TRANSPORT PROBLEMS 2021 Volume 16 Issue 2
PROBLEMY TRANSPORTU DOI: 10.21307/tp-2021-026

Keywords: overhead cranes; Monte Carlo simulation; variance reduction; parallel computing

Janusz SZPYTKO, Yorlandys SALGADO DUARTE*
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Cracow, Poland
**Corresponding author*. E-mail: salgado@agh.edu.pl

ROBUST SIMULATION METHOD OF COMPLEX TECHNICAL TRANSPORT SYSTEMS

Summary. In the optimization of technical systems focused on a specific functional purpose (reliability, safety, and availability) with the use of simulation methods, an important parameter is the digital simulation time of the research subject. With the complexity of the issue, the digital simulation time increases. The aim of the article is to present a method (combination of parallel computing and variance reduction techniques) of reducing the computer simulation time of the research technical object. An example of the application of the developed method was presented as a result of an experiment conducted for decision making and control processes aimed at optimizing the process of operating overhead cranes in critical conditions. In this paper, selecting parallel batch jobs computation and stratified sampling, we exponentially decreased the simulation time, finding fast and practical solutions and eliminating the time constraint in the search of solutions.

1. INTRODUCTION

Monte Carlo methods consist of generating consecutive random samples with computer algorithms to obtain numerical results. It is usually applied in three types of problems: optimization, numerical integration, and generation of samples from a probability distribution according to Kroese et al [7].

Engineering problems sometimes present countless free or unknown variables, making it suitable to use Monte Carlo methods to simulate the sensitivity of the system according to the unknown variables. Authors like Hubbard and Samuelson [5] highlight that a frequent application in system engineering problems is failure predictions based on Monte Carlo simulations.

According to the law of large numbers and the Central Limit Theorem, the calculation of integrals described by the expected value (centre of mass) of some random variable can be approximated to the empirical mean (also known as sample mean) of independent samples of the variable, and it is known that the dispersion around the expected value is associated with the sample variance and the sample length. Therefore, Monte Carlo methods are often used to solve probabilistically oriented problems, particularly those involving mixture or compound probability distributions. A frequent application is to generate random samples from parameterized probability distributions, commonly known as the Markov chain Monte Carlo (MCMC) sampler.

Examples of practical applications based on the Monte Carlo method are the papers, Monte Carlo simulation model to estimate the reliability of logistics and supply chain networks from Ozkan and Kilic [11]; Monte Carlo tolerance simulation on a prefabricated construction assembly, showing a proactive design tool with several key advantages for prefabricated and offsite construction from Rausch et al [12]; Monte Carlo simulation model to analyze and optimize the time interval between periodic inspections in cold standby systems considering the required availability and the lowest cost possible from Alebrant Mendes and Weber Lorenzoni [1]; and a model to improve cooling load

prediction reliability method in which the input variables are calibrated offline with Monte Carlo simulations and stochastic treatment before inputting them into the prediction model from Fan et al [4].

In the references cited before, all of them highlight the advantages of the simulation approach showing relevant results, but only a few of them show practical implementations within a reasonable CPU time.

The obvious limitation in the simulation approach is clearly time, and we believe that this is the reason why authors sometime just ignore the limitation, as we can observe in the references cited.

But, on the other hand, references such as Dieker et al [3] and Leahu et al [8], work strongly in parallel with methodological solutions on how to allocate time-consuming simulations depending on the length of the queues.

In this paper, we touch on a classical application of Monte Carlo simulation: generating samples from probability distributions fitted to historical degradation data. Specially, we estimate by convolution the loss capacity (risk indicator) given the stochastics availability (including degradation due the operation and maintenance planned) of the analyzed overhead crane system and the minimum set of overhead cranes needed to guarantee the production line. The risk indicator obtained is used as a criterion to evaluate the quality of the maintenance scheduling.

More details about the model adopted in this paper can be found in the references Szpytko and Salgado Duarte [16, 17].

As other applications cited, when the complexity of the simulation increases to assess a scenario, the computation time needed to generate a robust estimation can be an issue.

Knowing the limitation, in this investigation, we discuss two possible improvements to reduce the simulation time given a fixed scenario. As a starting point, we analyze the simulation time before and after the improvements proposed.

The investigation carried out in this paper is organized as follows: first, a full discussion of the proposed methodological and technical improvements, then the analyzed model adopted is discussed, and then the experiment performed to evaluate the time improvements is described. Finally, some conclusions are drawn to highlight possible outcomes in this research area.

2. MATERIALS AND METHODS

In this investigation, to reduce the Monte Carlo simulation time, the selected improvements are variance reduction (methodological) and parallel computing (technical). Each of them has a wide family of possibilities and definitions; therefore, we specify the selection later with a brief discussion about, starting with the variance reduction.

2.1. Variance reduction

Within Monte Carlo methods, variance reduction is a strategy (by referring to how the random sample is generated or by using some known features of the sampled distribution) applied to increase the accuracy of estimates (e.g., the expected value, in our case) that can be obtained from a computer simulation as highlighted by Botev and Ridder [2].

Each random variable generated from a computer simulation has an associated variance or variability, commonly speaking, that determines the accuracy of the result. To achieve a statistically efficient simulation, that is, to obtain an estimate with smaller confidence intervals for the random variable analyzed, variance reduction techniques can be used.

The main techniques are common random numbers, antithetic variates, control variates, importance sampling, and stratified sampling according to the authors Botev and Ridder [2].

Variance reduction techniques can be commonly separated into groups. First, we have antithetic sampling (for each generated uniform random number defined between [0, 1], its complement is used to compensate large deviations from the mean), stratification (the generated sample is strategically

divided into groups), and common random numbers (usually used when it is necessary to compare two different distributions, for a given common random sample). All these methods improve the variance by sampling the values strategically.

On the contrary, we have conditioning and control variates. These methods use information from the sampler to bound the generated outputs.

Finally, the importance sampling method. This method changes where we take the sample values from, i.e. it intentionally over samples from some regions and then corrects this distortion by re-weighting.

All the techniques are well defined in the literature and possible to implement, but given the model specificities, easy implementation and understanding, the case selected is stratified sampling.

In statistics, stratified sampling is a technique for sampling a population whose essence is to divide the population into subpopulations.

The precision of the estimation of a population statistic from pseudo-random samples, for example the mean, likewise the sample size, depends on the variability between the generated population samples. Consequently, in addition to increasing the sample size, another possible way to increase the precision of the estimate could be by dividing the generated pseudo-random samples into groups, so that the variability within groups is minimal and the variability between groups is maximum. In this way, smaller samples can be selected from each of the groups formed. The groups formed are called strata and the process of stratum formation is known as stratification according to Singh and Singh Mangat [14].

When we are in the presence of simple stratified random sampling, under the assumption that samples from different strata are selected independently, each stratum can be treated as an independent population. The estimation of the stratified mean will be more efficient than the usual simple random sample mean if the variation between the stratum means is large enough in relation to the variation within the stratum. However, the gain in precision also depends on the method used to create the strata. According to Singh and Singh Mangat [14], other points that need to be considered are as follows:

- Determining the number of strata to be constructed (for this point the characteristics of the population guide the decision).
- Allocation of total sample size to different strata (usually all the strata have the same size, but the technique allows for strata with different sizes).
- The choice of strata (the selection is mainly guided by the modelled problem).

This method introduces a challenge, the optimum allocation (or disproportionate allocation). As we can deduce, the allocation depends on the problem to be modelled, therefore, in next sections we discuss the topic by computational experiments after the evaluated scenario is fully described.

Relevant in this investigation is the expected value and the variance of the expected value, because the first reflects the risk value estimation of the system evaluated, and the second define the number of simulations needed to get a robust estimation, both redefined later in the following sections according to the model formulation.

Therefore, as starting point according to Singh and Singh Mangat [14], we can define the mean (Equation 1) and variance (Equation 2) of stratified random sampling as follows:

$$\bar{y} = \sum_{h=1}^{L} W_h \bar{y}_h \tag{1}$$

$$s_{\bar{y}}^2 = \sum_{h=1}^{L} W_h^2 \left(\frac{N_h - n_h}{N_h n_h} \right) s_h^2 \tag{2}$$

where: N_h = total number of units in the stratum h;

- $f_h = n_h / N$ = sampling portion for the stratum h;
- n_h = number of units selected in the sample from the stratum h;
- $w_h = N_h / N$ = proportion of the population units falling in the stratum h;
- Y_{hi} = the value of study variable for the i-th unit in the stratum, $i = 1, 2, \ldots, N_h$;
- $Y_h = \sum_{i=1}^{N_h} Y_{hi}$ = stratum total for the estimation variable based on N_h units;

- $\overline{Y_h} = \frac{1}{N_h}\sum_{i=1}^{N_h} Y_{hi}$ = mean for the estimation variable in the stratum;
- $\overline{y_h} = \frac{1}{n_h}\sum_{i=1}^{n_h} y_{hi}$ = stratum sample mean for the estimation variable;
- $\sigma_h^2 = \frac{1}{N_h}\left(\sum_{i=1}^{N_h} Y_{hi}^2 - N_h \overline{Y_h}^2\right)$ = stratum variance based on N_h units;
- $S_h^2 = \left(\frac{N_h}{N_h - 1}\right)\sigma_h^2$ = stratum mean square based on N_h units;

$s_h^2 = \frac{1}{n_h - 1}\left(\sum_{i=1}^{N_h} y_{hi}^2 - n_h \overline{y_h}^2\right)$ = sample mean square based on n_h sample units drawn from the stratum.

2.2. Parallel computing

Once the proposal for methodological improvement has been defined, we jump to the technological improvement in this section.

Nowadays, parallel computing is relevant to solve complex problems in science and engineering in practical times. Parallelism is a technological strategy to speed up calculations that require a lot of time and memory. According to the Keyes [6], historically, parallelism had acceleration as its main objective, which was characterized by Amdahl's law.

Gustafson-Barsis' law is another perspective of the same idea according to Keyes [6]; in this case, the idea is to measure how to keep time constant when the complexity of the problem increases. This definition is described by scalability. This new law leads the concept of designing systems with an increasing number of processors, a law applied and followed conceptually in this research.

As we know, a system design with these features is expensive when the required number of processors is relatively high, and sometimes the profitability of the software does not exploit all the available power. For this reason, the trend of parallelism is moving toward the era of clusters. For example, networked desktop computer labs can be configured as clusters. Regardless, it is well known that progress continues, at the microprocessor level, parallelism is found in multicores and many cores as highlighted by Trobec et al [18].

At the tip of the iceberg, with the idea of raising parallelism to its maximum expression, are the ideas of Grid and Cloud. These structures centralize computing power and provide access through the network, while also minimizing power consumption. In addition, on the somewhat more distant horizon, there are paradigms such as quantum computing, optical computing, and chips working on biological structures, which have a potential for future parallel computing according to Trobec et al [18].

All the advances described before walk at the same time with the development of methods and algorithms for these systems. It is well known that without effort in the field of software, it is impossible to exploit the computational power of the super modern computers that exist and are available.

The design of efficient and robust algorithms is essential to efficiently solve complex scientific problems where parallelism is in principle necessary.

The continuous trend over time is to increase the number of cores on a single chip. Now a simple conventional desktop computer can have 16 cores. To take full advantage of available capabilities, new tools, new algorithms, and a new way of looking at the programming will be required.

Today, a standard operating system can run different tasks in parallel depending on the available cores. However, in cases of serial software programme, the program or code needs to be restructured and parallelized to take full advantage of the multi-core architecture.

Following the paradigm of parallel computing, the simulation model adopted in this paper (defined later in the following sections), is implemented in MATLAB [10], parallelized, and tested assessing the simulation times.

The reason we selected MATLAB to implement the model is because it allows us, given a well-defined parallel code sequence, to run the simulations using batch jobs and parallel loops at the same time, as shows Fig. 1.

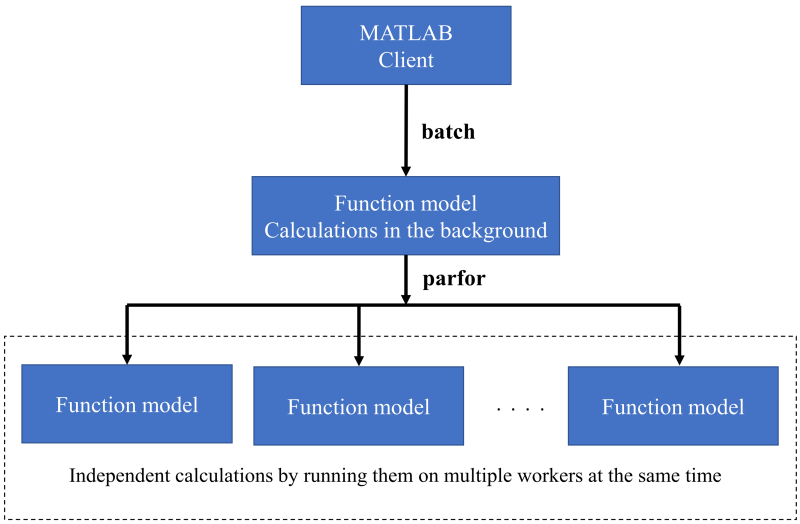

Fig. 1. Adopted MATLAB parallel jobs configuration (based on MathWorks online help)

The MATLAB architecture for parallel computing is frequently used by authors from different fields and is well accepted standard software for research purposes in the academy. The literature is full of applications with this tool.

According to the MATLAB online help, using batch jobs allows offloading the execution of long-running calculations in the background, increasing the efficiency of calculation time.

Additionally, Parallel Computing Toolbox of MATLAB enables interactive parallel computing and allows us to speed up independent calculations by running them on multiple workers at the same time. The idea is to use the *parfor* routines to run for-loop iterations in parallel on independent workers, as the online help describes.

The combined use of batch jobs and parallel loops exponentially increases the performance efficiency of the code and allows us to use all the computation resources on the computer to perform a task, as shown in Fig. 1.

When working interactively in a MATLAB session, we can download the job to a MATLAB work session to run as a batch job, which means we can create a group of workers for their batch job.

The workers can run on the same machine as the client or on a remote cluster machine. With that said, the code implemented in MATLAB can also be run on a cluster if needed.

2.3. Improvements proposed

The main outcome of the investigation is to evidence the reduction of the simulation times, given a parameterized scenario and a scheduling solution, for the adopted model, through the application of the methodological and technological improvements described in the sections 2.1 and 2.2.

As a general view of the research conducted, Fig. 2 summarizes the idea proposed, leaving clearly defined the experiment discussed in the following sections.

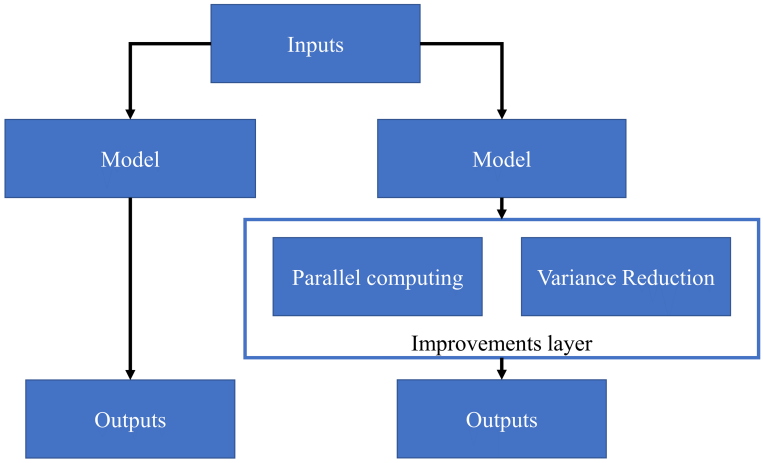

Fig. 2. Improvements proposed

3. MODEL DESCRIPTION

The model definition under study in this paper is taken from Szpytko and Salgado Duarte [16, 17]. The first reference describes the connection between the raw information sources and the model variables, and the second reference details the optimization model definition and the solution achieved given a parameterized scenario (optimal maintenance scheduling).

As a general description of the model used, we can say that the subject of the mathematical model is to manage the logistics-maintenance process of overhead type cranes operated in critical systems.

A hot rolling mill system in a steel plant, mainly supported by critical overhead cranes operating under hazard conditions and running continuously, was selected as a study case.

The model output (requested as an engineering solution) is an optimal risk-oriented maintenance scheduling for critical overhead cranes.

The model input (created as a formal structured information) is a digital database with historical information related to operation, maintenance-logistics, and management processes of the system.

The optimization model supporting the risk-oriented maintenance scheduling can be defined as stochastic, no-linear, with bounded constraints.

The model inputs are a SCADA (Supervisory Control And Data Acquisition) and SAP (Systems, Applications & Products in Data Processing) systems. Once the inputs are in place, the optimization algorithm proposes maintenance scheduling scenarios to be evaluated in the model proposed by Szpytko and Salgado Duarte [17] (risk oriented) given the corresponding restrictions. The best solution is the scenario with lower risk, and at the same time, the final output of the model.

From the database structure, the proposed optimization model takes the input variables and parameters. The model objective is to minimize the conditional expected value of the convolution function defined as $E[R]$ (risk indicator Capacity Loss), between the overhead crane capacity distribution function of the steel plant affected for the maintenance scheduling defined as X and the necessary load capacity distribution function defined as Y (see reference Szpytko and Salgado Duarte [17], model description).

In the model adopted, the time needed to obtain the optimal solution (maintenance scheduling) depends on the number of evaluations on the objective function (because a heuristic algorithm is used) and within each evaluation, on the number of simulations needed to achieve an accurate estimate of the conditional expected value of the convolution function (the variance of the expected value define

the accuracy of the estimation, and therefore the number of simulations needed to achieve a desired error).

Saying that, Fig. 3 shows an architecture diagram of the model flow and the improvements layers proposed (methodological and technical) to improve the simulation time.

The variance reduction technique layer impacts the computation of the conditional expected value by the Monte Carlo method given a proposed scenario of maintenance scheduling by the optimization algorithm.

In the case of the parallel computing layer, each scenario proposed by the optimization algorithm is independent (each maintenance scheduling scenario), therefore, can be evaluated independently.

Consequentially, the logic of the parallel computing improvement is to evaluate block-by-block scenarios depending on the available CPU in the computer.

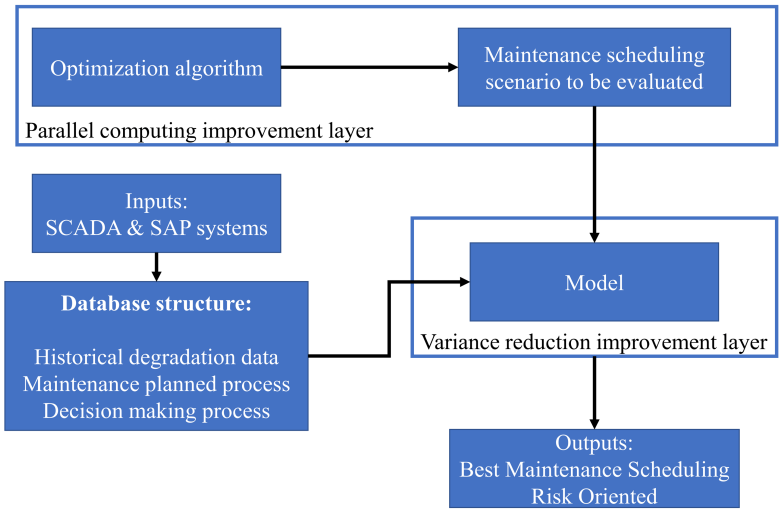

Fig. 3. Model flow with improvements

4. RESULT AND DISCUSSION

Given the starting point, the previous two cited references Szpytko and Salgado Duarte [16, 17] and a general overview description of the model, additional definitions are added in this section to complete how we improve the simulation times applying the layers proposed in practice.

The parallel computing layer is mainly oriented to the implementation and has no direct implication for the model definition, but the variance reduction technique layer has an impact on the definition of the conditional expected value, which is why it is discussed in depth in this section.

To estimate the conditional expected value of the risk function defined on Szpytko and Salgado Duarte [16], we must know the system distribution function. In this case the construction is possible by sampling random values from individual distribution functions (overhead crane individual distribution functions depend on the historical degradation data, fitting process, and the reliability block diagram system structure), and then following the definitions described on Szpytko and Salgado Duarte [16], the system distribution function can be computed by Monte Carlo method.

As we declare before, the complexity and dimension of the system analyzed is large. Therefore, the simulation time is an issue.

Equations 3, 3a and 3b state the criteria for robust expected value estimation (optimization target) and the clear dependency on number of simulations and the standard deviation of the conditional expected value:

$$Capacity\ Loss = E[R] \pm \beta E[R] \tag{3}$$

$$Capacity\ Loss = E[R] \pm \frac{\sigma[R]}{E[R] \cdot \sqrt{N}} E[R] \tag{3a}$$

$$Capacity\ Loss = E[R] \pm \frac{\sigma[R]}{\sqrt{N}} \tag{3b}$$

To compensate for an unnecessary increase in the number of simulations to achieve an accurate estimate (time-consuming), variance reduction techniques are a way to reduce the standard deviation by sampling strategically.

Relevant for this investigation and easy to implement in the model proposed by Szpytko and Salgado Duarte [16] is stratified sampling, as we declare above.

When stratified sampling are used, now adapting the definitions to the nomenclature of the model, the population mean (4) and variance (5) are given by equations 4 and 5:

$$E[R] = \frac{1}{K} \sum_{k=1}^{K} E[R]_k \tag{4}$$

$$V[R] = \frac{V[E[R]_k]}{N} \tag{5}$$

where $E[R]_k$ and $V[R]_k$ are given by the standard expected value and variance, respectively, for each k strata; K = number of strata (in our case, the number depends on the size of the generated sample); N = size of stratum k (estimated by experimental calculations).

Consequently, from previous definition, the Capacity Loss indicator estimated by Monte Carlo simulation, and defined in Equation 3, when stratified sampling are used, is defined by equations 6, 6a and 6b:

$$Capacity\ Loss = E[R] \pm \frac{\sigma[R]}{\sqrt{K}} \tag{6}$$

$$Capacity\ Loss = \frac{1}{K} \sum_{k=1}^{K} E[R]_k \pm \frac{\sqrt{\frac{V[E[R]_k]}{N}}}{\sqrt{K}} \tag{6a}$$

$$Capacity\ Loss = \frac{1}{K} \sum_{k=1}^{K} E[R]_k \pm \frac{\frac{\sigma[E[R]_k]}{\sqrt{N}}}{\sqrt{K}} \tag{6b}$$

Visible issues coming when this technique is used, the w_k estimation, where $w_k = N/K$ is the population weight of stratum k.

To estimate optimal weights, we use an experimental design and heuristic adjustments as a calibration (finding the optimal weights for the model) following some principles discussed on Mandies et al [9].

The experiment conducted is as follows: given a maintenance scheduling solution for the scenario, taken from Szpytko and Salgado Duarte [17], we compute by Monte Carlo method the conditional expected value (optimization target) and the error β continuously until we achieved the desired robust estimation without any stratification, pure Monte Carlo, saving in the end the string of random numbers needed, then with the same string of random number saved (for reproducibility reasons), by steps, we changed N_k (size of stratum k) and we recalculated the conditional expected value and the error β continuously every N_k simulations until we achieved the desired robust estimation.

Table 1 summarizes the results of the experiment conducted. Table 1 shows the number of simulations needed to achieve the same error for each stratum tested.

Based on Table 1 results and making a conservative decision in order to avoid singularities and be aware about the fact that we never know the sampled population size needed when the error β is fixed, we decided a size of stratum equal to 5, therefore, during the conditional expected value estimation of the following analysis, we grouped every 5 simulations and we estimated the mean, then we estimated the expected value (optimization target) and the error β continuously every group of 5 samples until we achieved the desired robust estimation.

Table 1 clearly shows the decreases in the number of simulations needed (from 1290 to 230 simulations).

Table 1

Calibration results

Cases	Expected Value	Error	Simulations Needed
Without	1042.45	0.009995	1290
$N_i = 2$	1046.51	0.009992	650
$N_i = 3$	1048.38	0.009998	420
$N_i = 4$	1058.69	0.009941	312
$N_i = 5$	**1059.10**	**0.009904**	**230**
$N_i = 10$	1059.98	0.009452	170
$N_i = 15$	1080.42	0.008750	120
$N_i = 20$	1080.42	0.009678	120
$N_i = 25$	1069.98	0.008906	100
$N_i = 30$	1133.33	0.001383	60
$N_i = 35$	1087.93	0.009099	70

All the estimations so far are performed with a personal computer i5 5250U 1.6 GHz CPU. The solution time of the optimization problem in this investigation depends on the number of samples (*Ns*) needed to ensure the error in each scenario simulated and the number of evaluations (*E*) in the objective function to reach the solution.

According with results from Szpytko and Salgado Duarte [17], the average time of one Monte Carlo simulation is $[(1.955813 \pm 0.072131) \cdot Ns \cdot \underline{E}]$ seconds.

Visible so far how the number of simulations is reduced by variance reduction (see Table 1). On the other hands, in the case of the number of evaluations, when parallel computing is used, in our case, four scenarios are evaluated at the same time because the features of the i5 5250U 1.6 GHz CPU allow us to run four computations at the same time (see Fig. 3).

Once the decision of the size of the strata has been made, we run the model by moving sensitive variables, in this case the efficiency indicator (see model description in reference Szpytko and Salgado Duarte [16]), evaluating the simulation time with and without applying the improvement layers.

As we declare here, using parallel computing reduces by four the time needed to assess the scenario. Once both improvements are described, below we define the experiment to analyze the simulation times. The experiment consists of changing the efficiency indicator between 80% to 85% and assessing the simulation times needed with, and without variance reduction techniques implemented. In the case of parallel computing the combined impact is divided into all the cases performed by four (i5 5250U 1.6 GHz CPU, has four CPU).

The experiment results shown in Table 2 and Table 3 corroborate the hypothesis of this research. Visible in Table 2 the reduction of the simulation times, only the variance reduction technique decreases the simulation time in almost five times less (depending on the scenario evaluated), and with the addition of the parallel computing, the reduction in time will happen to almost twenty times less.

On the other hand, Table 3 shows for the same scenarios, the estimation of the risk indicator, indicating estimations within confidence intervals in all the cases (robust estimation assumed, $\varepsilon = 0.01$). Of particular interest are the results in Table 2, because when the system is more reliable (loss capacity lower, see Table 3 also) the variance reduction technique is more pertinent, therefore the improvements behave well on a highly reliable system.

Fig. 4 adds other results to confirm the reduction of the simulation times (visual impact). Given a maintenance scheduling scenario, we simulated by Monte Carlo the evaluation without and with stratification, estimating the expected value in both cases.

In this experiment, the efficiency indicator is 75%. Fig. 4 shows the decreases in the simulations needed to achieve a robust estimation when the variance reduction techniques are used.

Table 2

Simulation time summary

Scenario	Time (seconds)	
Efficiency	Without variance reduction	With variance reduction
80%	8609.88	1846.05
81%	7289.31	1571.35
82%	7977.12	2052.64
83%	4767.47	896.93
84%	3531.06	722.11
85%	2129.12	696.81

Table 3

Expected value estimation with and without variance reduction

Scenario	Loss capacity (tons)	
Efficiency	Without variance reduction	With variance reduction
80%	111.91	111.47
81%	152.70	151.42
82%	212.68	214.56
83%	307.54	313.85
84%	472.55	477.45
85%	782.20	788.51

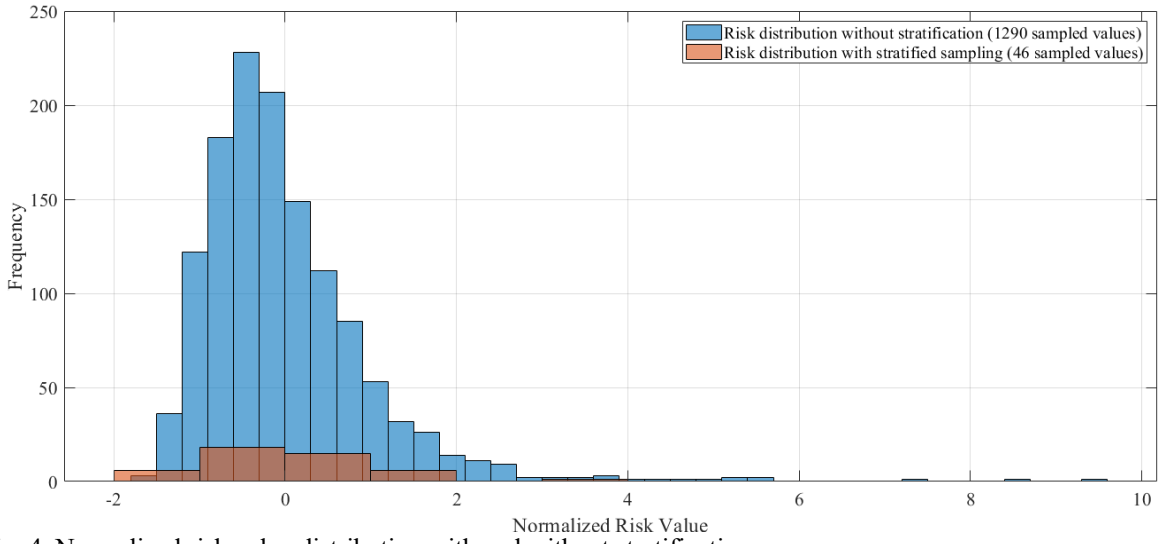

Fig. 4. Normalized risk value distribution with and without stratification

On the contrary, the model analyzed is a highly reliable system as we stated above, so if we do not apply variance reduction techniques, the number of simulations increases exponentially when the system is more reliable.

Simulation-based approaches are powerful for modelling stochastic processes with complex compound functions, but the time to simulate these processes can be a limitation with the current computing power. This paper evidence strategies to decrease the impact of the limitation and clearly improvements for Monte Carlo simulation approaches.

5. CONCLUSION

The paper presents a combination of parallel computing and variance reduction techniques and shows how they can help to reduce the computer simulation time in a case study of practical importance, the decision-making and control processes of overhead cranes operating optimally under critical conditions.

The paper describes two possible improvements to reduce the simulation times of the risk assessment approach, showing consistent results in all the scenarios. The above results confirm, in the presented scenarios, the time reductions of a simulation-based approach. The presented improvements open the way to keep using simulations approach but in a robust way.

Through Cloud Computing (Parallel Computing), complex NP problems based on simulations that seek solutions in exhaustive and complex decision-making diagrams, designed by humans, Cyber-Physical Systems can find fast and practical solutions, as the case presented in this paper, eliminating the time limitation in the search of solutions.

Acknowledgement

The work has been financially supported by the Polish Ministry of Education and Science.

References

1. Alebrant Mendes, A. & Weber Lorenzoni, M. Analysis and optimization of periodic inspection intervals in cold standby systems using Monte Carlo simulation. *Journal of Manufacturing Systems*. 2018. Vol. 49. P. 121-130.
2. Botev, Z. & Ridder, A. *Variance Reduction*. Wiley StatsRef: Statistics Reference Online: 1-6. 2017.
3. Dieker, A.B. & Ghosh, S. & Squillante, M.S. Optimal resource capacity management for stochastic networks. *Operations Research*. 2017. Vol. 65(1). P. 221-241.
4. Fan, C. & Liao, Y. & Zhou, G. & Zhou, X. & Ding, Y. Improving cooling load prediction reliability for HVAC system using Monte-Carlo simulation to deal with uncertainties in input variables. *Energy & Buildings.* 2020. Vol. 226. No 110372.
5. Hubbard, D. & Samuelson, D.A. *Modeling Without Measurements*. 2009. OR/MS Today: 28-33.
6. Keyes, D. Parallel numerical algorithms: An introduction. In: *Parallel Numerical Algorithms*. Keyes D.E. & Sameh, A. & Venkatakrishnan, V. (Eds.). Kluwer Academic Publisher. Norwell, MA. 1997.
7. Kroese, D.P. & Brereton, T. & Taimre, T. & Botev, Z.I. Why the Monte Carlo method is so important today. *WIREs Comput Stat*. 2014. Vol. 6(6). P. 386-392.
8. Leahu, H. & Mandjes, M. & Oprescu, A.M. A numerical approach to stability of multiclass queueing networks. IEEE. *Transactions on Automatic Control*. 2017. Vol. 62(10). P. 5478-5484.
9. Mandjes, M. & Patch, B. & Walton, N.S. Detecting Markov chain instability: a Monte Carlo approach. *Stochastic Systems*. 2017. Vol. 7(2). P. 289-314.
10. MATLAB. version 9.7.9.1319299 (R2019b). Natick, Massachusetts: The MathWorks Inc. 2010.
11. Ozkan, O. & Kilic, S.A. Monte Carlo Simulation for Reliability Estimation of Logistics and Supply Chain Networks. *IFAC PapersOnLine*. 2019. Vol. 52(13). P. 2080-2085.
12. Rausch, C. & Nahangi, M. & Haas, C. & Liang, W. Monte Carlo simulation for tolerance analysis in prefabrication and offsite construction. *Automation in Construction*. 2019. Vol. 103. P. 300-314.
13. Salgado Duarte, Y. & Szpytko, J. & del Castillo Serpa, A.M. Monte Carlo simulation model to coordinate the preventive maintenance scheduling of generating units in isolated distributed Power Systems. *Electric Power Systems Research.* 2020. Vol. 182. No. 106237.

14. Singh, R. & Singh Mangat, N. *Elements of Survey Sampling. Springer Science + Business Media Dordrecht*. 1996.
15. Spall, J.C. Estimation via Markov Chain Monte Carlo. *IEEE Control Systems Magazine*. 2003. Vol. 23(2). P. 34-45.
16. Szpytko, J. & Salgado Duarte, Y. Integrated maintenance platform for critical cranes under operation: Database for maintenance purposes. *Proceeding of 4th IFAC Workshop on Advanced Maintenance Engineering. Services and Technologies* Sept. 10-11, 2020. Cambridge, UK, IFAC PapersOnLine. 2020. Vol. 53(3). P. 167-172.
17. Szpytko, J. & Salgado Duarte, Y. Exploitation Efficiency System of Crane based on Risk Management. *Proceeding of International Conference on Innovative Intelligent Industrial Production and Logistics.* IN4PL 2020. 2-4 November 2020. ISBN: 978-989-758-476-3.
18. Trobec, R. & Vajtersic, M. & Zinterhof, P. (Eds.). *Parallel Computing Numerics, Applications, and Trends. Springer.* Dordrecht Heidelberg London New York. 2009.

Received 12.12.2019; accepted in revised form 11.05.2021