *Development of real-time simulation-based problem solving capability*
3. *True Plug-and-Play Interoperability of Simulations and Supporting Software within a Specific Application Domain*
4. *Greater Acceptance of Modeling & Simulation within Industry*

And, Shen et al. (2006) state six requirements for what they call "next generation manufacturing systems" where they refer to systems used for practical exploitation of potential benefits for the supply chain as a whole:

R1. Full integration of heterogeneous software and hardware systems within an enterprise, a virtual enterprise, or across a supply chain;

R2. Open system architecture to accommodate new subsystems (software or hardware) or dismantle existing subsystems "on the fly";

R3. Efficient and effective communication and cooperation among departments within an enterprise and among enterprises;

R4. Embodiment of human factors into manufacturing systems;

R5. Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments;

R6. Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the working environment. (Shen et al. 2006, p.416)

These industry requirements contrast the relevance of academic publications which claims successful solutions to synthetic[48] problems. Shen et al. (2006) reflect on the research with self-criticism:

"Many researchers (particularly Ph.D. students) working on agent-based manufacturing are still focusing on the fundamental research to enhance the rationality or intelligence of software agents and develop more efficient and effective coordination and negotiation mechanisms. While this kind of research is important and still needed, we believe that the future R&D work should focus on the integration of agent-based planning and scheduling systems with existing systems used in manufacturing enterprises. The most important integration is with real time data collection systems, ... Another important integration is with existing ERP and MRP systems. Note that a certification is required for integrating or interfacing with some commercial ERP/MRP systems. ***Only when such integrations are achieved and validated in industrial settings, will the***

[48] I found it more appropriate to write "synthetic problems" than irrelevant problems

> ***agent technology be widely applied in manufacturing industry.***" (Shen et al. 2006, p.427)

Oliveira & Gimeno (2014) take a different approach and attempt to bring SCM into the boardroom by informing shareholders of the impact of supply chain management as a cross functional task. Their presentation is very lightweight on matters that in previous literature was dominated by operations research, though it provides a general overview of the financial aspects of supply chain management. Though the classical operations researcher might frown upon the approach that Oliveira & Gimeno (2014) take, it is *necessarily superficial* on operational matters. This reinforces the influence of microeconomics in SCM: That money is an instrument to correct the direction of an organisation, and to depart from the myopic cost saving models. That being said, the financially oriented SCM literature is not unbiased in its choice of arguments either. Ramsay & Croom (2008) criticise the use of emotionally charged adjectives such as "strategic", "evolution" and "advancing" in association with purchasing and supply management literature, as they take the stance that decision making sciences is supposed to inform managerial decision making and leave the seduction using loaded terms to consultancies. Their first observation is that 99% of companies have 99 employees or less, which means that the "strategic" scope that is outlaid by researchers is of relevance to less than 1% of the businesses. Whilst they express understanding that the focus on "strategic" factors lead to social recognition as "doing something that is important", it does not justify complete ignorance of necessary non-strategic (clerical and administrative) tasks. This bias in the supply chain management literature is reflected in the models of supply chains where the dominant focused is on producing the most impressive business case and discovering whether an impact of change is significant or not. Ramsay & Croom (2008) find this research-bias unjustified, as disruption of operations can lead to the situation that anything suddenly becomes of "strategic importance" – even if the disruption may have been caused by a clerical mistake. Ramsay & Croom (2008) conclude:

> *"The critical evaluation of established beliefs in knowledge fields is essential to establish clarity in conceptual definitions, and in this paper we have argued that there are significant concerns around some of the concepts and metaphors currently in use."*

This critique is not hard to follow. Vollmann et al. (2005) uses the term "advanced" in the connection with classic linear and mixed integer programming for (i) sales and operations planning, (ii) propagation systems (just-in-time and material requirements planning) and (iii) scheduling. Stadtler & Kilger (2005) are no different, and use the term "advanced" in association with any coupled system.

A.3.4.2 Coherence and consistency of definitions

Given the development of optimisation software over the past 10 years, the choice of words might be justified, though it exposes research in the field of SCM as poorly informed about optimisation techniques. At the surface, the management literature leaves the impression from the moment mathematics or algorithms are introduced which extend beyond undergraduate topics, the subject suddenly becomes "advanced". This bias departs strongly from the legacy in systems dynamics theory and operations research (Nair & Vidal 2011) where mastery of undergraduate material may be assumed. The tendency in parts of the literature appears biased towards narratives which appeal to management consultancies and does not require the rigour and grit expected in other parts of system dynamics, such as in fluid dynamics or fields associated with physics. Dooley (2009) notes that this tendency applies for supply chain management in general and that the journals, societies, and doctorate programs have a vested interest in maintaining the dichotomy.

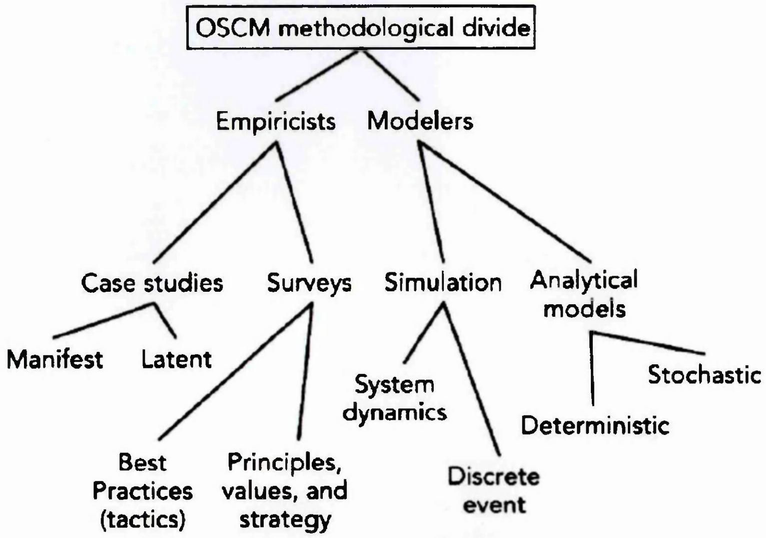


Figure 63 The Empiricism-Modeler Dichotomy in Operations and Supply Chain Management (OSCM) (Dooley 2009)

Dooley (2009) concludes that the disciplinary branches of SCM knowledge and theory inhibit the field from accumulating knowledge as rapidly as it otherwise could, and that the narrow focus of branches allow individuals to make cogent arguments about the novelty of their studies. Whilst that may be true for a particular branch that uses the label of SCM, the breadth and depth of SCM makes it harder to make such claims once moving closer to the point of methodological differentiation. Dooley advises that researchers should be afforded the chance to get training and exposure to methodologies from across the spectrum (Figure 63, above), as this enhances the methodological expertise. The cross-functional work produced by empiricists and modellers should result in a more coherent and consistent presentation of the system of the world (Dooley 2009).

A.3.5 Summary of critique

The summary of the critique is organised in three main groups as presented in Figure 61, above:

- Relationships and transformation of information
- Relationships and information exchange
- Information exchange and transformation of information

Supply chain is recognised as a complex adaptive system (CAS) and a part of the complex economic system (CES). The state-of-the art literature uses tools and knowledge from operations research, computer science, organisation theory, finance/economics and applied mathematics in pursuit to enable the supply chain of businesses to operate as productively as possible. The literature of SCM struggles to present the balance it pursue between optimal exploitation and being adaptive enough to cope with changes, arriving from the environment, which influence the operational condition.

Lack of transparency of models which integrate the many creative models, and lack of genuine theoretical contributions (Croom et al. 2000) are obstacles for the development of a coherent framework. This is supported by the recent years of failure to produce better management systems which integrate planning , information exchange and the interactive process of conflict resolution (Stadtler & Kilger 2005; Shapiro 2007; Christopher & Gattorna 2005) despite the known benefits of meta-interventions (Goldratt & Cox 2004; Shapiro 2007; Christopher 2005; Rzevski & Skobelev 2014; Womack 2008).

Three key dimensions are considered:

- The relationships amongst agents of the supply chain
- Agents' transformation of information into decisions
- Information exchange amongst agents

A.3.5.1 Relationships

Decision making in CAS/CES is recognised as a distributed transformation of information which is succeeded by transactions of information between the autonomous decision makers, as events. The chronology of events make the global future state in-computable, irreducible and sensitive to initial parameters (Neumann et al. 1944; Miller & Page 2007; Prokopenko et al. 2009; Rzevski 2011). Despite this view, SCM literature still treats decision making as prescriptive and deterministic. This is reflected in the SCM literature which is dominated by beliefs of planning methods, approaches and paradigms (Stadtler & Kilger 2005; Shapiro 2007; Vollmann et al. 2005), in contrast to meta-interventions which have proven successful across a range of applications (Goldratt & Cox 2004; Womack 2008; Christopher & Gattorna 2005). This has produced several case studies which report that practitioners struggle with representation of relationships with clarity:

- Forecast accuracy and demand variability
- Difficulty/inability to coordinate and synchronize end-to-end Supply chain processes remain
- Lack of visibility across the supply chain
- SC network complexity (perpetual change)
- Lack of internal cross functional collaboration in the supply chain.
- And the fact that SCM does not reflect the demography of businesses[49], as it focuses (75%) on case studies of major corporations, which only account for 1% of the legal entities.

A.3.5.2 Transformation of information

The SCM literatures treatment of the methods for optimisation is also praised for its rapid adoption of optimisation methods from computer science and operations research. However the methods are approached with bias of having perfect information which contradicts basic assumptions of the CAS/CES. In addition none of the discrete event models take notice of the heterogeneity of the computer environment where interaction is bound to have latency and be a major source of requirements for asynchronous information processing.

The critique is also directed towards the bias in favour of cost models, which do not represent the practitioners need consistently from "operational" to "strategic" levels. This (ab?-)usage of optimisation models to identify extrema in

---

[49] Office of national statistics reveal 99% of businesses < 100 employees http://www.ons.gov.uk/ons/rel/bus-register/uk-business/2013/rft---table-1.xls

a solution landscape without parsimonious presentation of the decision makers competitive environment, is open to the same critique as cost-accounting was a decade earlier (Dugdale & Jones 1998). In particular are assumptions about what may justifiably be reduced to make it possible to solve the optimisation problem with the tools which the researcher is trained to use.

This seems to induce issues for SC-modelling as simplification from a CAS/CES perspective is distortion of the model, which is acceptable without greater justification and even for cases where the reduction is to suit the optimisation library at hand and not the scientific objective.

Finally, the terminology such as "strategic" and "advanced" is used consistently to seduce the reader into thinking that a subject is import even though the content of the subject matter is taught on undergraduate courses in math and computer science. Whilst it may be of interest for prominent academics and institutions to preserve the dichotomy in the field, the integrity of the community suffers as more progress otherwise could be made.

A.3.5.3 Information exchange

The treatment of the relationship amongst agents is complimented for its attempt to cover variety and diversity of models for information exchange across (a) networks, (b) transaction models and (c) functional perspective.

The critique that follows is that the models are neither reproducible, well presented nor unbiased (Vidal & Goetschalckx 1997; Swaminathan et al. 1998; Angerhofer & Angelides 2000). The usage of discrete event simulation, agent based models, system dynamics modelling and hybrid systems are poorly represented to other researchers and are cumbersome, if not impossible to extend upon, due to the novelty (or immaturity) of the SCM community to manage large software development projects . In addition several case studies report that practitioners struggle to translate the "scientific" propagation methods used on models into actions at an operational level, as summaries of stochastic properties do not provide insight about the consequence of a given choice (Vidal & Goetschalckx 1997; Shapiro 2007).

A.3.6 The conclusion

To the review question "what is the critique of current intervention methods for SCM" the answer is that the critique is dominated by the following characteristics:

- Overall Theoretical framework is absent (Burgess et al. 2006) though the creativity in the field is impressive (Melo et al. 2009). Sub-problems may be solved for scheduling such as CPM, Gantt, ... but no abstraction unifies the theoretical framework. In fact, correct application of OR methods leave results in conflicting interest and Insufficient level of abstraction of the supply chain problem to model multiple supply chains using the same framework.
- Theoretical approach is inconsistent on its treatment of stochastic and discrete events, as stochastic observations are used with determinism (Swaminathan et al. 1998) resulting in premature commitment of resources.
- Models are made with convenient simplifications, rather than delimitations with evident absence of falsifiable influence of excluded parts (Melo et al. 2009).

- Characteristics of complex systems as defined by Rzevski & Skobelev (2014) is not recognised (Choi et al. 2001; Rzevski & Skobelev 2014), nor is "complexity" used consistently as a term.
- The economic system is not recognised in SCM, made evident as more than 75% of the models only consider costs, profits, revenue or other single variables (Melo et al. 2009; Shapiro 2007) even though the imported methods from CAS/CES contain these features (Tesfatsion 2006; Rzevski & Skobelev 2014)
- Models are functional oriented - despite that SCM claims to be cross functional - and does not pursue holistic optimisation (Shapiro 2007; Badole et al. 2012)
- Choice of terms, such as "strategic" and "advanced" are often exaggerated and/or used without clear definition of what is superior/subordinate (Ramsay & Croom 2008).
- Finally, the most apparent divide in the SCM literature, is the treatment of "planning versus management" which is repeatedly raised in evaluations of IT, financial investments and operational resource allocation. One could argue as if it was "off limits" for research.

The literatures review question *"What is the critique of current intervention methods for SCM?"* may thereby be answered as a set of short-comings, which the thesis should attempt to address. The short comings are:

1. How to reduce the divide between planning and management using a suitable theoretical framework for supply chain management based on agent based modelling?
2. How to evaluate consequences of decisions in real-time for strategic, operational, functional and cross functional perspectives?
3. How to approach optimisation where there is no centralised entity which subordinates all other agents, as such assumption is unrealistic for SCM.
4. How to define an interface to the wider economic system?
5. How to justify reductions of the supply chain model using falsification?
6. How to bridge the gap between stochastic information and decision making for discrete events?
7. How to select appropriate terms for descriptions?

This makes the field a very promising area for theory development.