PROMETHEE-i SELECTING THE BEST SIMULATION MODEL CONFIGURATION BASED ON MULTIPLE PERFORMANCE MEASURES

H. Pastijn, F. Van Utterbeeck, and R. Van Loock
Royal Military Academy
Renaissance Avenue, 30
B-1000 Brussels
Belgium
E-mail: Hugo.Pastijn@rma.ac.be

KEYWORDS

Discrete event simulation, system configuration comparison, outranking methods, multicriteria decision making, performance measures.

ABSTRACT

We introduce the use of a variant of the original multicriteria decision making method Promethee, in order to select the best simulation model configuration among a finite set of alternatives. For each alternative configuration a number of replications of terminating simulation runs is performed. At the end of each replication, and for each configuration, the result of a number of performance measures is obtained. In the selection problem, these performance measures are typically conflicting criteria for which the alternative configurations have been assessed by a number of computer simulation replications. We submit these data to an interval version of the Promethee outranking method, in order to select the best model configuration. We illustrate this by means of an incident management model for a call centre.

INTRODUCTION

Comparing alternative system configurations based on stochastic computer simulation output has been extensively studied in the literature (Law and Kelton, 1991) (Kleijnen, 1975). More generally, there is an impressive bibliography available about stochastic ordering (Mosler and Scarsini, 1993). However, most of the effort is done when only one performance measure is involved. In reality various performance measures are simultaneously involved in assessing the behaviour of a system. Different system configurations will typically improve some performance measures and deteriorate some other ones. The selection of the best system configuration among a finite set of alternatives assessed for a finite set of criteria (performance measures) is a typical multicriteria decision making problem. For these problems there is a wide variety of methods available nowadays. The original Promethee methods (Brans et al., 1986), belonging to the outranking category of multicriteria decision making tools, are appropriate to handle these problems in the deterministic case. In order to take into account the stochastic character of the computer simulation runs, these methods have to be adapted (Mareschal, 1986). We propose to use an interval version: Promethee-i (Mareschal and Le Téno, 1992).
First we describe the main features of Promethee-i.
Secondly we describe a simulation model of a call centre which was implemented in ARENA® (Kelton et al., 1989). Finally we compare alternative designs of this model based on traditional performance measures: waiting times in queues, productivity, cost and service level. Therefore we run a number of replications of terminating simulation runs and we apply Promethee-i. The selection result is studied as a function of the number of replications of the simulation runs.

PROMETHEE-i

For the original Promethee method we refer to Brans et al.(1986). This method is based on crisp data assessing the n alternatives on k criteria (perfomance measures).
For each criterion a pairwise comparison (difference of assessment) of alternatives a and b is translated by a preference indicator $P_j(a,b)$ on the interval [0,1]. These $P_j(a,b)$ are aggregated over the set of all criteria by :
$\pi(a,b)= \Sigma_j \omega_j\, P_j(a,b)$, with ω_j in [0,1] being the weight of criterion j. Then are calculated for each alternative a:
the strength $\phi^+(a)=(1/(n-1)).\Sigma_x \pi(a,x)$,
the weakness $\phi^-(a) =(1/(n-1)).\Sigma_x \pi(x,a)$ and
the net dominance $\phi(a) = \phi^+(a) - \phi^-(a)$. The best alternative is the one with the highest net dominance.

When we have m replications of computer runs, then we obtain for each alternative configuration m assessments for each criterion (performance measure). These m assessments can be represented for alternative a by an interval $[a^l,a^u]$, which we take here either as the inter-quartile interval, or as the 99.9% confidence interval on the mean (results with both methods will be compared). All the arithmetic of Promethee is now extended, keeping intervals all along the calculations, by means of the following definitions:

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)

ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

$[a^l,a^u] + [b^l,b^u] = [a^l+ b^l , a^u+ b^u]$ and
$[a^l,a^u] - [b^l,b^u] = [a^l- b^u , a^u- b^l]$.
We obtain consecutively
$P_j(a,b) = [P_j^l(a,b) , P_j^u(a,b)]$,
$\pi(a,b) = [\pi^l(a,b) , \pi^u(a,b)]$,
with $\pi^l(a,b) = \Sigma_j\omega_j P_j^l(a,b)$ and
$\pi^u(a,b) = \Sigma_j\omega_j P_j^u(a,b)$,
$\phi^+(a) = [\phi^{+l}(a) , \phi^{+u}(a)]$,
with $\phi^{+l}(a) =(1/(n-1)).\Sigma_x\pi^l(a,x)$ and
$\phi^{+u}(a) =(1/(n-1)).\Sigma_x\pi^u(a,x)$,
$\phi^-(a) = [\phi^{-l}(a) , \phi^{-u}(a)]$,
with $\phi^{-l}(a) =(1/(n-1)).\Sigma_x\pi^l(x,a)$ and
$\phi^{-u}(a) =(1/(n-1)).\Sigma_x\pi^u(x,a)$,
$\phi(a) = \phi^+(a) - \phi^-(a) = [\phi^l(a) , \phi^u(a)]$.

In addition the original Promethee method is applied by taking into account all the worst bounds of the assessment intervals $[a^l,a^u]$ and another time by taking all the best bounds of these assessment intervals $[a^l,a^u]$ for all alternatives on all criteria. This yields for each alternative another interval $[\phi^{l'}(a) , \phi^{u'}(a)]$.

Finally this Promethee-i procedure is resulting into a trapezoidal fuzzy number $[\phi^l(a) , \phi^{l'}(a) , \phi^{u'}(a) , \phi^u(a)]$ for each alternative *a*.

On these fuzzy numbers we apply the Yager operator Ψ (Yager, 1981)(Detyniecki et al., 2001), and the best alternative corresponds to the highest value for this Yager operator Ψ.

INCIDENT MANAGEMENT MODEL FOR A CALL CENTRE

The simulation model we consider is the incident management process of a call centre. We will describe the elements of the model (the incidents, the resources and the skills matrix) and the process flow (Fig.1). (Van Loock et al., 2003)

The *incidents* are initiated by the customers of the call center. These incidents are represented by the calls that arrive at the centre. These *incoming calls* follow a stochastic arrival pattern. The calls are subdivided into 2 categories and 6 subcategories, depending on the area of expertise required by the customer. Each category has a specified probability of occurrence, while the subcategories within a certain category are assumed to be equiprobable.

The *resources* in our model are the dispatcher(s) and the system engineers. Each resource has its own weekly working schedule, an hourly cost (based on the number of skills known) and a FIFO queue associated with it. Incoming calls will wait in the FIFO queue if the resource is busy. A call will be *rejected* (and leaves the system immediately) if the time spent waiting in a FIFO queue exceeds a certain fixed threshold.

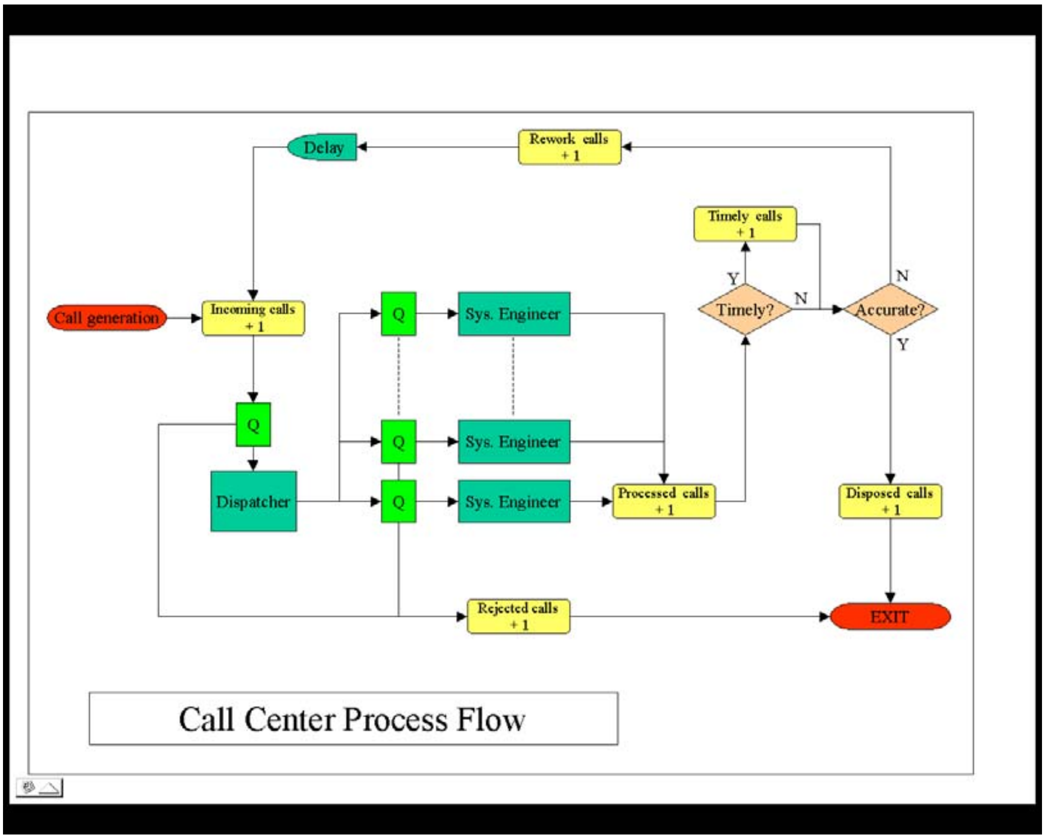

Figure 1: Call Center Process Flow

Every system engineer has his own areas of expertise, which are specified in the *skills matrix*. Every line in the matrix represents a subcategory, while every column represents a system engineer.

The *process flow* used in our model can be summarized as follows. Every incoming call must pass through a dispatcher. The dispatcher will rout the call to a system engineer whose area of expertise covers the category and subcategory of the call. If multiple system engineers are eligible, the dispatcher will rout the call to the resource with the shortest queue. Ties are broken in favour of the resource located the most to the left in the skills matrix. The *dispatching time* (the time needed by the dispatcher to decide on the routing of the call) follows a triangular distribution.

The *processing time* (the time the system engineer needs to handle a call) follows an exponential distribution, regardless of the subcategory. For every *processed call*, there is a fixed probability that the customer is not completely satisfied with the assistance provided. These customers will call back after a stochastic delay. These subsequent *rework calls* will result in a decrease in the performance of the call centre. If the customer is satisfied with the assistance provided, the call is *disposed* and leaves the system.

PERFORMANCE MEASURES

We use four performance measures: waiting times in queues, resource utilisation or productivity, service level and system cost. Waiting times in queues and resource utilisation are average values obtained from standard ARENA® statistics. Service level is expressed as the percentage of arriving calls which are finally disposed after a successful handling by the available resources (and as a consequence were not ejected from the system). The cost of a system engineer depends on his degree of polyvalence (number of skills). The overall system cost is a stochastic entity due to the fact that the resources continue to work at the end of their daily schedule until all calls waiting in their queue at the end of the working day have been processed.

SIMULATION

Ten different configurations of the system were simulated, and the observed performance indicators were compared. Reconfiguration of the system is simulated through changes in either the number of resources (additional system engineers or additional dispatchers), or through changes in the skills matrix. More radical changes, like the implementation of a frontoffice-backoffice strategy, were not considered. We restricted the eligible configurations by imposing a maximum allowable weekly cost and a minimum acceptable servicelevel. The OptQuest for Arena software was then used to heuristically identify eligible scenarios that optimise one of the four selected performance indicators. Finally we selected the four optimal configurations identified by OptQuest, as well as six variations of those as the ten “likely candidates” for our simulation study.

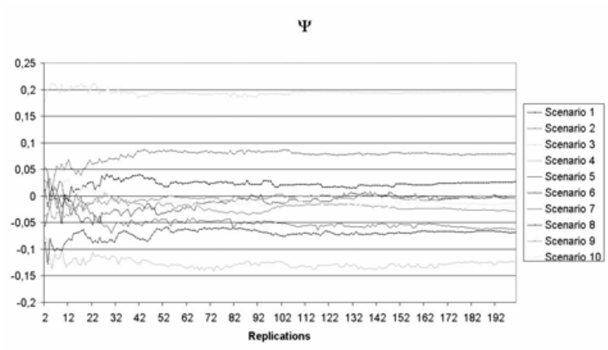

Figure 2: Ψ Operator Based on Interquartile Intervals

Figure 2 shows the evolution of the Ψ operator based on the inter-quartile intervals, during 200 replications for the 10 system configurations (scenarios).

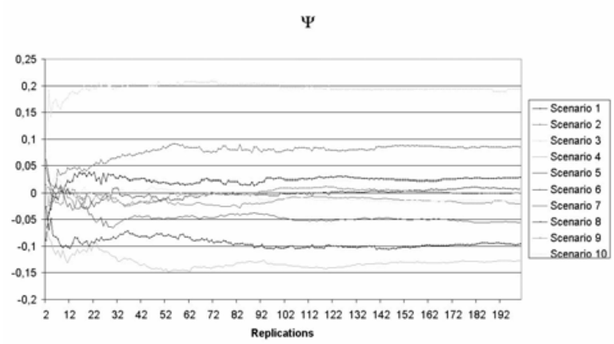

Figure 3: Ψ Operator Based on Confidence Intervals

Figure 3 shows the evolution of the Ψ operator based on the 99.9% confidence intervals of the mean performance measures, during 200 replications for the same 10 system configurations (scenarios).

We remind that the 99.9% confidence intervals become of course narrower as the number of replications increases. As a result, this method is expected to converge towards the crisp version of Promethee, by using the average performance measures as an input for the computations at each replication. This is confirmed by figure 4 which is showing the evolution of the net dominance ϕ of Promethee during 200 replications for the 10 system configurations.

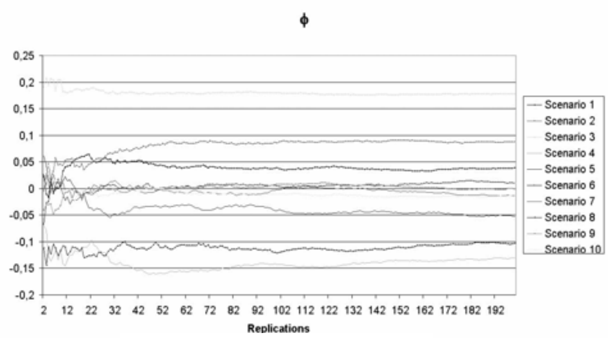

Figure 4: Net Dominance

We notice that the ranking of the alternative system configurations is quite stable for the three methods for a number of replications exceeding about 30 in this experiment. Rank inversions occur even at a high replication number for the standard (crisp) Promethee method and also for the Promethee-i method using 99.9% confidence intervals on the mean performance measures. These ranking inversions occur especially when the Ψ operator values or the net dominances ϕ are very close to each other. We see however in this experiment that the discrimination in ranking between system configurations which are quite distant in terms of Ψ operator values and net dominances ϕ after 200 replications, are identified earlier (after fewer replications) when we use the interquartile intervals in the Promethee-i method. This is clearly illustrated by a zoom-in on the evolution of the Ψ operator value and net dominance ϕ during the earliest replications for 5 system configurations which happen to become those at the top and the bottom of the ranking, and are sufficiently distant to avoid late (after many replications) rank inversions. Comparison of figure 5 (Ψ operator based on interquartile intervals), figure 6 (Ψ operator based on 99.9% confidence intervals) and figure 7 (net dominance ϕ with the standard crisp Promethee method), illustrate this observation.

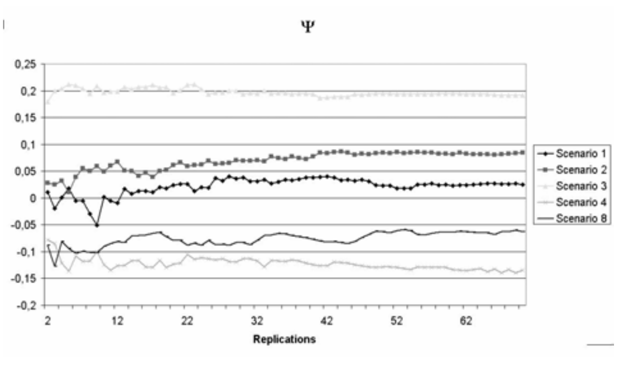

Figure 5: Ψ Operator Based on Interquartile Intervals

It seems indeed that the use of more information about the variability of the performance measures through the entire replication scheme by means of the interquartile Promethee-i method, is sooner discriminating between system configurations. The final ranking becomes the same by the other methods, but it is stable only after a larger amount of replications.

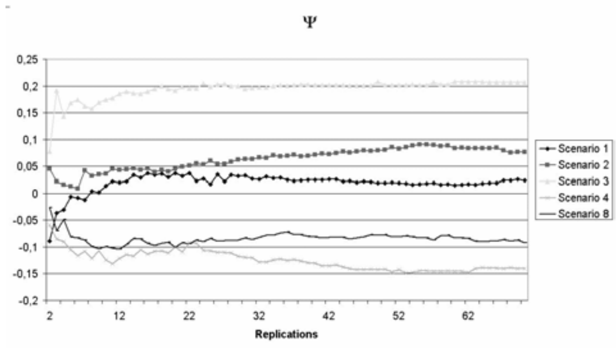

Figure 6: Ψ Operator Based on Confidence Intervals

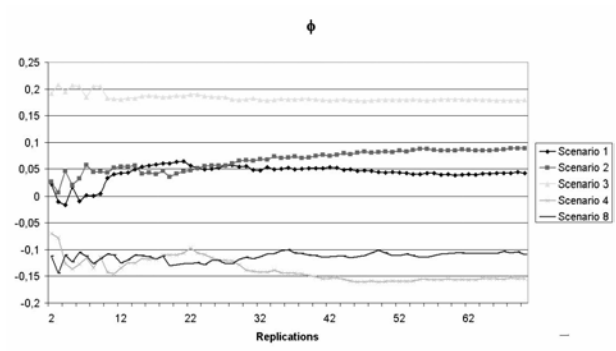

Figure 7: Net Dominance

The same observations were made with the same experiments but by replacing the exponential distribution of system engineers' service times by triangular distributions with a lower variability of the input data in the model. Discrimination between the three computational schemes was then not so obvious. It is indeed more relevant to use the Promethee-i method when the variability of stochastic elements in the model becomes higher.

CONCLUDING REMARKS

The results for the different computational schemes using the Promethee-i method, compared with the one replacing the assessment intervals by the observed mean values and then applying the original Promethee method, show that the results for all methods are similar if the number of replications increases. However, it seems that the interquartile assessments of performance measures, combined with the Promethee-i method is discriminating very fast between system configurations.

This method seems to be very promising when the number of replications should be kept low (for instance in an almost real-time environment for crisis management decision support).
No assumptions about independence of citeria (performance measures) assessments are of course necessary for applying Promethee-i. No assumptions are made about the probability distribution of assessments.
It is of course not evident to evaluate the probability of correctly selecting system configurations. More experimental research might shed some light on this issue.
Further research about the use of alternative multicriteria methods could be inspired by Fodor and Roubens (1994) or by Pastijn and Leysen (1989).

REFERENCES

Brans, J.P., Vincke P. and Mareschal, B. 1986. *How to select and how to rank projects: the promethee method.* European Journal of Operational Research 1986(24), pp. 228-238.

Detyniecki, M. and Yager, R.R. 2001, *Ranking fuzzy numbers using α-weighted valuations,* International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol.8(5), 2001, pp. 573-592.

Fodor, J. and Roubens, M. 1994. *Fuzzy Preference Modelling and Multicriteria Decision Support.* Kluwer Academic Publishers.

Kelton, W.D., Sadowsky, R.P., and Sadowsky, D.A. 1998. *Simulation with Arena*, WCB/McGraw-Hill.

Kleijnen, J.P.C. 1975. *Statistical Techniques in Simulation – Part II,* Marcel Dekker, Inc.

Law, A.M. and Kelton, W.D. 1991. *Simulation Modeling and Analysis*. 2nd ed, McGraw-Hill, New York.

Mareschal, B. 1986. *Stochastic multicriteria decision making and uncertainty.* European Journal of Operational Research, 1986 (26), pp. 58-64.

Mosler, K. and Scarsini, M. 1993. *Stochastic Orders and Applications – A Classified Bibliography.* Lecture Notes in Economics and Mathematical Systems (Vol 401), Springer-Verlag.

Mareschal, B. and Le Téno, J.F., 1992. *An Interval version of Promethee for the comparison of building products design with ill-defined data on environmental quality*, European Journal of Operational Research, 109, pp.522-529.

Pastijn, H. and Leysen, J. 1989. *Constructing an outranking relation with ORESTE.* Mathematical and Computer Modelling 1989 (12), pp. 1255-1268.

Van Loock, R. and Pastijn, H. 2003. *Performance measures in simulating incident management processes of a call centre.* Orbel 17, Brussels Jan 23-24 2003

Van Loock, R., Van Utterbeeck, F. and Pastijn, H. 2003. *TAPE Performance measures implemented in an incident management model,* Valencia (Spain) ISC2003

Yager, R. R. 1981. *A Procedure for ordering fuzzy subsets of the unit interval.* Information Science 1981 (24), pp. 143-161.