OPTIMAL CONTROL AND DESIGN OF COMPLEX SYSTEMS BY SIMULATION AND GENETIC ALGORITHMS

Peter Köchel and Ulf Nieländer
Department of Computer Science
Chemnitz University of Technology
D–09107 Chemnitz
Germany
E-mail: pko@informatik.tu-chemnitz.de
uni@informatik.tu-chemnitz.de

KEYWORDS

Simulation optimisation, Genetic algorithms, Kanban systems, Inventory models

ABSTRACT

In operations research numerous approaches and algorithms exist to solve design and control problems for systems of such different areas like inventories, logistics, transportation, manufacturing etc. Nevertheless, the complexity of real-world systems prevents the application of almost all classical approaches. One method to overcome these difficulties is simulation optimisation where a simulator for the considered system is combined with an appropriate optimisation tool.

In our presentation we suggest to combine simulation with the genetic optimisation tool LEO. We briefly discuss the application of that software tool to find optimal order policies for multi-location inventory models and to design an optimal Kanban controlled manufacturing system. Finally, we report on some experiences and further developments.

INTRODUCTION

One main topic in operations research is the optimal control and / or design of stochastic systems. Such systems may come from various application areas like inventories, logistics, manufacturing etc. However, the complexity of real-world systems prevents the application of almost all of the classical solution approaches. One method to overcome these difficulties is to combine a simulator for the system to be considered with an appropriate optimisation tool (simulation optimisation). The basic idea of such a combination is very simple. The results of a simulation experiment will be used to estimate performance measures that relevant for the problem. Next, on the basis of such estimations one has to answer two questions: To stop or to continue searching for a good solution, and, if to continue, how to find an improved solution. Whereas the first question can be answered by defining various stopping criteria, the second question is the crucial one.

The present paper is dealing with the solution of complex optimisation problems, which are actually for multi-location systems and for Kanban systems. Our approach combines simulation with the non-standard genetic optimisation tool LEO (Laboratory of Evolutionary Optimisation), developed at Chemnitz University of Technology. Such a combined implementation has at least three advantages. First, the searching process for a good solution can be realised without intermediate actions of the designer. Second, by defining suitable interfaces in fact arbitrary simulators and optimisers can be connected. Third, parallel processing can be realised either for the simulation or for the genetic algorithm as well as for both.

The paper is organised as follows. In the next section we give a brief introduction into the simulation optimisation approach. Then that approach is applied to solve the optimal design problem for Kanban systems followed by the application to the optimal control problem of Multi-location inventory models. Finally, we report on some experiences and further developments.

THE SIMULATION OPTIMISATION APPROACH

To find a sufficiently good solution for the above-mentioned complex optimisation problems we will follow the simulation optimisation approach as outlined in figure 1. An optimiser gets an optimisation problem as input. Now the following cycle will implement the search process – proposal of a solution, realisation of a simulation experiment and accumulation of data that are relevant for the problem, performance analysis on the basis of these data, decision to accept the proposed solution or not. That cycle will be repeated until a stopping criterion is fulfilled.

We remark that once started, the search process runs automatically without interaction of the user. After stopping the search process, the best solution of all up to now considered ones will be returned. We remark that the output of the whole process can be more extensive. Thus it is possible to return the second best solution and further information on the optimisation process.

Proceedings 14th European Simulation Symposium
A. Verbraeck, W. Krug, eds. (c) SCS Europe BVBA, 2002

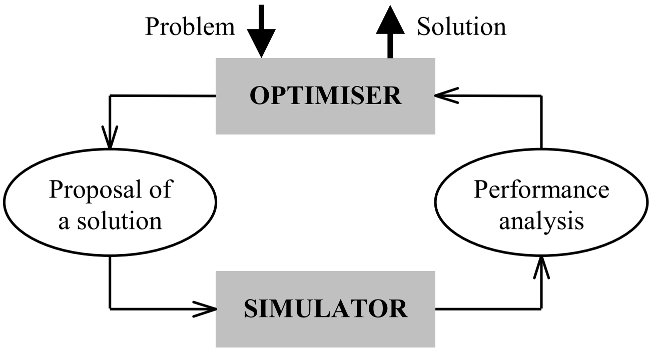

Figure 1: Scheme of simulation optimisation

To apply simulation optimisation in the described form in general, we need only two things – a simulator and an optimisation tool. For the latter, we prefer genetic algorithms (GA), since they possess several advantages – independence of the application domain, suitability for very general optimisation problems, robustness with respect to starting points. Furthermore, they excellently deal with the random output of simulation experiments, they leave local optima and find the global one, and finally they need only a small amount of input information. For problems of the here-considered type we used the possibilities of *LEO* – *L*aboratory of *E*volutionary *O*ptimisation. See e.g. (Nieländer 1999) for more information.

OPTIMAL DESIGN OF KANBAN SYSTEMS

To implement the Just-In-Time (JIT) idea in production, logistics or supply chain systems efficient control mechanisms such as Kanban are necessary. To explain that Kanban control mechanism we use a single-item, multi-stage, serial production system as shown in figure 2.

Since items will be produced and moved through the system in lots with corresponding lot sizes, containers with given capacity are used. Stage *n*, which is represented in figure 2 in more detail, owns a finite number of Kanbans (cards). The Kanbans are collected in the Kanban box KB. If there is at least one Kanban in that box, and if there is at least one full container in the output hopper OH of stage *n*–1, then one of those containers is moved to stage *n*. Here that container is tied with one Kanban from the Kanban box of stage *n*. Now the pair (container, Kanban of stage *n*) moves to buffer B. In case the server S is busy the pair has to wait, otherwise processing starts immediately. After processing, the pair goes to the output hopper OH where it is waiting for withdrawal by the succeeding stage *n*+1. Withdrawal takes place as soon as in the Kanban box of the succeeding stage *n*+1 there is a Kanban. Then the pair (container, Kanban of stage *n*) is separated, the Kanban of stage *n* is returned to its box, and the container is moved to stage *n*+1. To understand that control principle of Kanban systems, it is not necessary to explain the kind of Kanbans used. Throughout this section we consider one-card systems only, and we remark that in modern information systems a Kanban may be represented by a corresponding signal.

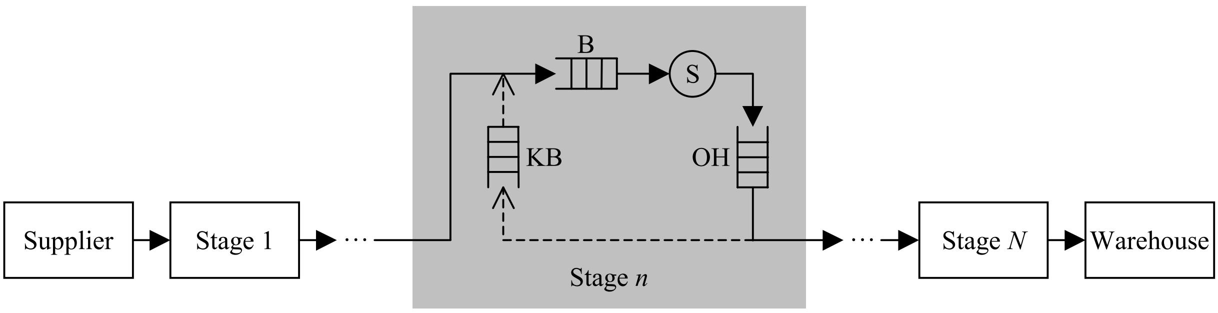

Figure 2: Flow of Kanbans (- - - ->) and containers (——►)

In the context of the present paper we are interested in solutions to the *optimal design problem* of the Kanban system in order to maximise the *steady-state expected gain* (per time unit). For that reason, we assume *gain and cost factors* denoting the selling reward of one item final product, the holding cost of one unit Work-In-Process (WIP) in stage *n* per time unit for every single item for the period from entering stage *n* to entering the succeeding stage, transportation cost for the transport of one container from stage *m* to stage *n*, shortage cost, waiting cost per time unit and per backlogged demand unit, and rejection cost per rejected demand unit. The decision variables in our model represent the *number* and the *volume* of the Kanbans / containers in the stages of the system. We remark that choosing the gain and cost factors in an appropriate way, the optimal design problem can be reduced to a problem where one of the classical, important steady-state performance measures such as throughput, average WIP, average flow time of items, or average queue length of waiting customers has to be optimised. Now the *Kanban allocation problem* can be verbally formulated as 'the problem to allocate a given total number of Kanbans among the stages of a multi-stage system such that a given criterion will be optimised'. To apply the scheme of figure 1 to the optimal design problem of Kanban systems we had to implement a simulator KaSimIR (see Köchel et al. 2002 for more information) for such systems, and currently a new version of KaSimIR is being developed (see next section for some remarks).

Let us now report on results for two examples that are based on the following manufacturing system (see figure 3). It was chosen to demonstrate that our approach yields appropriate solutions for sufficiently complex systems, and the examples shall underline our point of view that for an optimal system design various cost factors should be taken into account. For that reason we define a solution for the Kanban allocation problem, where the throughput is maximised under the assumption of constant volumes equal to one. This solution is compared to the solution for the optimal design total cost minimising problem.

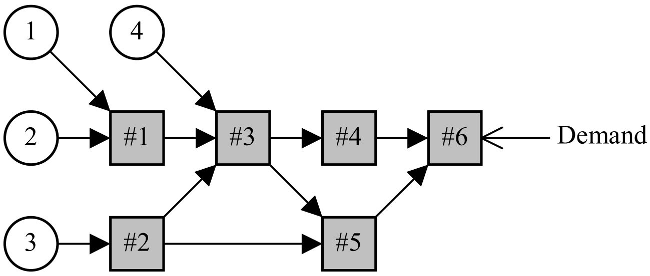

Figure 3: Layout of the exemplary manufacturing system

The exemplary manufacturing system has four external suppliers 1 to 4 of raw material and six manufacturing stages #1 to #6, the latter one being the warehouse. The transport of raw material from outside suppliers to the factory is rather expensive and time consuming, whereas transportation within the factory is cheaper and faster. Note that all transportation times are constant throughout the examples, and we use minutes as time unit. Demand orders of one unit each arrive according to a Poisson process with a rate of 0.01 per minute and max 10 demand orders can be backlogged in the waiting queue with waiting costs of 1 monetary unit per minute and backlogged demand order. Rejection costs of 20 monetary units result from each demand order that meets a full waiting queue, since that order is rejected and lost. We assume no selling reward, i.e. we are interested in the minimising the steady-state total costs of the system. All shortage costs are 0, and the following table contains further cost and time data for all the stages to finally specify the exemplary system. We remark that N stands for a normal distribution, whereas U indicates a uniform distribution.

Table 1: Further cost and time data for the stages of the exemplary manufacturing system

Manufacturing stage n	#1		#2	#3			#4	#5		#6	
Service time per item in stage n	N (5;2)		N (5;1)	U [2;6]			N (10;4)	U [6;12]		N (4;1)	
Holding cost in stage n	2		2	3			4	5		7	
Supplier m	1	2	3	4	#1	#2	#3	#2	#3	#4	#5
Holding cost for supplier m	1	1	1	1							
Transportation cost from m to n	1 000	1 000	2 000	500	20	20	20	20	20	20	20
Transportation time from m to n	240	240	300	120	5	10	2	15	10	5	5

Firstly, we examine the Kanban allocation problem, where all costs are set to be zero and the volumes of the Kanbans / containers in the stages of the system are $v_1 = \ldots = v_N = 1$. A total of 33 Kanbans are to be allocated, and the goal is to maximise the system's throughput. The simulation experiments consisted of three runs that each had a transition phase of 2.85 years followed by 190.25 years simulated real time with about 1 000 000 demand unit orders. Our simulation optimisation solution is $k^{(1)} = (6, 6, 11, 3, 1, 1, 1, 1, 1, 1, 1)$ as the number of Kanbans / containers in the stages of the system (from the left to the right in the data table 1), and it was found as 840th evaluated decision with an estimated throughput of 14.376 units per day. Simulating this solution ($k^{(2)}$, (1, …, 1)) with the costs according to table 1, an estimated total of 221 019.273 monetary units arise per day as sum of 1 441.466 waiting costs, 0.601 rejection costs, 160 726.239 transportation costs, and 58 850.967 holding costs respectively.

Secondly, we are interested in the solution for the optimal design total cost minimising problem for this example with the additional constraints $k_n \leq 11$ and $v_n \leq 11\ \forall\ n = 1\ (1)\ N$ respectively. The simulation experiments again consisted of three runs that each had a transition phase of more than 25 years followed by 190.25 years simulated real time with about 1 000 000 demand unit orders. Our simulation optimisation solution is $k^{(2)} = (1, 1, 1, 1, 1, 1, 5, 2, 5, 1, 4)$ and $v^{(2)} = (2, 2, 1, 2, 1, 2, 1, 1, 1, 1, 1)$ as the number and the volume of the Kanbans / containers in the stages of the system (from the left to the right in the data table 1), and it was found as 7 127th evaluated decision with estimated total costs of 42 032.348 monetary units per day as sum of 14 233.839 waiting costs, 256.079 rejection costs, 13 855.987 transportation costs, and 13 686.443 holding costs respectively.

Comparing this solution ($k^{(2)}, v^{(2)}$) with the one ($k^{(1)}$, (1, …, 1)) above we remark that the estimated total costs decreased by more than 80 %, whereas the estimated system's throughput decreased by nearly 90 %. Of course this result seems to be contra productive – the total cost minimising solution results in only 10 % of the throughput of the throughput maximising solution. But this is the consequence of the chosen cost parameters in table 1, where the cost for transportation from the suppliers are very high and in fact dominate the total cost. Thus, one should consider both, the total costs and the system's throughput when designing a real system.

We remark that in (Köchel and Nieländer 2002b), further examples can be found that have a more general structure, i.e. the number of used parts per manufactured item is not just equal to one. Thus, one item of the final product no longer consists of two units of raw material from the suppliers 1, 2, and 4 respectively, and three units of raw material from supplier 3.

OPTIMAL CONTROL OF MULTI-LOCATION INVENTORY MODELS

One of the main topics in mathematical inventory theory is the definition of optimal order and transhipment policies in multi-location inventory models (MLIM). Up to now, the multitude of the investigated MLIM's can be divided into models with *vertical*, *horizontal*, and *mixed* structure. Broadest investigated are models with a vertical structure, the multi-echelon models. For a review on these models see Chapters 3 and 4 in (Graves et al. 1993).

Although the first paper on MLIM with horizontal structure is about 50 years old, up to now the analytical results on such systems are less voluminous. The main reason for that is due to the complexity of the problem caused by the consideration of later possible transhipments at the ordering moments in the locations. Thus for a long time only single-period models were considered. (Köchel 1982) was the first who derived results on the dynamic model. A survey on MLIM with horizontal structure can be found in (Köchel 1998). Since we cannot expect to get analytical solutions for real-world MLIM's we started at Chemnitz University of Technology to apply the simulation optimisation approach to a horizontal structured MLIM with lateral transhipments (Arnold and Köchel 1996). Last of all, results are known on MLIM's with mixed structure, where the flow of goods through a given number of locations is realised according to a defined predecessor–successor relation and where lateral transhipments are allowed between locations. To investigate those models the simple search method in (Arnold and Köchel 1996) is replaced by a GA in (Köchel and Nieländer 2002a).

To give an impression of how to realise simulation optimisation for an MLIM we describe the essential things. For this we assume an exemplary echelon system composed of N serial stages. The whole system has to satisfy a random demand for a single product. Stage N is the producer, and stage 1 the retailer who meets the demand. The intermediate stages represent various storages. Orders can be made to the previous stage only. The optimisation goal is to define such order policies that minimise the expected average cost of the whole system. Costs arise, among others, from an order of product as well as from transportation, holding inventory, rejection of demand, and shortage of product. To apply a GA we parameterise the set of admissible policies. This can be done by considering for instance so-called (s, Q) order policies (we order the quantity Q if and only if the inventory position at a location dropped below s). In that case the decision vector for the N-stage echelon problem consists of $2{\cdot}N$ variables s_i and Q_i for i from 1 to N. The cyclic search process according to the scheme of figure 1 works as follows. To define a solution for each of the N stages of the echelon model we choose a real number s and a positive real number Q. These parameter values are given to the simulator who realises a simulation experiment with corresponding output data. These data are used to compute sample averages for the expected average cost and other performance measures needed. The resulting values are given back to the GA which – on the base of that information – either stops the search process or defines new solutions.

The use of simulation optimisation to a MLIM has some important advantages. First of all, the application variety depends only on the available MLIM-simulator. At present we are implementing a simulator, which allows to simulate very general MLIM's including the multi-item case, arbitrary structures, different inventory policies, arbitrary delivery and demand processes, random lead and transportation times and others. A second advantage is that we can optimise with respect to various goals (expected cost, average cost, service levels) without and with restrictions. Finally, we remark that from the simulation data we can compute estimations for the separate parts of the goal function, and thus we are able to investigate the influence of several decision variables on different cost parts.

More details and first numerical results can be found in (Köchel and Nieländer 2002a). Future work will concentrate on a more empirical research to answer such essential questions like "Which first and second order properties hold for a given performance measure?" or "Under which conditions a given set of policies dominates another one?"

Finally, we want to point to an important similarity between Kanban systems and MLIM's. By defining numbers, volumes, and trigger points of Kanbans in a suitable way we can realise various order policies for a MLIM. That circumstance will be used in the new version of KaSimIR, which allows to model as well Kanban systems as MLIM's.

CONCLUSION

In the past we applied the simulation optimisation approach to complex control and design problems. Here we briefly reported on applications to inventory problems as well as to manufacturing systems. An other problem, which is important for practice, is investigated in (Köchel et al. 2002). In all cases we found solutions whose performance was better than the performance of before known solutions. Moreover, since our approach is based on a simulator for the system to be investigated we could solve problems in a very general formulation. From our collected experiences with the simulation optimisation approach we can deduce some generally valid consequences.

Consequences with respect to the approach:

1. If the interface between the optimiser and the simulator is defined in an appropriate way then as well the same optimiser can handle problems from various application areas as different optimisers can solve a given problem.

2. Clearly, the simulator consumes most of the computing time. Thus a parallel simulator seems to be advantageous. However the considered problems resp. systems are dealing with a set of very interdependent elements. Thus a parallel simulator does not make sense. We expect more from the parallelisation of as well the simulation experiment as the Genetic Algorithm.

3. Analytically proved properties of the optimal policies do not decrease the solution time considerably but they improve the solution quality.

Consequences with respect to the application areas:

1. For all problems where we have to control some common resources there exist a lot of solutions, which criterion value is in a neighbourhood of the optimal value.

2. Existing analytical or approximation algorithms can lead to very bad solutions in situations, where corresponding assumptions are not fulfilled.

3. The approach allows to handle more realistic systems or problems.

4. The approach allows a holistic investigation of systems.

REFERENCES

Arnold, J.; Köchel, P. 1996. "Evolutionary Optimization of a Multi-location Inventory Model with Lateral Transshipments". In *9th Internat. Working Seminar on Prod. Econ.,* Igls 1996, Pre-Prints v.2, 401-412

Graves, S.; Kan, A.R.; P. Zipkin (Eds.). 1993. *Handbook in Operations Research and Management Science, Vol. 4, Logistics of Production and Inventory.* North Holland

Köchel, P. 1982. "A dynamic Multi-location inventory model with transhipments between locations". (In German) *Math. Operationsforsch. Stat., Ser. Opt.* 13, 267-286

Köchel, P. 1998. „A Survey on Multi-location Inventory Models with Lateral Transshipments". In *Inventory Modelling in Production and Supply Chains,* S. Papachristos and I. Ganas (Eds.). 3rd ISIR Summer School, Ioannina, Greece,183-207

Köchel, P.; Kunze, S.; Nieländer, U. 2002. "Optimal control of a distributed service system with moving resources: Application to the fleet sizing and allocation problem." Forthcoming in *Int. J. Production Economics* 78

Köchel, P.; Nieländer, U. 2002a. „Defining optimal policies in multi-echelon inventory systems: The simulation optimisation approach". In *12th Intern Working Seminar on Prod. Econ*., Igls 2002, Pre-Prints v.1, 165-173

Köchel, P.; Nieländer, U. 2002b. "Kanban optimisation by simulation and evolution". Forthcoming in *Production Planning & Control*.

Köchel, P.; Nieländer, U.; Sturm, M. 2002. KaSimIR – object-oriented Kanban Simulation Imaging Reality. Chemnitzer Informatikberichte CSR-01-03, Chemnitz University of Technology, Chemnitz, Germany.

Nieländer, U. 1999. "Simulation Optimisation of Kanban Systems using a non-standard Genetic Algorithm." In *4th ISIR Research Summer School on Inventory Modelling, Exeter 1999, UK,* 137-146.