OR for Simulation and Its Optimization

Nico M. van Dijk and Erik van der Sluis

Fac. of Economics and Business, University of Amsterdam, The Netherlands
N.M.vanDijk@uva.nl
H.J.vanderSluis@uva.nl

Summary. This is an expository paper to promote the potential of OR (Operations Research) for simulation. Three applications will therefore be presented which include call centers, check-in at airports, and performance bounds for production lines. The results indicate that (classical and new) OR results might still be most fruitful if not necessary for practical simulation and its optimization.

1 Introduction

Simulation or more precisely as meant in the setting of this paper: discrete event simulation is known as a most powerful tool for the evaluation of logistical systems such as arising in manufacturing, communications or the service industry (banks, call centers, hospitals). A general characterization is that these systems:

- are complex
- involve stochastics and
- require some form of improvement
  (such as by infrastructure, lay-out or work procedures).

Analytic methods, such as standardly covered by the field of OR (Operations Research), in contrast, only apply if:

- the systems are sufficiently simple and
- special assumptions are made on the stochastics involved.

On the other hand, also simulation has a number of limitations:

1. The lack of insights;
2. Techniques for optimization;
3. The confidence that can be adhered to the results.

In this paper, therefore, it will be illustrated how the discipline of OR (Operations Research) might still be beneficial for each of these three aspects.

First in section 2, it is shown that a classic queueing insight in combination with simulation already lends to counterintuitive results and ways for optimization for the simple question of whether we should pool servers or not such as in call centers. Next, in section 3, a check-in problem is presented. First, it is argued that simulation is necessarily required. Next OR-technique, in this case LP-formulation, is used for optimization. Finally, in section 4, it is shown that analytic queueing results can still be most supportive for the simulation of queueing network such as production lines.

The results in this paper all rely upon earlier and practical research that has been reported in more extended and specific technical papers for each of the applications separately (cf. [1], [2], [3]) but without the common message of OR for simulation as promoted in this paper.

2 To Pool or Not in Call Centers

Should we pool servers or not? This seems a simple question of practical interest, such as for counters in postal offices, check-in desks at airports, physicians within hospitals, up to agent groups within or between call centers. The general perception seems to exist that pooling capacities is always advantageous.

An Instructive Example (Queueing)

This perception seems supported by the standard delay formula for a single (exponential) server with arrival rate λ and service rate μ : $D = 1/(\mu - \lambda)$. Pooling two servers thus seems to reduce the mean delay by roughly a factor 2 according to $D = 1/(2\mu - 2\lambda)$.

However, when different services are involved in contrast, a second basic result from queueing theory is to be realized: Pollaczek-Khintchine formula. This formula, which is exact for the single server case, expresses the effect of service variability, by:

$\mathbf{W}_G =$ $\frac{1}{2}(1 + c^2)\mathbf{W}_E$ with $c^2 = \sigma^2/\tau^2$ and

$\mathbf{W}_G$ the mean waiting time under a general (and E for exponential) service distribution with mean τ and standard deviation σ.

By mixing different services (call types) extra service variability is brought in which may lead to an increase of the mean waiting time.

This is illustrated in the figure below for the situation of two job (call) types 1 and 2 with mean service (call) durations $\tau_1 = 1$ and $\tau_2 = 10$ minutes but arrival rates $\lambda_1 = 10\lambda_2$. The results show that the unpooled case is still superior, at least for the average waiting time $\mathbf{W}_A$. Based on these queueing insights, a two-way or one-way overflow scenario can now be suggested, which leads to further improvement as also illustrated in Fig. 1.