Les cahiers Leibniz

Issues in the complementary use of simulation and optimization modeling

Anne-Laure Ladier, Allen G. Greenwood, Gülgün Alpan, Halston Hales

Laboratoire G-SCOP
46 av. Félix Viallet, 38000 GRENOBLE, France
ISSN : 1298-020X

n° 211

January 2014

Site internet : http://www.g-scop.inpg.fr/CahiersLeibniz/

ISSUES IN THE COMPLEMENTARY USE OF SIMULATION AND OPTIMIZATION MODELING

Anne-Laure Ladier

Grenoble-INP / UJF-Grenoble 1 / CNRS G-SCOP
UMR5272 Grenoble, F-38031
46 av. Félix Viallet, 38000 Grenoble, France
anne-laure.ladier@g-scop.grenoble-inp.fr

Allen G. Greenwood

Department of Industrial and Systems Engineering
Mississippi State University
Mississippi State, MS 39762, USA
greenwood@ise.msstate.edu

Gülgün Alpan

Grenoble-INP / UJF-Grenoble 1 / CNRS G-SCOP
UMR5272 Grenoble, F-38031
46 av. Félix Viallet, 38000 Grenoble, France
gulgun.alpan@g-scop.grenoble-inp.fr

Halston Hales

Department of Industrial and Systems Engineering
Mississippi State University
Mississippi State, MS 39762, USA
hrh61@msstate.edu

ABSTRACT

Simulation models and discrete optimization models are oftentimes used together in a variety of ways. In this paper, we discuss the issues that modelers must address in cases where simulation models are used to test a discrete mathematical programming optimization model's performance in a stochastic environment. The issues arise during validation of simulation models, when checking agreement between deterministic optimization results and simulation models operating under deterministic conditions. In our case, the issues are derived from validating simulation models that are used to test the performance of scheduling and resource allocation models (integer and mixed-integer programming optimization models) under various types of uncertainty. While the concerns we describe are from our work in the logistics domain (cross-docking operations), they are relevant to a wide variety of problem domains. In addition to describing the issues, we offer suggestions on how modelers might address the concerns.

Keywords: Discrete-event simulation, discrete optimization, validation, verification

1 INTRODUCTION

Modelers oftentimes employ both simulation and optimization models to address a particular problem. When used together, simulation and optimization models can be combined or related in various ways. This paper identifies four relationships between simulation and optimization models that allow the two disparate modeling types to be combined to address a specific problem. These relationships are illustrated in Figure 1 and described below.

a- An optimal decision is made within a simulation model. For instance, Clausen et al. (2012) simulate the operations within a network of LTL (less than truckload) terminals, using optimization (multi-stage mixed-integer program, solved with a modified tabu search) to make decisions regarding the routing between the different terminals.

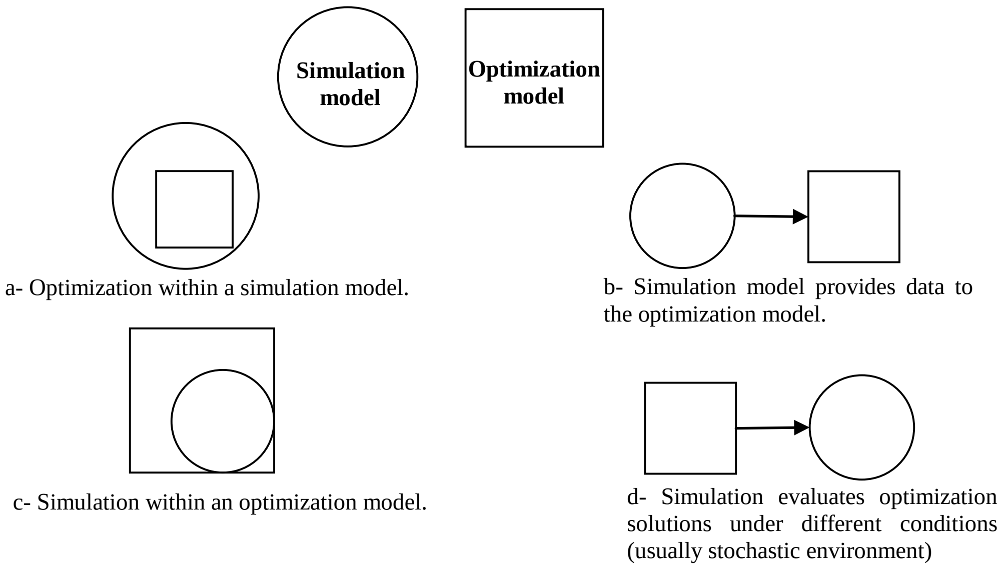

Figure 1 *Complementary uses of simulation and optimization models.*

b- Simulation models generate data that are then used in an optimization model. For example, to address a personnel planning problem at a cross-docking center, Liu and Takakuwa (2009) use a simulation model to determine the workload needed (man-hour requirement of each operating activity for each work-hour). These data are inputs for an integer programming model which produces an optimal schedule for the operators, taking their skills into account. In another example, Hauser (2002) uses a simulation model to provide data on alternative layouts in an automotive manufacturing plant. Genetic algorithms are used to select the layout that minimizes walking distance during the dispatching operation and balancing the workload.

c- A simulation model is embedded within an optimization model. Typically, the simulation can be used to evaluate the objective function associated with a solution obtained from an optimization model. Olafsson and Kim (2002) and April et al. (2004) provide tutorials for this technique called "simulation optimization." The former is rather generic while the latter gives several applications in project portfolio optimization and supply chain management. Another example is a discrete-event simulation model used by Vanguri et al. (2006) to drive an evolutionary strategies algorithm to generate improved production sequences. In another case, Greenwood et al. (2005) describe embedding simulation and optimization models in a decision support system to improve shipbuilding operations. In the logistics field, Aickelin and Adewunmi (2006) use simulation as a black box to evaluate the objective function within a meta-heuristic for the cross dock truck-to-door assignment problem. Instead of using the simulation as a black box, Almeder and al. (2009) translate the solution of the optimization model into decision rules for the discrete-event simulation, and apply the procedure iteratively until a stable point is reached.

d- A simulation model tests the results of an optimization model, e.g. a schedule. Gambardella et al. (1998) model an intermodal container terminal with a discrete-event simulation tool in order to check the validity and robustness of the resource allocation within the terminal, generated by an integer linear program. Wang and Regan (2008) propose two time-based algorithms for the inbound truck scheduling problem in a cross dock in order to minimize the transfer time of pallets. The algorithms and other policies (first-come, first-served and look-ahead) are evaluated with a detailed simulation model. Liu and Takakuwa (2010) test the inbound truck schedule and the employees' schedule in a fresh-food cross dock operation using a simulation model. Deshpande et al. (2007) use discrete-event simulation to evaluate the performances of various heuristics for the dock assignment problem.

The focus of this paper is on the fourth type of relationship. Based on our experience in developing and testing simulation models interacting with optimization models as described in (d), we detail and explain the modeling issues raised by such an optimization/simulation relationship. We explain how those issues can be solved or circumvented. The goal is to provide the community of researchers with useful insights on this technique, and to therefore encourage the use of discrete-event simulation as a means to assess the performance of optimization models.

Discrete optimization and mathematical programming can model a situation in a very realistic way, taking into account as many details as the simulation does; however, adding too many details makes the solution non-computable. Oftentimes, assumptions are made in order to simplify the optimization model and focus on the most salient aspects. A simulation model can be used to validate those assumptions and to determine their validity range. On the other hand, some simplifications can be made in the simulation model in order to closely follow the assumptions made in the optimization model. This is important in order to validate those assumptions.

To validate a model is to determine whether or not it is a meaningful and "accurate" representation of the real system, and contains sufficient accuracy to meet its intended use. It is about "building the right model." Verification is the process of determining whether a model is working as intended. It is about "building the model right."

In order to validate and verify the simulation model, one expects it to behave similar to the optimization model under deterministic conditions. This paper describes how, due to differences in the modeling approaches, disparities can occur even when the models are developed to represent the same system in the same operating environment.

The paper is organized as follows. We begin in section 2 with a brief description of the two cases that provide the foundation for the issues raised in this paper: in the next two sections we define and discuss a set of issues that modelers need to address when developing a simulation model to test results from an optimization model under different conditions. In section 3 we address issues related to the foundational differences between optimization and simulation models – time and space representation and model structure and size. In section 4 we address operational differences in the two modeling approaches – task dependencies, resource assignments, decision making, and performance measures. Section 5 gives the conclusions and perspective of this work.

2 BASES FOR IDENTIFICATION OF MODELING ISSUES

The modeling issues that we define in this paper are the result of testing two optimization models using simulation for robustness under operational conditions that differ from those explicitly considered in the optimization models, e.g. operating in a stochastic environment. One expects differences in the model results and those are the subjects of many research endeavours, such as our two cases described below. Therefore, it is well beyond the scope of this paper to discuss results from such studies.

In the first case (referred later as "Case 1"), we test schedules obtained with the integer program described in Ladier and Alpan (2013). They provide a schedule for inbound and outbound trucks to a cross-docking facility that maximizes transportation providers' satisfaction (in terms of the closeness to their desired arrival and departure times) and minimizes total quantity of items placed in temporary storage (rather than being directly loaded onto an outbound truck). One key assumption in the optimization model is that unloading, scanning, transfer, loading and leaving operations are all done within the same time period (e.g. 60 minutes). That is, the time period is long enough to ensure pallets are processed in masked time. Also, the distance of the transfer (thus the location of the doors) is not taken into account.

The simulation model is used to test the schedules' robustness when subjected to various levels of randomness, e.g. early or late arrivals, variations in process times.

In the second case ("Case 2"), we test the schedules obtained with three mixed-integer linear programs (MILP) solved in sequence, as described in Ladier et al. (2013). The sequential solution process results in detailed timetables (with 15 minutes precision) for the employees of a logistics facility. The task assignments have to cover all of the workload for one day, while taking into account the employees' competencies by assigning each of them to the task for which they are most proficient.

The simulation model is used to test the robustness of the timetables generated by the MILPs, when subjected to randomness in the amount of workload.

Both simulation models were developed using the simulation software *FlexSim*© (www.flexsim.com). Although both models are addressing cross-docking scheduling problems, we believe that the issue raised in the next sections are not related to logistics or cross-docking, and can be encountered by other modelers testing a discrete optimization model by means of a simulation model, regardless of the field of application.

3 ISSUES RELATED TO FOUNDATIONAL DIFFERENCES BETWEEN SIMULATION AND OPTIMIZATION MODELING

The first set of dissimilarities between optimization and simulation that modelers need to consider is related to foundational differences in modeling the underlying system. These differences are described in terms of time representation, spatial representation, model structure, and model size.

3.1 Time representation

How the passage of time is addressed in models, i.e. time representation, constitutes a major difference between discrete optimization and simulation. While this is due to fundamental differences in the modeling approaches, it is also oftentimes due to granularity. On the one hand, temporal optimization employs discrete time intervals where events and resulting activities occur within a time period, e.g. a truck arrives for unloading or a 15-minutes task is assigned to an employee within a one-hour time interval. All that is considered is that these events/activities occur somewhere within the interval; the exact time is not important to the model. On the other hand, discrete-event simulation has a much finer granularity, events occur at precise instances of time; e.g. a truck arrives 27.1752 minutes after the arrival of the previous truck. Also, in simulation, events trigger and are triggered by other events, so timing is an important element.

Therefore, because of these key differences, the behavior of an optimization model using discrete time intervals and the behavior of a discrete-event simulation can never be matched exactly.

In Case 1, the integer programming model only allows a truck to leave at a multiple of 60 minutes, while the trucks in the simulation model leave at any time; they leave when a specified condition is met, e.g. when a truck is empty (inbound) or full (outbound). Therefore, if we measure the difference between the trucks departure time as calculated by the optimization model and the trucks departure time as observed in the simulation, we can get gaps as large as 59 minutes even though the system behaves as expected. Those gaps can be reduced by shortening the time intervals used in the optimization model; however, that makes the optimization model more complex (and possibly incomputable) and some gaps will always be observed. One way to circumvent this issue is to measure performance in terms of intervals. For example, if according to the simulation model the departure of a truck occurs at 17:11 and if it is planned at 17:00 in the optimization model, then it is considered "on time" if the masked time interval considered in the optimization model is, as in the example above, sixty minutes.

Modelers should therefore be aware of the differences between their different models in terms of granularity, and can circumvent them by using time intervals rather than absolute time for their simulation measures.

3.2 Spatial representation

Simulation models not only consider events in time, they oftentimes consider spatial effects on system behavior and performance. Most simulation software integrates spatial data such as the current location of the transporting resource, its destination(s), its speeds and possibly acceleration, etc., to determine travel times. This granularity is not always considered in optimization formulations.

Optimization can take into account speed and acceleration – but they make the model considerably more complex. Therefore, spatial effects are taken into account in mathematical programming only if they are a key part of the decision to be taken – for instance, if the objective is to minimize the distance walked by the workers in a cross-dock facility. When the spatial dimension is not at the core of the problem, modelers will tend to ignore the travel time or use masked time in order to simplify the optimization models. An action short enough can be considered as instantaneous; a set of actions that takes about one hour can be considered as done within a one-hour time interval (Case 1).

Because of the spatial nature of the actions, such assumptions do not adapt well to discrete-event simulation. There is a tradeoff to be made between fidelity in the optimization model (zero travel times) and closeness to realistic operations. The solution we propose is a compromise approach: we control the transfer time by making it a process step in the simulation instead of a distance- and speed-affected move from one point to another. This way, in both our testing cases we can validate the consistency of the simulation model with the optimization model by setting the transfer time to zero, yet be able to easily extend the simulation model to incorporate more realistic aspects by modeling the transfer time with probability distributions or constructs that consider actual location and speed of the traveling resource.

The solution we propose for our special case has to be adapted to each individual case; but the main advice for modelers here is to reflect whether the simulation model should be closer to the optimization model or to the real-life operations, and how the tradeoff can be made.

3.3 Model structure and size

The structure of the model and the size of an instance are defined differently for each type of model.

In optimization models, the size of an instance is driven by the number and nature of decision variables, which depends on the number of objects considered in the decision (number of workers, number of trucks to be processed...) and the number of time periods considered. Size is not of special concern in formulating or describing the model since the constraints are specified in a tight mathematical notation and system-specific input parameters are provided in a structured manner.

Although the size of the instance does not change the formulation of an optimization model, the complexity of the problem drives the solution method chosen, and therefore the solution accuracy and speed. The complexity depends on the nature of the decision variables (real or integer), the number of constraints and their formulation, the quality of the bounds created by those constraints. If the problem cannot be solved in polynomial time, an integer program gives optimal results for small instances but cannot be scaled up for large instances; whereas, a meta-heuristic does not guarantee optimality but can be easily scaled.

In a simulation model however, the size of an instance is defined in terms of the number of objects considered (number of processor units, number of workers, etc.) and, more importantly, the number and type of relationships among the objects. The objects and their relationships form the structure of the model and thus even if the simulation model has been designed to be easily scalable, it is necessary to change the structure in order to change the size of the instances it can process. Time, however, is implicitly a part of all discrete-event simulation models: it is easy to increase the length of the time horizon considered.

The complexity of the simulation model does not affect the choice of solution method, only slightly impacts solution speed (model run time), and has no effect on accuracy.

Typically models are validated, at least initially, using small instances, either few objects or few time periods, or both. However, it may be necessary to test models in larger contexts. For example, Case 1 was validated for a cross-dock facility model with 3-input doors and 3-output doors, but a realistic case would be a 50-input doors by 50-output doors arrangement.

Since it is difficult to scale up the structure of simulation models, and since changing the optimization solution method requires considerable research and development, it is important to specify the size of the cases to be considered early on in the project.

4 ISSUES RELATED TO OPERATIONAL DIFFERENCES BETWEEN SIMULATION AND OPTIMIZATION MODELING

The second set of dissimilarities between optimization and simulation that modelers need to consider is related to operational differences in modeling the underlying system. These differences are described in terms of task dependencies, resource assignment, process logic, and performance measures.

4.1 Tasks dependencies

Precedence relationships are necessary for an optimization model to take into account the order in which items are processed or tasks are carried out. Again, if the decision regarding order is not the key decision being considered, it will be taken out of the optimization model for the sake of simplification and computation time. It will therefore only determine a given amount of tasks that have to be carried out within a given time interval: the order, the batch size, the parallelism of the tasks are not taken into account. The simulation, if not given further information, will assign a default order to process the different tasks – usually FIFO (first in, first out) order.

Although it is not explicitly mentioned in the optimization model, the decision of whether the items are processed one after the other or as a batch has to be accounted for in the simulation, since it can lead to discrepancies between the models. For instance, one would assume that a single-channel process ($c = 1$) working at a rate r produces the same amount of products as a multi-channel process with c channels and rate r/c per channel. Therefore, outputs $r \times c$, are the same and the two options appear interchangeable. While this is true on the average, it is not true within every time interval.

We use the pallet transfer process from Case 1 as an illustration. Assume the transfer rate per resource is $r = 10$ pallets/hour and the number of available resources is $c = 3$. If an outbound truck arrives at 10:00, then any pallet transferred from inbound before that time goes to storage, while any pallet processed after 10:00 goes directly into the outbound truck. A process with capacity $c = 1$ and rate per channel of $r = 30$ pallets/hour transfers each pallet in 2 minutes. Therefore, between 9:55 and 10:00, two pallets are processed and they both go into storage. However, a process with capacity $c = 3$ and rate per channel of $r = 10$ pallets/hour transfers each pallet in 6 minutes. Therefore, between 9:55 and 10:00, no pallet is fully transferred and no pallet goes into storage.

Modelers should therefore keep in mind that the processing order, batch size and precedence relationships have to be accounted for in the simulation, even if they have been simplified in the optimization model.

4.2 Resource assignment

The basic manner in which resources are selected for use may differ between optimization models and simulation models, thus leading to disparity of results and validation challenges. Differences are oftentimes narrowed through the application of additional customized logic in the simulation model.

One common application of mathematical programming models is to make assignments between resources and tasks, as in our Case 2. When simulation is used to test the implementation of the assignment, the default simulation approach may not result in comparable results. For example, in a simulation if a task needs to be performed by a resource and several resources are available, a default first-in, first-out criteria may not match the optimized assignment. Therefore, information on the optimization assignments must be provided to the simulation so that the task can select the appropriate resource. In addition, if none of the available resources result in a match with the optimized assignment, then logic must be provided in the simulation in order to guide the task's selection from the available resources; or, the task must wait until the appropriate resource is available.

Similarly, in a simulation if a resource becomes available and there are multiple tasks that need to be completed, a default first-in, first-out criteria may not match the optimized assignment. Therefore, as indicated above, information on the optimization assignments must be provided to the simulation so the resource can select the appropriate task. In addition, if none of the tasks result in a match with the optimized assignment, then logic must be provided in the simulation in order to guide the resource's selection of the available task; or, the resource must be made idle and wait until an appropriate task becomes available.

In short, modelers should think of ways to incorporate the optimization results in the simulation models, and add logic beyond the default logic in order to take this data into account in the decisions made by the simulation.

4.3 Process logic

By its nature, simulation is greedy, i.e. it processes as many items (pallets in our cases) as possible in one event (instance in time) while the integer programming model can transfer less pallets per time period if it improves the objective function in the optimization. In Case 1, in order to force the simulation model to obtain a result similar to the optimization model, it is necessary to limit the amount of pallets that can flow through the model during each time period. This can be done by using the output of the integer programming model as inputs that give the capacity of the transfer process in the simulation model; this capacity will vary through time. It is interesting to note that this fix can also make the simulation closer to reality. Therefore, this issue can be mitigated by adding process logic to the simulation model that provides flexible capacity over time to the transfer operation.

Since simulation is event-driven, priorities are oftentimes required in order to represent the appropriate behavior. For example, if both an inbound truck and an outbound truck need to be unloaded, which should the resource service first? In Case 1 and Case 2 we include process logic in the simulation model to push items from inbound trucks and pull resources from the arriving outbound truck. The pulling algorithm gives the priority to the outbound trucks; thus, we first seek to fill the outbound trucks that have to leave rather than emptying the inbound docks. The logic implemented by the simulation model is close to what a manager would do; however, it does not give the optimal solution (i.e. exactly the same solution as the one given by the integer programming model) in all cases. In some cases, it leads to having outbound trucks leaving earlier than planned while inbound trucks leave late. The solution approach described in the first paragraph, which consists of adding a transfer capacity, would not be appropriate in that last case, due to the structure of the problem.

It is therefore important to keep in mind, when testing an optimization model by means of a simulation model, that the former gives the optimal solution (when exact solving solutions are used) while the latter does not. The simulation can be driven towards a solution closer to the optimal, but it cannot determine the optimal solution unless it embeds an optimization module (this is the case (a) in Figure 1, and beyond the scope of this paper). Using the optimum solution determined by the optimization model as an input in the simulation model is a good solution, but the simulation model should also have its own decision logic, in order to be able to handle stochastic changes.

4.4 Performance indicators

The performance indicators needed when testing an optimization model with a simulation model differ from the indicators that would classically be used in a simulation. The main goal here is to compare the performance of the simulation with the performance of the optimization model. The indicators should ensure a fair comparison between the two models. For example, as mentioned in section 3.1, punctuality should be measured with intervals rather than with an absolute value, so as to account for the difference in time granularity between the two models.

The performance indicators, once determined in the validation phase, will also be used in the robustness assessment phase, in order to compare the performance of the simulation model in the deterministic case with its performance when some elements of the model follow random distributions.

The indicators play a key role in the validation process, and should therefore be chosen carefully.

5 CONCLUSIONS

Simulation and optimization modeling take very different approaches to address operations problems. Of course this is due to fundamental differences in the way the two types of models are structured and solved. Even though quite different, simulation and optimization are oftentimes used in complementary roles to improve the decisions that result from using the models. These inherent differences provide challenges to modelers, especially in terms of validation and verification.

This paper is based on the cases of two simulation models used to evaluate the performance of discrete optimization models (IP, MILP) in a stochastic environment. Those application cases model cross-docking problems, but the issues we point out can occur regardless of the application field. We describe several key challenges occurring when the simulation models have to be validated, i.e. when the behavior of the simulation model and the optimization model are compared under deterministic conditions. We offer suggestions for mitigating those challenges.

We hope that the insights given on these issues can and will encourage an increase in the use of discrete-event simulation to assess the performance of mathematical optimization models. We also hope that other modelers encountering various modeling issues will be encouraged to communicate them so that the community can benefit from their experience.

ACKNOWLEDGMENTS

This work has been supported by the Rhône-Alpes Region.
G-SCOP is partner of the LabEx PERSYVAL-Lab (ANR−11-LABX-0025).

REFERENCES

Aickelin, U. & Adewunmi, A., 2006. Simulation optimization of the crossdock door assignment problem. In *UK Operational Research Society Simulation Workshop*. Leamington Spa, UK, pp. 1–3.

Almeder, C., Preusser, M. & Hartl, R.F., 2009. Simulation and optimization of supply chains: alternative or complementary approaches? In *Supply Chain Planning*. Springer Berlin Heidelberg, pp. 1–25.

April, J. et al., 2004. New advances and applications for marrying simulation and optimization. In *Proceedings of the Winter Simulation Conference*. Washington, DC, USA, pp. 80–86.

Clausen, U. et al., 2012. Combining simulation and optimization to improve LTL traffic. *Procedia - Social and Behavioral Sciences*, 48(1), pp. 1993–2002.

Deshpande, P.J. et al., 2007. Simulating less-than-truckload terminal operations. *Benchmarking: An International Journal*, 14(1), pp. 92–101.

Gambardella, L.M., Rizzoli, A.E. & Zaffalon, M., 1998. Simulation and planning of an intermodal container terminal. *Simulation*, 71(2), pp. 107–116.

Greenwood, A.G. et al., 2005. Simulation optimization decision support system for ship panel shop operations. In *Proceedings of the Winter Simulation Conference*. Orlando, Florida, USA, pp. 2078–2086.

Hauser, K., 2002. *Simulation and optimization of a crossdocking operation in a just-in-time environment*. University of Kentucky Doctoral dissertations.

Ladier, A.-L. & Alpan, G., 2013. Scheduling truck arrivals and departures in a cross dock: earliness, tardiness and storage policies. In *International Conference on Industrial Engineering and Systems Management*. Rabat, Marocco.

Ladier, A.-L., Alpan, G. & Penz, B., 2013. Joint employee timetabling and rostering: a decision-support tool for a logistics platform. *European Journal of Operational Research: to appear*.

Liu, Y. & Takakuwa, S., 2010. Enhancing simulation as a decision-making support tool for a crossdocking center in a dynamic retail-distribution environment. In *Proceedings of the Winter Simulation Conference*. Baltimore, MD, USA, pp. 2089–2100.

Liu, Y. & Takakuwa, S., 2009. Simulation-based personnel planning for materials handling at a cross-docking center under retail distribution environment. In *Proceedings of the Winter Simulation Conference*. Austin, TX, USA, pp. 2414–2425.

Olafsson, S. & Kim, J., 2002. Simulation optimization. In *Proceedings of the Winter Simulation Conference*. San Diego, CA, USA, pp. 79–84.

Vanguri, S., Hill, T.W. & Greenwood, A.G., 2006. Optimizing flow shop sequencing through simulation optimization using evolutionary methods. In *Industrial Engineering Research Conference*.

Wang, J.-F. & Regan, A., 2008. Real-time trailer scheduling for crossdock operations. *Transportation Journal*, 57(3), pp. 5–20.

Les cahiers Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication sur des problèmes liés au mathématiques discrètes.

Responsables de la publication : Nadia Brauner et András Sebő
ISSN : 1298-020X