Simulation-based optimisation of multi-echelon inventory systems

P. Köchel and U. Nieländer
Chemnitz University of Technology
pko@informatik.tu-chemnitz.de
uni@informatik.tu-chemnitz.de

ABSTRACT

Multi-location inventory models are one of the most widely investigated fields in mathematical inventory theory, but the analytically tractable models suffer from various restrictive assumptions. To overcome these restrictions simulation can be used. However, simulation itself is not an optimisation approach. Therefore, we propose the simulation optimisation approach where a simulator is combined with an appropriate optimisation tool.
In the present paper we show that simulation optimisation successfully can be applied to define optimal policies in very general multi-echelon inventory systems. A numerical example demonstrates the usability of our approach.

KEYWORDS

Multi-echelon inventory models, Simulation optimisation, Genetic algorithms

1. Introduction

For today's complex production and distribution systems it becomes more and more important to have efficient and easy applicable tools that model and control the flows of goods through the various locations of the system. One of the top questions is how to guarantee in defined sense optimal inventories. In the past answers were found above all by analytical investigations of multi-location inventory models (MLIM). From the structural viewpoint MLIM can be divided into models with *vertical*, *horizontal*, and *mixed* structure. The first fundamental results on so-called multi-echelon inventory systems, which have a vertical structure, were given by Clark and Scarf in 1960. Clark and Scarf's pioneering work was the starting point for an enormous amount of publications on multi-echelon systems (see e.g. Axsäter 1993 and Federgruen 1993 for a review). The analytical tool mainly used is dynamic programming.
Although the first paper on an MLIM with horizontal structure dates from 1958 (see Allen 1958) the results on such systems up to now are less voluminous. The main reason for that lies in the complexity of an analytical solution, caused by the consideration of later possible transhipments at the ordering moments in the locations. Thus for a long time only single-period models have been considered (Krishnan and Rao 1965, Aggarwal 1967, Köchel 1975). Köchel 1982 used a periodic-stationary Markovian decision model as analysing tool and derived the first results on the dynamic model. A survey on MLIM with horizontal structure can be found in Köchel 1998.
Least of all results are known on MLIM with mixed structure, where the flow of goods through a given number of locations is realised in accordance with a defined predecessor-successor relation and lateral transhipments being allowed between locations. Of course with respect to a somewhat realistic model we cannot expect to get a solution by an analytical approach. Usually simulation will be applied in such a situation. However, simulation is not an optimisation tool by itself. Thus we propose to combine simulation with an appropriate optimisation tool and to derive by such a way solutions for complex control and design problems in MLIM.
The application of simulation optimisation to complex design and control problems started at Chemnitz University of Technology with a problem, where optimal order decisions could be

found for an horizontal structured MLIM with lateral transhipments by coupling simulation with a simple search method (Arnold and Köchel 1996). Later a Genetic algorithm replaced that simple search method. For this aim the optimisation tool *LEO* – *L*aboratory for *E*volutionary *O*ptimisation (Nieländer 1999) was developed. In combination with the simulator *KaSimIR* (*Ka*nban *Sim*ulation *I*maging *R*eality) we used *LEO* for defining optimal numbers and volumes of Kanbans for a Kanban system with mixed structure (Köchel and Nieländer 2002b). The following idea was to modify *KaSimIR* in such a way that it could be used for optimal design and control of MLIM with arbitrary structure.
The aim of the present paper is threefold – to show the broad applicability of the proposed approach, to describe briefly the implemented tools, and to report on first numerical experiments. In Section 2, after introducing a cost structure for the MLIM, we define a relatively general optimisation/control problem. In Section 3 we give a short introduction into our realised simulation optimisation approach. We describe as well the simulator used for the MLIM as the optimisation tool *LEO*. Some numerical examples in Section 4 demonstrate the applicability of the proposed approach. Section 5 concludes the paper with a short summary and an outlook for future research.

2. Problem formulation

Assume a network of M locations, which have to satisfy a demand for a single product. The locations are connected by a given predecessor-successor relation. We call a location without predecessors a *producer*, and a location without successors a *retailer*. Let R denote the number of retailers in the system. The demand D_r arrives to retailer r from outside. Retailer r can backlog demand until a given amount K_r. Demand that cannot be backlogged is lost. In the consequence steady state regime always exists, independently of the applied order policies. Our approach allows policies that can be described by a finite number of parameters. The collection of these parameters over all locations represent the decision variables. Thus it is possible that different locations can apply different policy types. Here we focus on continuous-review, order-point, order-quantity policies. This means that the orderings of location n are controlled by two vectors – the order point vector $s_n = (s_{1n}, s_{2n}, \ldots, s_{Mn})$ and the order quantity vector $Q_n = (Q_{1n}, Q_{2n}, \ldots, Q_{Mn})$. Of cause, s_{in} and Q_{in} make sense if and only if location i is a predecessor of location n. Otherwise we can omit these parameters or we can assign them a dummy value. In the simplest case of a serial system these vectors degenerate to single numbers.
The order lead time consists of the production time by the predecessor plus the transhipment time from the predecessor to the location plus possible delays due to an out-of-stock at the predecessor. To estimate the performance of a given system design we assume the following cost factors: Holding cost h_n and shortage cost p_n at location n per item and per time unit, cost function c_{mn} for order and transhipment from location m to location n, waiting cost w_r per time unit and per backlogged demand unit in retailer r, and rejection cost a_r per rejected demand unit in retailer r. We remark that a retailer r has zero shortage cost p_r but instead waiting cost w_r. The reason for this distinction is that all p_n arise within production itself whereas w_r count for demand.
The inter-arrival times of demand orders at a given retailer are assumed to be independent and identically distributed (iid) random variables. The same we assume for the production times at a location, the transhipment times between two locations, and the delivery times of raw material to a producer. The distributions of these times may be different for different locations.
For the *steady state expected cost* f (per time unit) we have now

$$f(s,Q)=\sum_{m=1}^{M}\left[h_m\cdot H_m(s,Q)+p_m\cdot P_m(s,Q)+\sum_{n=1}^{M}c_{mn}(Q)\cdot T_{mn}(s,Q)\right]+\sum_{r=1}^{R}\left[w_r\cdot W_r(s,Q)+a_r\cdot A_r(s,Q)\right].$$

We remark that the decision variables are the order point matrix $s = (s_{mn})$ and the order lot size matrix $Q = (Q_{mn})$, m, n = 1, 2, …, M. The other terms are *performance measures* of the system – the average inventory on hand $H_m(s, Q)$ in location m, the average number $T_{mn}(s, Q)$ of realised transhipments per time unit between locations m and n, the average amount of shortages $P_m(s, Q)$ in location m, the average queue length $W_r(s, Q)$ of waiting customers, and the average number $A_r(s, Q)$ of rejected demand per time unit as complex functions of s and Q. Thus we have the minimisation problem

$$\text{(P)} \qquad f(s,Q) \quad \rightarrow \quad \underset{(s,Q)}{Min} \; .$$

With respect to problem (P) we want to point to two things. In the case that a reward e_r will be earned for an item sold by retailer r we can still solve problem (P) with modified rejection cost parameters $a'_r = a_r + e_r$. From the relation *throughput* = *demand* – *rejected demand*, i.e., $Th_r(s, Q) = D_r - A_r(s, Q)$ for r = 1, 2, …, R, it follows that the *steady-state expected gain* (per time unit) is equal to the total reward $\sum_{r=1}^{R} e_r \cdot D_r$ minus the modified expected cost. Finally, we can assume that there exists a set X^* as a proper subset of I^{2M^2} – the set of all $2M^2$-dimensional vectors with non-negative integer components – that defines the set of admissible pairs (s, Q). Such a set X^* can be caused by constraints like $K_1 + ... + K_M \leq K$, $H_1(s, Q) + ... + H_M(s, Q) \leq H_{\max}$, $W_r(s, Q) \leq W_{max}$, $E_r(s, Q) \geq E_{min}$ or $A_r(s, Q) \leq A_{max}$. Other constraints may result from technological conditions as for instance necessary proportions between lot sizes.

We further remark, that choosing the constants h_n, p_n, w_r, a_r and function c_{mn} in an appropriate way problem (P) can be reduced to a problem where one of the performance measures has to be optimised. For instance, if $a_r = 1$, $h_n = p_n = w_r = c_{mn} = 0$ then problem (P) is equivalent to the minimisation of the expected number of system-wide lost demand, i.e.,

$$\text{(P-1)} \quad \sum_{r=1}^{R} A_r(s,Q) \rightarrow \underset{k\in X^*}{Min} \; .$$

Other variants for problem (P) are given in Köchel and Nieländer 2002b. However, in all cases the performance measures contained in (P) are complex functions of the decision variables and further system parameters, which are not given in an analytical form. Thus an analytical approach to solve problem (P) is not possible. Therefore we briefly describe in the following section our simulation optimisation approach.

3. The simulation optimisation approach

To find a sufficiently good solution for problem (P), and similar complex optimisation problems as well, we will follow the simulation optimisation approach as outlined in Figure 1. An optimiser gets an optimisation problem as input. After that the search process will be realised by repeated processing of four stages – proposal of a solution, generation of relevant data for the problem by a simulation experiment or evaluation of analytical terms, performance analysis on the basis of those data, decision to accept the proposed solution or to continue the search process. That cycle will be passed through until a stopping criterion will be fulfilled. We remark that once started the search process runs automatically without interaction of the user. After leaving the cycle the best of all considered solutions will be returned. The output of the whole process can be extended, e.g., it can be returned the second best solution and so on. In general we need two things – a simulator and an optimisation tool. We prefer Genetic algorithms as the optimisation tool because they have such advantages like independence of the application domain, suitability for very general optimisation problems, robustness with respect to starting points, they excellently deal with the random output of simulation experiments, they can leave local optima and find the global one, and finally they need only a small amount of input information. For the problem considered here, we used the possibilities of *LEO* (see e.g. Nieländer 1999 for more information). With respect to problem (P) *LEO* needs only pairs (s, Q) and the values $f(s, Q)$ of the criterion function. However,

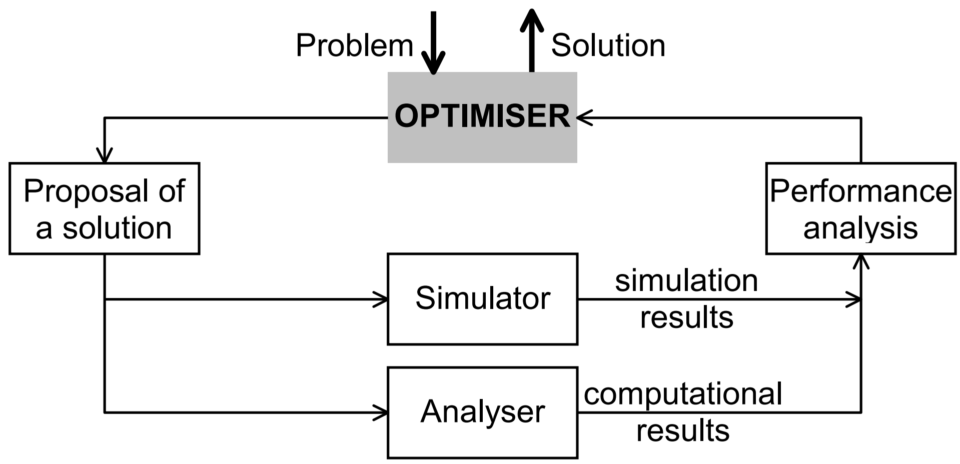

Figure 1. Principle of the used optimisation approach

these values $f(s, Q)$ can only be estimated by a simulation experiment. For that purpose we use the simulator *KaSimIR*, who originally was developed at the Chemnitz University of Technology for the simulation of Kanban systems (Köchel and Nieländer 2002b). Since the concepts of a Kanban system and MLIM are very similar, only a few modifications were necessary. Above all we had to create a possibility to generate and to destroy Kanbans. We regard a Kanban as an order – the volume and the trigger point of the Kanban is equivalent to the order quantity Q and the order point s. Other inventory policies can be realised too, for instance by allowing different numbers of Kanbans to circulate between the locations of the MLIM.

We will finish this section with a short description of those multi-echelon inventory systems, which can be simulated with *KaSimIR* (see also Köchel and Nieländer 2002b):

Number of products:	Single-item or multi-item.
Structure of the system:	Serial, assembly-tree, arbitrary.
Order policy of raw materials:	Common policies of inventory theory.
Delivery of raw materials:	
a) Delivery rate:	Infinite or finite (Poisson or arbitrary process).
b) Lead time:	Zero, constant, random.
Location description:	
a) Inventory selection rules:	FCFS, random, others.
b) Production times:	Constant or random.
Order policy characteristics:	
a) Volume:	One or an arbitrary number of items.
b) When triggered:	State dependent.

c) Transportation time:	Zero, constant, random, equal or different between locations.
Retailers:	Zero, finite or infinite backlogging queue.
Demand characteristics:	
a) Rate:	Infinite or finite (Poisson or arbitrary process).
b) Size:	Single, multiple (constant or random).
Operating regime:	Continuous or discrete, finite or infinite horizon.

An example to get some impressions on our approach is considered in the following section.

4. A numerical Example

In this section we want to demonstrate the applicability of the proposed approach by a numerical example. We consider an echelon inventory system composed of five serial stages (see Fig.2). First results on that system we presented at the 12th International Working Seminar on Production Economics (see Köchel and Nieländer 2002a). Here we report on some empirical investigations of how different order and transhipment cost functions c_{mn} and different backlogging limits K_r influence the optimal policies and the value of the criterion function *f*. We assume a single-item model, Poisson demand with a rate of 0.02 per minute, zero shortage cost at each location, waiting cost of 4 per minute and backlogged demand unit, and rejection cost of 2 000 per rejected demand unit. Further data for all locations are given in Table 1, where *C*, *N* and *U* stand for a constant, a normal and a uniform distribution, respectively.

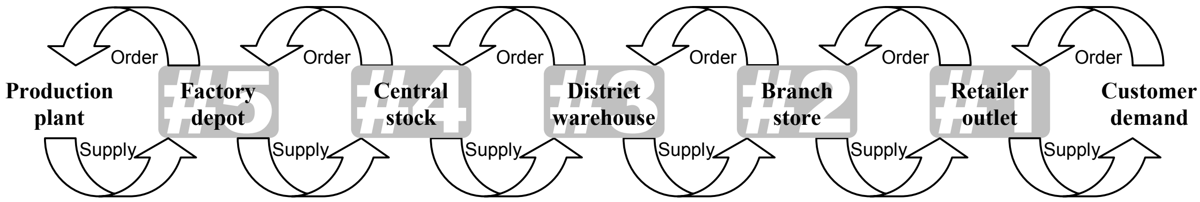

Figure 2. Layout of the exemplary echelon inventory system

Echelon stage	#5	#4	#3	#2	#1
Holding cost per item and minute	1	1	1	1	1
Production time per item in minutes	*N* (10;5)	*U* [30;90]	*N* (40;10)	*C* = 20	*U* [5;15]
Transportation time to stage in minutes	*C* = 10	*C* = 50	*C* = 250	*C* = 100	*C* = 60

Table 1. Further cost and time data for the exemplary echelon inventory system

To keep this numerical example comprehensible, a serial layout was chosen. Although this exemplary system is quite simple, it cannot be solved analytically due to the different distributions for the production times in the locations. For the simulator, it does not matter if we had a more complicated layout within the system to be simulated. The only consequence is that there would be more events (orderings, transhipments, and production of items) taking place during the simulation, resulting in more computational time. For the optimiser, only the number of decision variables but not the layout of the system is relevant.

For this exemplary system we will investigate two continuous-review, order-point, order-quantity strategies (*s, Q*). Applying decentralised control, decisions to order from an upstream installation only depend upon the local inventory position, i.e. the local inventory position computes of the inventory on hand plus outstanding upstream orders minus downstream backorders. With centralised control, end-item demand information is available for decision

making at all stocking points, thus the echelon inventory position computes now of the echelon stock plus outstanding upstream orders, where the echelon stock of some location is the number of items in the system that are at, or have passed through, that location but have as yet not been specifically committed to outside customers.
We will consider 6 cases, which are described in Table 2. For all these cases we assume that $c_{mn}(x) = c'_{mn} + c''_{mn} \cdot x,\ x > 0$. Thus the cases 1 to 3 assume only a set-up part whereas the cases 4 to 6 assume only quantity-dependent cost.

Case	K_1	c'_{65}	c''_{65}	c'_{54}	c''_{54}	c'_{43}	c''_{43}	c'_{32}	c''_{32}	c'_{21}	c''_{21}
1	1	1	0	25	0	200	0	150	0	20	0
2	10	1	0	25	0	200	0	150	0	20	0
3	40	1	0	25	0	200	0	150	0	20	0
4	1	0	1	0	5	0	25	0	10	0	6
5	10	0	1	0	5	0	25	0	10	0	6
6	40	0	1	0	5	0	25	0	10	0	6

Table 2. Investigated cases for the exemplary echelon inventory system

Our objective is to find optimal order-points *s* and order-quantities *Q* for all the stages with respect to minimise the total cost per year. For each configuration to be tested, the total cost per year were estimated by three simulation runs. Each run consists of a transition phase and of 700 days of simulated real time. During that time approximately 20 000 demand events were simulated. Then all data were transformed to a one-year time period. The optimisation process by the Genetic algorithm was stopped after considering 5 000 solutions. The tables with the results contain also the estimated values for different cost parts. This is an additional advantage of simulation – we can investigate the changes of various performance measures inside of a single simulation experiment. The 95%–confidence intervals stated in these tables are based on the assumption of a log-normal distribution of the different cost parts collected in the three simulation runs carried out for each configuration. We also tested some example with up to 51 simulation runs, and it showed that having only three runs yields in reliable results too. This is due to the fact that our simulation runs are comparatively long.
Let us consider now the cases 1 to 3, i.e., the cases with set-up but no quantity-dependent transportation cost (Tables 3 to 5).

Control strategy	**Decentralised control**					**vs.**	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \le s \le 20$	**0**	**0**	**6**	**2**	**5**		**8**	**12**	**19**	**13**	**15**
Order quantity $1 \le Q \le 100$	**1**	**2**	**2**	**2**	**1**		**1**	**1**	**2**	**2**	**1**

	Decentralised control	vs.	Centralised control
Total cost per year	**14 343 178.902** $^{-1.5\%}_{+1.6\%}$	>	**14 071 672.218** $^{-2.6\%}_{+2.9\%}$
Waiting cost	226 813.397 $^{-10.2\%}_{+15.3\%}$	<	272 999.563 $^{-6.9\%}_{+9.1\%}$
Rejection cost	5 302 603.200 $^{-3.5\%}_{+4.1\%}$	<	5 629 176.000 $^{-5.8\%}_{+7.3\%}$
Transportation cost	1 610 980.644 $^{-0.4\%}_{+0.4\%}$	<	1 664 852.366 $^{-1.2\%}_{+1.3\%}$
Holding cost	7 202 781.661 $^{-0.0\%}_{+0.0\%}$	>>	6 504 644.289 $^{-0.0\%}_{+0.0\%}$
Served customer orders	7 726.495 $^{-0.4\%}_{+0.4\%}$	>	7 534.125 $^{-1.2\%}_{+1.3\%}$
Rejected customer orders	2 651.302 $^{-3.5\%}_{+4.1\%}$	<	2 814.588 $^{-5.8\%}_{+7.3\%}$
Configuration found	2 823-th of 5 000		4 976-th of 5 000

Table 3. Optimisation results for the case 1 (Set-up transportation cost and K_1 = 1)

Control strategy	**Decentralised control**					**vs.**	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \le s \le 20$	**0**	**0**	**0**	**3**	**10**		**5**	**16**	**18**	**11**	**10**
Order quantity $1 \le Q \le 100$	**2**	**2**	**2**	**2**	**10**		**2**	**2**	**2**	**3**	**12**

	Decentralised control	vs.	Centralised control
Total cost per year	**28 587 318.703** $^{-1.0\%}_{+1.1\%}$	>	**26 825 367.642** $^{-2.9\%}_{+3.3\%}$
Waiting cost	11 076 231.680 $^{-0.6\%}_{+0.7\%}$	>	10 335 030.711 $^{-2.4\%}_{+2.6\%}$
Rejection cost	3 715 291.200 $^{-7.7\%}_{+10.5\%}$	<	3 800 788.800 $^{-12.2\%}_{+19.9\%}$
Transportation cost	1 625 502.096 $^{-0.6\%}_{+0.6\%}$	>>	1 394 365.291 $^{-0.4\%}_{+0.5\%}$
Holding cost	12 170 293.726 $^{-0.2\%}_{+0.2\%}$	>	11 295 182.839 $^{-0.5\%}_{+0.5\%}$
Served customer orders	8 557.644 $^{-0.6\%}_{+0.6\%}$	>	8 470.920 $^{-0.4\%}_{+0.4\%}$
Rejected customer orders	1 857.645 $^{-7.7\%}_{+10.5\%}$	<	1 900.394 $^{-12.2\%}_{+19.9\%}$
Configuration found	4 601-th of 5 000		4 736-th of 5 000

Table 4. Optimisation results for case 2 (Set-up transportation cost and K_1 = 10)

Control strategy	**Decentralised control**					**vs.**	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \le s \le 20$	**0**	**0**	**8**	**4**	**8**		**18**	**20**	**20**	**18**	**9**
Order quantity $1 \le Q \le 100$	**2**	**2**	**2**	**2**	**50**		**5**	**5**	**2**	**2**	**48**

	Decentralised control	vs.	Centralised control
Total cost per year	**86 492 105.309** $^{-0.9\%}_{+0.9\%}$	>	**78 600 547.636** $^{-2.1\%}_{+2.3\%}$
Waiting cost	47 836 825.681 $^{-1.5\%}_{+1.6\%}$	<	49 320 422.749 $^{-2.2\%}_{+2.4\%}$
Rejection cost	3 764 347.200 $^{-6.6\%}_{+8.6\%}$	~	3 782 918.400 $^{-8.5\%}_{+12.0\%}$
Transportation cost	1 603 106.280 $^{-0.2\%}_{+0.2\%}$	>	1 536 598.258 $^{-0.6\%}_{+0.6\%}$
Holding cost	33 287 826.148 $^{-0.3\%}_{+0.3\%}$	>>	23 960 608.229 $^{-1.1\%}_{+1.2\%}$
Served customer orders	8 521.903 $^{-0.4\%}_{+0.4\%}$	~	8 518.925 $^{-0.4\%}_{+0.4\%}$
Rejected customer orders	1 882.174 $^{-6.6\%}_{+8.6\%}$	~	1 891.459 $^{-8.5\%}_{+12.0\%}$
Configuration found	4 737-th of 5 000		3 122-th of 5 000

Table 5. Optimisation results for case 3 (Set-up transportation cost and K_1 = 40)

At least three conclusions can be made from the results in Table 3, 4 and 5.

Conclusion 1.
The centralised control outperforms the decentralised control, whereas the difference of the total cost per year between the decentralised and centralised control will increase with increasing backlogging limit.

Conclusion 2.
The increase of the backlogging limit has the consequences that

a) the total cost per year increase (above all by increased waiting and holding cost);
b) the rejection cost decrease;

c) the transportation cost remains more or less constant.

Conclusion 3.
Above a finite backlogging limit the average number of served and rejected demand orders is independent from a further increase of that limit.

Let us consider now the cases 4 to 6, where we have quantity-dependent transportation cost. Table 6 shows an unexpected result – the decentralised policy outperforms the centralised one. Thus Conclusion 1 holds for the cases 5 and 6 only. Conclusion 2 and Conclusion 3 are valid on the whole.

Control strategy	**Decentralised control**					vs.	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \leq s \leq 20$	**0**	**0**	**2**	**5**	**4**		**4**	**11**	**16**	**10**	**10**
Order quantity $1 \leq Q \leq 100$	**1**	**1**	**1**	**1**	**1**		**4**	**4**	**1**	**1**	**1**

Control strategy	**Decentralised control**	vs.	**Centralised control**
Total cost per year	**12 153 265.161** $^{-3.2\%}_{+3.6\%}$	**! < !**	**12 774 886.663** $^{-1.7\%}_{+1.8\%}$
Waiting cost	263 467.891 $^{-8.3\%}_{+11.6\%}$	>	237 936.407 $^{-4.8\%}_{+5.9\%}$
Rejection cost	5 610 955.200 $^{-6.2\%}_{+8.0\%}$	>	5 331 686.400 $^{-4.7\%}_{+5.8\%}$
Transportation cost	355 668.264 $^{-1.2\%}_{+1.3\%}$	<	362 583.583 $^{-1.0\%}_{+1.0\%}$
Holding cost	5 923 173.806 $^{-0.1\%}_{+0.1\%}$	<<	6 842 680.273 $^{-0.8\%}_{+0.8\%}$
Served customer orders	7 568.115 $^{-1.3\%}_{+1.3\%}$	<	7 715.107 $^{-1.0\%}_{+1.0\%}$
Rejected customer orders	2 805.478 $^{-6.2\%}_{+8.0\%}$	>	2 665.843 $^{-4.7\%}_{+5.8\%}$
Configuration found	4 607-th of 5 000		4 736-th of 5 000

Table 6. Optimisation results for case 4 (No set-up transportation cost and $K_1 = 1$)

Control strategy	**Decentralised control**					vs.	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \leq s \leq 20$	**0**	**1**	**3**	**1**	**6**		**8**	**14**	**17**	**13**	**15**
Order quantity $1 \leq Q \leq 100$	**1**	**1**	**1**	**1**	**8**		**3**	**3**	**1**	**1**	**12**

Control strategy	**Decentralised control**	vs.	**Centralised control**
Total cost per year	**26 928 533.841** $^{-1.5\%}_{+1.6\%}$	**>**	**25 314 102.508** $^{-1.5\%}_{+1.6\%}$
Waiting cost	12 308 801.564 $^{-0.2\%}_{+0.2\%}$	>>	10 479 299.396 $^{-2.3\%}_{+2.5\%}$
Rejection cost	3 767 500.800 $^{-9.9\%}_{+14.8\%}$	<	3 923 078.400 $^{-4.7\%}_{+5.7\%}$
Transportation cost	401 508.117 $^{-0.8\%}_{+0.9\%}$	>	395 802.554 $^{-0.6\%}_{+0.7\%}$
Holding cost	10 450 723.359 $^{-0.2\%}_{+0.2\%}$	~	10 515 922.157 $^{-0.5\%}_{+0.5\%}$
Served customer orders	8 545.555 $^{-0.8\%}_{+0.9\%}$	>	8 424.142 $^{-0.7\%}_{+0.7\%}$
Rejected customer orders	1 883.750 $^{-9.9\%}_{+14.8\%}$	<	1 961.539 $^{-4.7\%}_{+5.7\%}$
Configuration found	3 845-th of 5 000		4 960-th of 5 000

Table 7. Optimisation results for case 5 (No set-up transportation cost and $K_1 = 10$)

Control strategy	**Decentralised control**					**vs.**	**Centralised control**				
Echelon stage	#5	#4	#3	#2	#1		#5	#4	#3	#2	#1
Order point $0 \leq s \leq 20$	**0**	**1**	**9**	**3**	**13**		**2**	**14**	**13**	**8**	**16**
Order quantity $1 \leq Q \leq 100$	**1**	**1**	**1**	**1**	**55**		**2**	**2**	**1**	**1**	**43**
Total cost per year	**84 536 866.552 − 0.3% + 0.3%**					>>	**77 741 579.897 − 1.0% + 1.0%**				
Waiting cost	43 899 450.059 − 0.6% + 0.6%					<<	52 658 372.730 − 1.4% + 1.5%				
Rejection cost	3 456 696.000 − 8.5% +12.0%					<	3 835 828.800 − 3.7% + 4.3%				
Transportation cost	407 461.764 − 1.3% + 1.3%					>	397 000.747 − 0.6% + 0.6%				
Holding cost	36 773 258.729 − 0.5% + 0.5%					>>	20 850 377.619 − 0.9% + 0.9%				
Served customer orders	8 682.036 − 1.3% + 1.3%					>	8 452.699 − 0.7% + 0.7%				
Rejected customer orders	1 728.348 − 8.5% +12.0%					<	1 917.914 − 3.7% + 4.3%				
Configuration found	3 706-th of 5 000						4 736-th of 5 000				

Table 8. Optimisation results for case 6 (No set-up transportation cost and K_1 = 40)

We remark that there exist more than $4 \cdot 10^{16}$ possible configurations of the exemplary system, whereas an optimisation run stopped after testing only 5 000 of them.

5. Summary and outlook

In the present paper, we described how to use the simulation optimisation approach for solving very complex control problems. The proposed approach was applied to multi-echelon inventory systems, which differ from the usually investigated ones above all by highly general assumptions with respect to the systems structure, the demand processes, the cost and gain functions, and the applicable order policies. Of course the results presented here can be only a starting point for further research with different investigation goals.

Future research can go into miscellaneous directions, i.e., at least into the operations research direction or into the computer science direction. With respect to the first variant it seems to be promising to investigate the influence of various model parameters on interesting performance measures. This in general can be done only by empirical work. The consideration of other policies will be interesting. The same will hold if we try for instance to find conditions under which a given policy class dominates other ones.

On the other hand, since the proposed approach needs a great amount of computing time it is necessary to apply principles of distributed and parallel computing both to the simulation and the optimisation part.

References

Aggarwal, S.P. (1967). Inventory Control Aspects in Warehouses. *Symposium on Operations Research*, Indian National Science Academy, New Delhi.

Allen, S.G. (1958). Redistribution of Total Stock over Several User Locations. *Naval Research Logistics Quarterly*, Vol. 5, pp. 337–345.

Arnold, J.; Köchel, P. (1996). Evolutionary Optimization of a Multi-Location Inventory Model with Lateral Transhipments. *Pre-Prints of the Ninth International Working Seminar on Production Economics*. Kongresszentrum Igls, Innsbruck, Vol. 2, pp. 401–412.

Axsäter, S. (1993). Continuous Review Policies for Multi-Level Inventory Systems with Stochastic Demand. In Graves, S.C. et al. (eds). *Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory.* North Holland, Amsterdam, pp. 175–197.

Clark, A.J.; Scarf, H.E. (1960), Optimal Policies for a Multi-Echelon Inventory Problem. *Management Science*, Vol. 6, No. 4, pp. 475–490.

Federgruen, A. (1993). Centralized Planning Models for Multi-Echelon Inventory Systems under Uncertainty. In Graves, S.C. et al. (eds). *Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory.* North Holland, Amsterdam, pp. 133–174.

Köchel, P. (1975). A Stochastic Inventory Model for some Interconnected Locations [in German]. *Mathematische Operationsforschung und Statistik*, Series Optimization, Vol. 6, No. 3, pp. 413–426.

Köchel, P. (1982). A Dynamic Multi-Location Inventory Model with Transhipments between Locations [in German]. *Mathematische Operationsforschung und Statistik*, Series Optimization, Vol. 13, No. 2, pp. 267–286.

Köchel, P. (1998). A Survey on Multi-Location Inventory Models with Lateral Transhipments. In Ganas, I.; Papachristos, S. (eds). *Proceedings of the Third ISIR Summer School on Inventory Modelling.* University of Ioannina, Department of Mathematics, pp. 183–207.

Köchel, P.; Nieländer, U. (2002a). Defining Optimal Policies in Multi-Echelon Inventory Systems: The Simulation Optimisation Approach. *Pre-Prints of the Twelfth International Working Seminar on Production Economics.* Kongresszentrum Igls, Innsbruck, Vol. 1, pp. 165–173.

Köchel, P.; Nieländer, U. (2002b). Kanban Optimization by Simulation and Evolution. *Production Planning & Control*, Vol. 13, No. 8, pp. 725–734.

Krishnan, K.S.; Rao V.R.K. (1965). Inventory Control in *N* Warehouses. *Journal of Industrial Engineering*, Vol. 16, No. 3, pp. 212–215.

Nieländer, U. (1999). Simulation Optimization of Kanban Systems Using a Non-Standard Genetic Algorithm. In Hill, R.; Smith, D. (eds). *Proceedings of the Fourth ISIR Summer School on Inventory Modelling.* University of Exeter, School of Mathematical Sciences, pp. 137–146.