Copyright Notice:

Materials published by Intaver Institute Inc. may not be published elsewhere without prior written consent of Intaver Institute Inc. Requests for permission to reproduce published materials should state where and how the material will be used.

Improving Performance of Monte Carlo Simulations for Project Portfolios Risk Analysis

Intaver Institute Inc.

sales@intaver.com
www.intaver.com

Project portfolios tend to be large and very complex and any comprehensive analysis of them mirrors this complexity. Project portfolio Monte Carlo risk analysis requires additional consideration as they can have a large number projects with interdependencies between projects. This article explains a number of modeling methods, which are intended to improve performance of Monte Carlo simulations.

Challenges of Portfolio Risk Analysis

On 24 June 1812, 450,000 men (up to 685,000 according to some estimates) of the Napoleon's Grande Armée, the largest army assembled up to that point in European history, crossed the river Neman and headed towards Moscow (Britten Austin 2000). Under the guidance of Napoleon, this huge army moved rapidly towards the heart of Russia, Moscow. On September 1812 Napoleon forces entered Moscow, which had been set ablaze by retreating Muscovites. Napoleon's army had made it to Moscow, but the city was empty and spoils few. After a month, Napoleon realized that he could not sustain the invasion and decided a retreat was in order. On the long march back to France the army faced extremely cold weather, starvation, disease, persistent attacks from the Russian army and local peasants, and inevitable desertions, which led to great losses. By November 1812, of the original half a million or so soldiers, only 27,000 fit remained in the Grand Armée, with 380,000 dead and 100,000 captured. The campaign ended on the December 14, 1812 when the last French solder left Russia and spelled the beginning of the end for Napoleon.

To get a better understanding or what actually happened, the map below provides a good illustration (Figure 1). It shows how the invading Grand Armée gradually shrank as it moved across Russia towards Moscow (grey line, from left to right) and back (black line, from right to left). The size of army is proportional to the width of the line. Temperatures during the retreat are plotted on the lower graph.

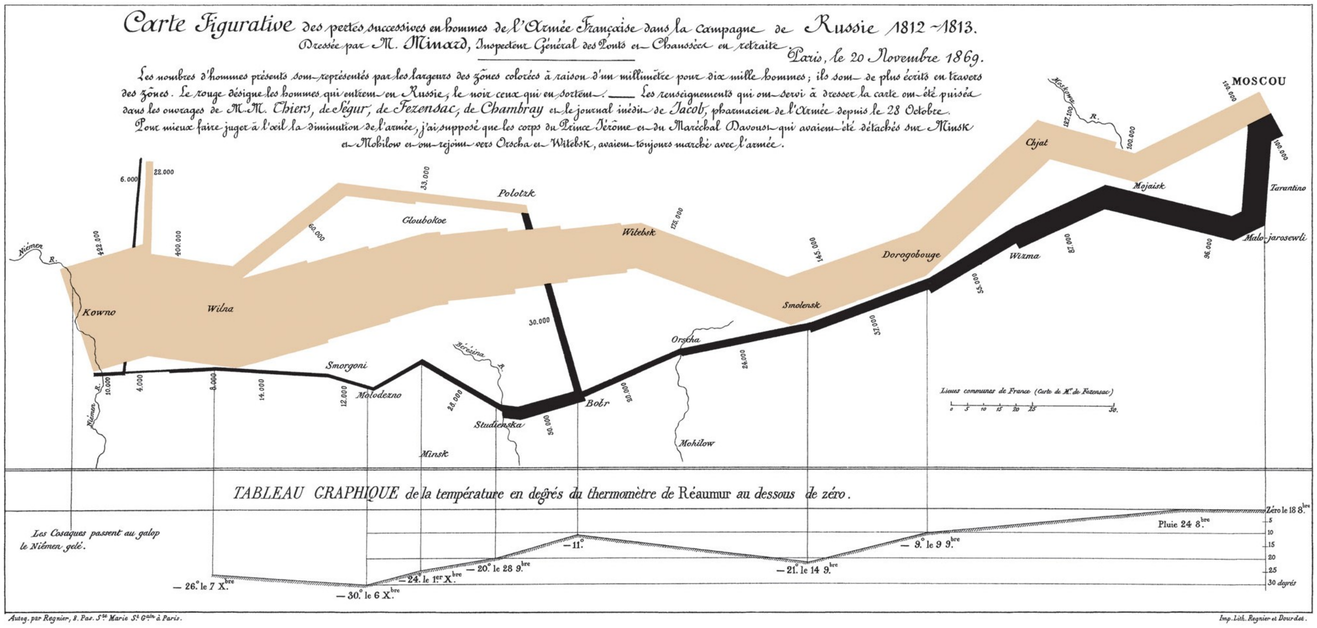

Figure 1. Map of Napoleon's invasion to Russia and retreat.

Napoleon lost this war without losing a single major battle, he lost because his army was not ready for a prolonged military operation across a vast land and in harsh winter conditions. Any war, especially a war on this scale is a complex project portfolio and each project has many different activities which can be impacted by many different risks. The war of 1812 was an unbelievably complex portfolio. To put this another way, could you imagine taking the entire population of Memphis, Tennessee and ordering them to go by foot or horseback (if they were so fortunate) to Chicago, Illinois over poorly maintained roads and extreme weather? Add in occasional attacks by the local population and Napoleon was planning to live off the land as he had in Central Europe, you can start to grasp the logistical nightmare Napoleon was undertaking. In the sparsely populated rural Russia of the early 19thy century, supplies, including food and forage for horses were scarce and could not keep up with the marching troops. Each battle, each maneuver, each delivery a supply convoy was essentially a project or even program and in many cases, some of the activities have predecessors and successors in other projects. For example, before a battle, you must have supplies, but the delivery of supplies=es depends on how the security and dependability of the supply chain. This in turn, depends on the army's ability to control the territory that they have already captured. Looking at it this way, we can start to see event chains emerging, where one risk can trigger risks across the project portfolio. In the 21st century, with all of our technology, with just in time communications etc., it is still extremely challenging to manage these type of portfolio risks, in 1812 it was virtually impossible.

Unfortunately for Napoleon's army, any high-level portfolio risk analysis that was possible given the knowledge at the time was done incorrectly. How do we know? If had done it properly, he surely would not have attempted such an undertaking and he would turned his gaze away from Russian and looked elsewhere for glory and riche. 19th century European history might have been quite different if Napoleon had understood the value of portfolio risk analysis.

Monte Carlo Simulation of Project Portfolio

Given what we have learned from Napoleon 1812 invasion, it becomes clear that in many cases, projects cannot be analyzed and managed in isolation as they are often have cross dependencies with other projects. These dependencies are:

- Start and finish times of tasks in one project depend on the start and finish times of tasks in different project
- Shared resources across project portfolio that may impact the duration and cost of the projects
- Risks can be part of complex event chains that affect multiple projects.

Because of these and other issues, it is sometimes the case that an integrated Monte Carlo simulation of the entire portfolio needs to be performed. As portfolios can be very large and complex, the simulations can consume a lot of computer resources and time. Even if you are able to run a simulation, the results are also that much more difficult to analyze. In most cases and abundance of data does not lead to better analysis, but the opposite. For example, let us assume that Napoleon miraculously obtained a modern computer with the most recent analysis software and was able to perform an analysis of his Russian campaign. Through this analysis, he discovered that his chance of successfully invading Russia and seizing Moscow was strongly correlated with hundreds of tasks from multiple projects related to early the preparations and the first phase of the invasion of Russia. Napoleon and his generals would have been overwhelmed by the amount of information making it impossible to understand what risks are critical and what the key decisions they would need to make. The solution is to create simplified models of the portfolio for the purposes of risk analysis.

How to Simplify Project Schedules for Risk Analysis

The idea behind schedule simplification is to create an equivalent schedule that uses a smaller number of tasks, but contains the essential elements (start times, finish times, cost, precedent network) such that it has enough detail to capture the project objectives, without too much detail. This can be a fine balancing act, but there are ways that it can be done, without losing essential data.

The easiest way to simplify a project schedule is to perform a simulation at a specified level of activities and above. The project schedule on Figure 2 has 4 levels (links between tasks are not shown). It is possible to perform risk analysis using only the tasks situated at levels 1, 2, and 3. Costs, resources, and uncertainties for tasks on level 4 will be allocated to their summary tasks on level 3. There are problems with this approach. It is possible that we could have a risk that is assigned to a single task on level 4. Would it be possible to assign its summary task on level 3? Well, it depends on the risk impact and the task to which it is assigned. The same is true for resources and costs, it may not be possible to create a credible risk model if we roll up the data. So, we can see simplifying the project model may not be equivalent to the original more detailed version.

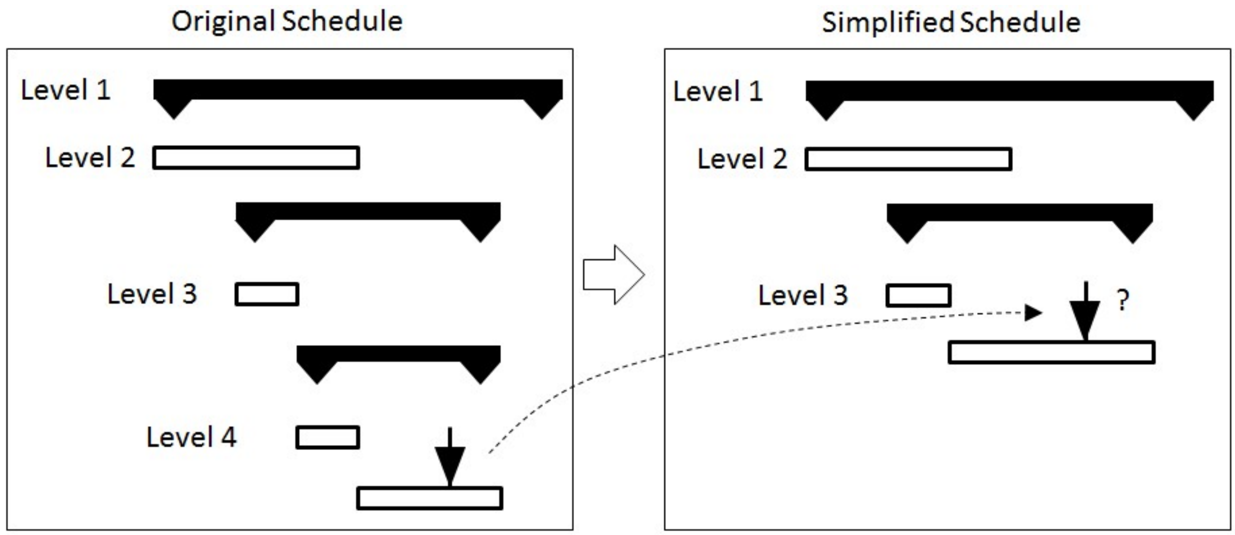

Figure 2. Simplifying project schedule base on level

Another way to simplify project schedules is to use schedule segmentation or "condensation". There is a similar idea in mathematics referred to as "*Model order reduction (MOR)*" (Schilders, et al, 2008). It uses a number of techniques to reduce the complexity of mathematical models in numerical simulations. One such techniques is called static condensation (Paz and Leigh, 2001). It is used in structural analysis using the finite element method. Let's assume that you have a structure with many trusses. You need to determine how the structure will react under certain loads. For example, you need to calculate a displacement for different points, including many points within the truss's span (Figure 3). These type of structural calculation models can be very large. So, to simplify the process, a separate truss is calculated only once. You then create an equivalent model of the truss with only two points, where this truss is connected to the walls. The stiffness of the equivalent truss is equal to the stiffness of the original truss. In this way the analysis of whole structure is much simpler, because all internal points of the trusses are excluded. The calculation model of the structure will be "condensed". After whole analysis is completed it is possible to determine displacements in these internal points.

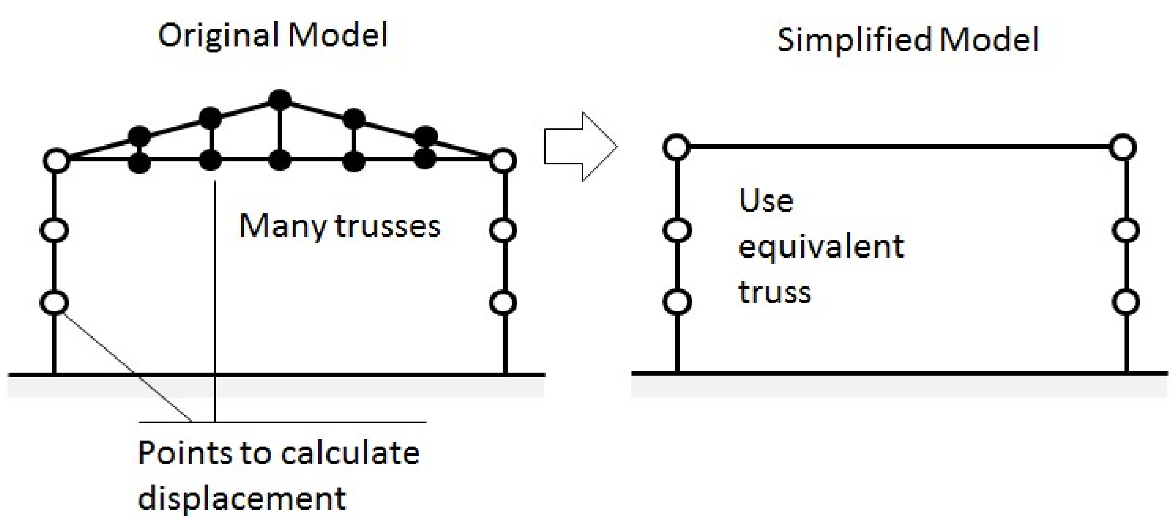

Figure 3. Static condensation process in structural mechanics

The idea of condensation can be applied to project scheduling as well.

Most schedules contain repeated group of tasks. The whole group may start in different times, but they will have the same risks, costs and resources. For example, in Napoleons campaign, they had many groups of tasks related "Battalion marches from point A point B" that cab repeated multiple times for the different battalions. The idea is that these group of tasks can be "condensed". Here is how we do it: (Figure 4):

1. Select a group of tasks or projects within a portfolio that have no more than one "entry point" or predecessors and don't have any constraints, such as "Must Start On", or "Start No Later Than".
2. Run a Monte Carlo simulation of this group and generate a statistical distribution for cost and duration.
3. Convert these statistical distributions into discrete or custom distributions for the whole group. Separate input distributions can be created for cost and duration.
4. Insert this group into the portfolio project schedule as one task with the statistical distributions created in Step 3. The simulation of portfolio will be accelerated because there will be less tasks will involved in the calculation using critical path method. Essentially this approach "condenses" project schedule.

Another advantage of this is that after the simulation is complete, you can "drill down" to the group of tasks and determine the statistical distributions for cost and duration of individual tasks within the group.

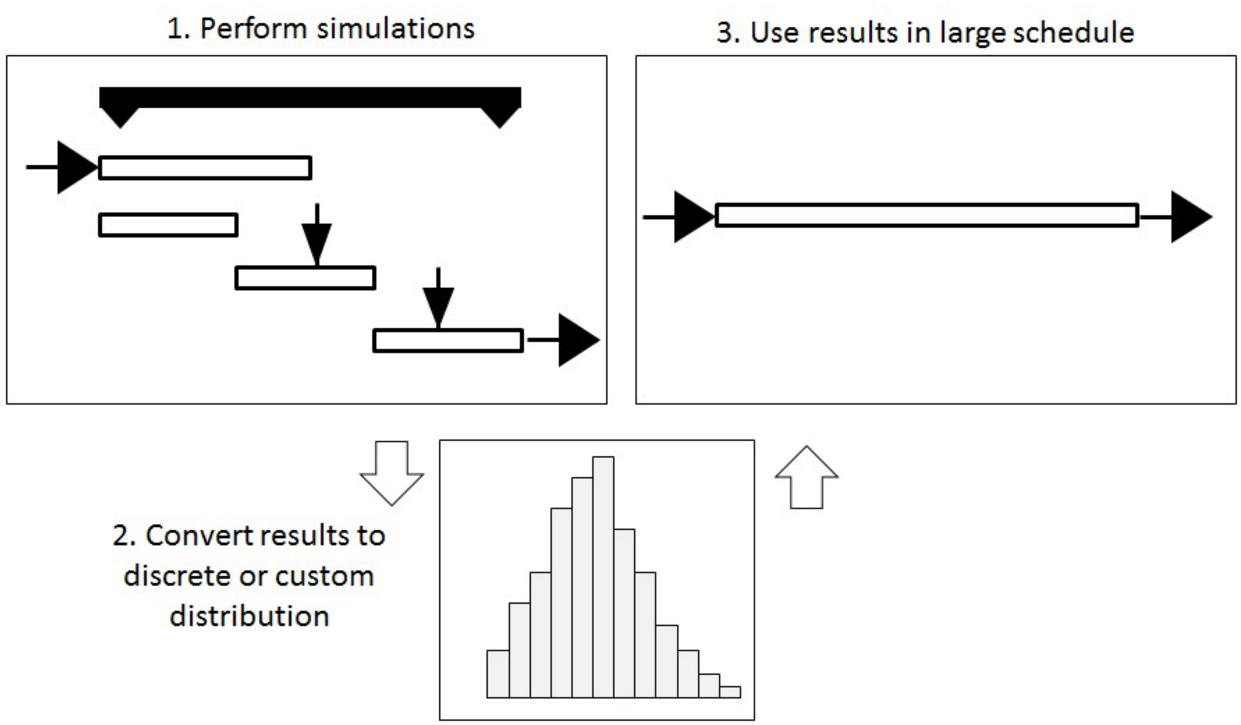

Figure 4. "Condensation of the schedule"

An equivalent schedule for groups of tasks or projects can be used in two ways:

1. You have a project portfolio in which some projects are in execution and others are in the planning stage. Let's assume that you performed an analysis of the same portfolio many times using actual project performance. Certain projects or group of tasks will not change because they have not started yet, however their start time can be shifted due to its predecessor's

actuals. In this case, you can create equivalent tasks for these and refer to these equivalent tasks each time to perform new calculation of the portfolio. This is very useful approach when different people in an organization perform analysis on different projects. In this way, you or other people can separately perform valid analyses of portfolios and/or its projects (group of tasks). Using the invasion of Russia as an example, lower level officers could perform analysis of their projects, while Napoleon's marshals and senior generals would perform analysis of higher level projects. Napoleon himself could perform analysis of the portfolio.

2. You have repeated projects or group of tasks within a portfolio. In this case, when you calculate a portfolio, you can refer to the equivalent schedules for the project or group of tasks. To efficiently use this approach, you need to find as many repeated group of tasks with the same task duration, cost, risks and uncertainties, and constraints as possible. To ensure that you have these repeated segments you need to minimize number of constraints and link tasks only when it is meaningful. It is especially important to minimize number of links between projects in your portfolio, because it will not only cause problems with segmentation, but also significantly complicate your portfolio schedule.

There is yet another way to accelerate calculations and condense project schedules. Assume that you have a number of sequential tasks. Each of these tasks has a statistical distribution for task duration but do not have any risks or constraints. In some cases, there is an analytical solution for adding statistical distribution together as the statistical distribution for the sum of all tasks is a function of individual statistical distribution.

Why Do We Need to Improve Performance of Monte Carlo Simulations?

There are a number of reasons why the calculation performance of Monte Carlo simulations can be important:

1. Large integrated project portfolios can consist of tens of thousands of tasks. Sometimes it is impossible to significantly simplify these portfolios and therefore it would require the ability to perform Monte Carlo simulation on a large number of tasks. Add resources, cost, and risks and the simulation may take hours.
2. Scenarios are integral to the decision-making framework. In theory, the number of scenarios can be quite significant because it is hard to accommodate all risk responses in one schedule. If a single Monte Carlo simulation can take a long time, running multiple scenarios can add to significant time to your analysis. .
3. Analysis can be repeated multiple times during project execution based on actual project performance data. This can also be time consuming.

Therefore, vendors of project risk analysis software are in a race to develop more efficient algorithms for Monte Carlo simulations. If you are going to be using Monte Carlo simulation software properly, it is important to be familiar with these algorithms.

Parallel Monte Carlo Iterations

You may wonder, where is the bottleneck that has the most impact on the time required to run Monte Carlo simulations? It is the critical path calculation. While, mathematically it is not very complex algorithm as it is essentially it is based on the formula: start time + duration = finish time. The bottleneck is created as it must be run hundreds if not thousands of times. The other calculations related to Monte Carlo simulations, such as the calculation of individual task durations based on risks assigned to his task, or the generation of frequency and cumulative probability plots take much less time. Therefore, software developers of the software are looking to optimize their critical path calculation algorithms. One way to do it is to use multi-processing calculation and graphic processing units.

The computer that is sitting on your desk is a marvel of technology and is capable of doing many this at once. One of the reasons for this is multiple processing cores. These cores can perform parallel calculations of each Monte Carlo iteration. All calculations can be done independently and then results merged at the end. As a result, calculation performance can be increased dramatically. Here is how it works:

1. Create a precedence network model with uncertainties and risks and calculate one Monte Carlo iteration. Please note that a precedence network model for calculation is not the same thing as project schedule. It only includes information necessary to perform calculation using critical path method: start and finish times of tasks, their dependences, information about summary tasks, constraints, and other inputs for calculation. During this step, the task durations and cost are adjusted based on risks. For example, if on this iteration a risk occurs and task duration increases, this new duration will be set in the calculation model.
2. Generate the precedence network model in the computer multiple times and save them to the computer's memory. The number of copies depends on the total available computer memory and number of processing cores. The number of copies of the precedence network should be at least 4 times greater than number of processing cores because some cores unavailable due to unrelated processing tasks such as running the operating system. Send all the models from the memory to the different cores of the computer. Each core will perform critical path calculations.
3. When all calculations are completed, the results are saved back into the memory. Results from each precedence network model are saved in a global data set representing the simulation results of all the tasks and the project.
4. Clean and release memory.
5. Repeat steps from 1 to 5 until desirable number of iterations is achieved or Monte Carlo simulation process converges.
6. Process and generate the results of the simulation from the memory.

Another way to improve performance is Monte Carlo simulation is to use a graphic processing unit or GPU. The GPU is a specialized computer chip that is used for graphic processing and comes with your computer's graphic card. They are particularly useful for generating 3D graphics as they are extremely fast at performing specific mathematical operations GPUs can be used to perform Monte Carlo simulations for large as long as the calculation algorithm is programmed certain way. It is even possible to use multiple graphic cards to do such parallel computation. This process is called General-purpose computing on graphics processing units (GPGPU). This approach is used widely for different scientific applications (Thompson et al 2002) and may well be the future of project portfolio simulations

References

Britten Austin, P. 2000, 1812: Napoleon's Invasion of Russia, Greenhill Books

Paz M., Leigh W. 2001. Integrated Matrix Analysis of Structures. Theory and Computation. Springer.

Schilders W.H., Van der Vorst, H., Rommes J. 2008. Model Order Reduction: Theory, Research Aspects and Applications (Mathematics in Industry). Springer. 2008th Edition

Thompson C., Hahn S., and Oskin M., 2002. Using modern graphics architecture for general-purpose computing: A framework and analysis. In SIGGRAPH/International Symposium on Microarchitecture, Turkey, Nov. 2002.