IMPACT HTA

Improved methods and actionable tools for enhancing HTA

Deliverable 2.2

Validation of the DICE modelling technique against existing methods and speeding up execution of the simulation engine

AJ McGuire, M Woods, JJ Caro

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 779312. The results presented reflect the author's views and not those of the European Commission.

Contents

Background

Most health technology assessments (HTA) require use of a model to integrate the available data and make projections over time about the health and economic impacts of the technology being assessed. HTA agencies, particularly those that accept submissions from industry, require that these modelling frameworks be transparent (in the sense that all aspects of the model and underlying assumptions can be viewed and tested) and this implies that they must be formulated in a software that the assessor is familiar with and can execute. As MS Excel® is widely available and most researchers in HTA use it routinely, it has become a *de facto* standard for implementing these models.

Unfortunately, this has led to gross oversimplification of these models because of the perceived limitations of spreadsheets for handling calculations that happen over time rather than all at once. Thus, the cohort state-transition ("Markov") technique remains extremely prevalent, despite its well-known limitations for this purpose[i]. A Markov model considers the HTA problem in terms of the health "states" that people can be in and the "transitions" from one state to another that happen over time[ii]. Although time passes continuously in reality, the Markov model simplifies this by computing the transitions repeatedly in cycles of time. Further departures from reality are introduced by the limitation to states as many of the things that happen during a disease and its treatment are actually events[iii]. Markov modelers have developed various workarounds and approximations to try to address these issues, but these often lead to unnecessary complication and loss of transparency. Moreover, they make debugging the model and reviewing the final product more difficult and, regrettably, often introduce errors.

Recently, a new approach to conceptualizing and implementing models has been introduced, Discretely Integrated Condition Event (DICE) simulation[iv]. This technique allows the modeler to design and structure whatever model is felt to be most appropriate to the problem in hand and the implementation is fully in MS Excel® with no hidden components. Needless-to-say, HTA agencies charged with making appraisals with considerable potential impact on patients, their families and the health care system are somewhat reluctant to adopt a novel technique without reassurance that is works correctly and produces estimates that can be trusted.

As part of WP2, thus, a validation of the technique was undertaken by replicating existing models implemented in the more traditional manner. In addition, as some of these models involve billions of calculations, particularly when undertaking probabilistic sensitivity analyses, and MS Excel® can be slow to execute these more complex models, there was strong interest in speeding up the execution engine. In this report, the validation of DICE simulation process and speeding up of the engine are described and the validation of the first model is detailed.

DICE simulation

DICE provides a modelling approach that can be used to create anything from survival partition models[v], through Markov models, to unconstrained discrete event simulations[vi]. DICE conceptualizes a model in terms of three fundamental components. All of the information processed by the model, including its outputs, is represented in *conditions*. Each condition is one piece of information (e.g., age, mortality rate, relative risk, time, etc.). The values stored in conditions can change during the simulation and these changes are specified in *events*. These events can reflect real occurrences (e.g., diagnosis of cancer, start of treatment, death); or they can be specified purely for modelling purposes (e.g., start simulating, end simulation, update values). The required change to a condition's value is specified in an *expression* that states what is to happen (e.g., age increases by 1, change treatment to none, accrue QALYs by adding the product of utility and time since last accrual). A set of these expressions that are to happen at a particular point in time specifies an event. The events are "triggered" whenever it is appropriate to do so during the simulation (e.g., the Start event at time zero; the End event at time horizon; the one-year update after one year; the treatment switch when a side-effect occurs).

Both the conditions and the events are tabulated. The conditions table lists each condition, its starting value (if known) and the current value. The all-events table lists each event, its initial time, its next time of occurrence and the name of the Excel table that contains its set of expression. For each event, there is a named table that lists in each row the Condition (or output) that is to be modified and the expression specifying what that change is. As the expression is in text (i.e., without an equal sign in front), Excel does not execute it.

Execution of a DICE model requires a *discrete integrator* that reads the tables and, beginning with the Start event, processes each row in each event table at the appropriate time. As the text expressions in each event table row respect Excel® syntax, the discrete integrator interprets it as an Excel formula it can compute by inserting the missing equal sign. The sequential processing required by the model is implemented by introducing the equal sign in each row, one at a time, and deleting before introducing it in the next row. Once all rows in a table are processed, the macro finds the next event (the one with the lowest scheduled time), processes its table rows and continues this process until the End event, at which point it stops and reports the contents of those conditions that have been specified as outputs. The discrete integrator is a short macro written in Visual Basic for Applications (VBA)[vii].

Validation against existing models

To validate the DICE methodology, the team responsible for WP2 selected 4 models that had been built by the NICE Guidelines group. These models were chosen to reflect a range of complexity. The plan was to rebuild each model as a DICE simulation and then compare the results produced by the new version with the original ones. In addition, the time taken to rebuild each model and the difficulties encountered were recorded.

The process followed was:

2017

June 14: Initial DICE Training with NICE in London

2018

- Planning TC
- Planning F2F meeting
- Meeting with NICE for final selection of models for validation
- Presentation and discussion of first model (breast cancer chemoprevention)

2019

- Discussion of second model (hyperphosphatemia in chronic kidney disease)
- Discussion of third model (medical cannabis)
- Discussion of fourth model (management of post-operative Crohn’s disease)

2020

- Review of validation of hyperphosphatemia in chronic kidney disease model
- Review of validation of medical cannabis model
- Review of validation of management of post-operative Crohn’s disease model
- Publication of paper documenting validation of first model

Validation of breast cancer chemoprevention model

The NICE model

The HTA problem was to compare no chemoprevention with chemoprevention with one of three drugs in postmenopausal women who have risk factors for breast cancer. NICE created a Markov model to assess this use of chemoprevention in women at moderate to high risk for breast cancer[viii]. This model had been updated several times and the most recent update was selected for this validation. The model that was developed was a fairly standard Markov model (Figure 1). Three states were considered: no breast cancer, breast cancer, and dead. The model was implemented in MS Excel® in two separate workbooks (one for each risk stratum: high, moderate) to keep the size of the files manageable. Each workbook contained one worksheet per chemoprevention strategy, organized into four sets of columns, one for each age stratum. Time, in annual cycles, was represented by the rows. Additional sets of columns on each worksheet computed weighted averages of the annual results across ages, using the proportion of the population in each age group as weights. A half cycle correction was applied in another set of columns. Treatment discontinuation at one year was applied in additional columns that created weighted averages of chemoprevention results and no chemoprevention for the women stopping treatment early. Costs were accrued in a separate worksheet. Inputs were spread across several additional worksheets. A results worksheet reported the overall results.

To simplify the comparisons, the original model was applied separately to women at moderate risk and at high risk, and in four age intervals (age is a major modifier of risk). Thus, the Markov model was replicated 32 times. The risk of breast cancer without chemoprevention was estimated for each of the resulting eight strata. The efficacy of each chemoprevention agent in terms of reducing the risk of breast cancer was estimated from clinical trials and inputted as an agent-specific relative-risk. The risk of three possible adverse effects (AEs)—endometrial cancer, thromboembolism (TE), and fractures (hip, wrist, vertebral, or other)—were estimated, assuming they were constant over time. Non-compliance with chemoprevention was incorporated by assuming that 50% of the women would discontinue after one year. It was assumed that upon discontinuing chemoprevention, all of its benefits would be lost but also the risk of AEs would return to zero (not to a background rate). Based on treatment guidelines, it was assumed that the remaining 50% of women would continue chemoprevention for an additional four years (i.e., total of five years of treatment) and then discontinue. At this point, however, the benefits would be retained throughout the time horizon. The risks of cancer AEs were set to zero after that, but the fracture risk persisted for another five years. The costs of chemoprevention, its associated monitoring, and a one-time cost of breast cancer were considered. No costs were assigned to AEs, diagnostic testing, or other aspects and no effects on quality of life were considered. Background mortality was obtained by age stratum from the UK life table but no effect on was applied for breast cancer or AEs.

This model was selected by the team at NICE because it was more complex than a standard Markov model and, thus, would allow for testing of the various features of DICE and comparison with a traditional build. It was chosen as the first exercise because it was still reasonably straightforward and would allow for completion of the first validation in a reasonable not so complex that it sufficiently complex that it would test the theorized benefits of DICE over acceptable short time without substantial effort by the team at NICE (who were engaged in building several new models at the time). Construction of the original model took approximately 50 hours of work by a team at NICE (not including the months spent understanding the problem, obtaining inputs, and doing the statistical analyses). For this first exercise, there was no attempt made to modify the original design – faithful replication was the goal.

The scheduled treatment stop at five years, the noncompliance after one year and the persistence of fracture risk for an additional five years did not fit well into the Markov structure. These were addressed in the model via various IF statements and weighted averages. The complexity of the implementation made it difficult to review and understand the structure, greatly increased the challenge of tracing inputs and events and debugging of calculations, increasing the potential for error. Various desirable analyses were not done (e.g., structural sensitivity and scenario analysis). Probabilistic sensitivity analyses were done via an external macro.

The DICE implementation

To validate the DICE simulation approach, the breast cancer chemoprevention Markov model described above was rebuilt as a DICE simulation. No changes to the design of the original model were made. During the re-implementation, some errors were found in the original model. These were corrected in the original. Otherwise, it would have been necessary to try to reproduce the errors in the DICE implementation to be able to validate that it produced the same results. Forcing the DICE version to reproduce the original errors would have been extremely difficult because DICE is designed to avoid precisely those kinds of errors in formula linkages.

The implementation of the DICE version took 4.5 hours, compared to more than 50 hours for the original. An additional 7 hours were spent verifying the implementation and making comparisons with the original results. The entire DICE model is specified in a 122.5 kilobyte MS Excel® workbook. This DICE model and the DICE macro can be downloaded from https://dice.impact-hta.eu/.

Events in the DICE model (Figure 2) and their initial times are tabulated in the All Events table (Table 1). The core of the model is a **Transition** event (as it is for all Markov models implemented in DICE). This **Transition** event (Table 2) has 3 rows that apply the transition probabilities (e.g., the proportion of the healthy state that develop breast cancer is obtained using the expression *Healthy*pBreastCa*). The number of women making each transition is stored in a corresponding condition (e.g., *HealthyBreastCa* stores the proportion who develop breast cancer in each cycle). Simple expressions then compute the new state memberships by adding transition in and subtracting transitions out (e.g., *Healthy* is updated to *Healthy-HealthyDead-HealthyBreastCa*, as no return to *Health* is allowed). The **Transition** event is to occur again after one *Cycle*. This happens until the time horizon is reached, at which point the **End** event is triggered.

The main Markov states are represented in DICE as conditions: *Healthy*, *BreastCa*, *Dead*. The value of each state condition is the proportion of the population (of number of women) in that state. Of note, the choice of conditions was driven by the original model. No attempt was made to optimize these for DICE as the objective was to show that the original could be faithfully replicated using DICE.

The additional components of the chemoprevention model are represented in DICE using additional events. A **non-adherence** event (Table 3) is triggered at the end of the first year (note that more flexible scheduling can be easily implemented). At this event, the proportion of women who stop chemoprevention early is applied by reducing the value of *OnTmt* to 50%. This is used to recalculate the affected transition probabilities and costs. The **StopTreatment** event (Table 4) is scheduled to happen after five years. In this event, the costs of chemoprevention and its monitoring are removed, and the risks of the cancer AEs are set to the untreated ones. Since fracture risk persists for another five years, it is not turned off. Instead, an event **EndFractureRisk** is scheduled to happen after an additional five years. At that event, the fracture risk is reset to that without chemoprevention.

In the **Start** event (Table 5) the simulation of each intervention is set up by assigning its RR, costs, and AE risks. In the **End** event the simulation is terminated. Another modelling event is **Valuate** (Table 6) where all the calculations for outputs are implemented (e.g., treatment costs accrued as *CostTmt+Cycle***CostInterv*, with *CostInterv* updated in the **Start** according to the chemoprevention strategy being simulated. The half-cycle correction is implemented by averaging the number of people alive at beginning and end of the cycle and using this to adjust the output. All outputs are specified in the Outputs table (Table 7).

The eight strata (four age groups for each of two risk levels) are implemented by defining a *Profile* for each one. Profiles are a DICE feature that allows specification of a set of factors that characterize a particular group. For this model, the profile simply lists the age group and risk level. These factors are copied into corresponding conditions, *AgeGrp* and *RiskStratum*, and their values are updated at the beginning of each replication. In the **Transition** event, the values of these conditions are used to select the corre4sponding probability of breast cancer (*pBreastCa*) without chemoprevention. *Age* increases by one each cycle (because mortality is applied annually) and the age group is updated when the next interval is reached.

Verification of the DICE model involved reviewing each DICE table to ensure that all Conditions and Outputs were correctly named, and all expressions were in proper Excel syntax (without the equal sign). The log produced during execution was inspected to ensure each expression was executed correctly.

Results of the validation

The DICE simulation correctly replicated the original Markov model but in a much simpler form. The build of the DICE model in standard MS Excel® took much less time and is specified in a much smaller file (the original totalled 17 mb) that is very easy to review and verify. The main drawback to using spreadsheets for modelling is their relatively long execution times. On a laptop running Office 365 32 bit with the current implementation of the DICE macro, this model takes 7.6 seconds to run a replication with 8 profiles and four interventions. This much quicker execution was achieved by developing a new engine that is much faster (version f).

Speeding up the engine

The original DICE engine (DICEd, “d” for demo) consisted of a very simple macro that read each event table, sequentially inserted an equal sign into each expression and stored the result in the corresponding condition. All calculations were done in the spreadsheets themselves. For this first version, it was felt important to be very transparent about the calculations and to allow people to see what was happening. While very didactic, this proved to be quite slow as the macro had to trigger the worksheets every time an expression was computed.

The next iteration of the DICE engine (DICEe, “e” for educational) retained some of the didactic aspects of the original version (e.g., the value of each condition was still stored on the worksheet) but expanded the macro to handle more of the execution of a simulation. This sped up the model considerably but for complex models and large probabilistic sensitivity analyses, it remained too slow.

As part of WP2, the engine was considerably modified to maximize the execution speed while remaining fully specified in MS Excel®. This new engine (DICEf, “f” for fast) minimizes the triggering of worksheets by parsing the expressions in the event tables and executing them in memory rather than using the Excel worksheets to do so. The Conditions table still lists all the conditions and their initial values but is no longer used to store their ongoing values as they are modified during execution. This is done in memory as well. Similarly, all outputs are accrued in memory and only reported at the end of a simulation. The engine now automatically handles many aspects of the execution, such as finding the next event and triggering it; and turning off each event after it has executed. A robust scenario handler, deterministic uncertainty analysis and probabilistic sensitivity analysis modules have been added. Many new utilities that help in model development, testing and examining the execution have been incorporated in an updated custom ribbon. Error checking and messaging have also been improved. The new engine is more than one thousand times faster than the original, bringing execution times for most models under one minute.

Conclusion

This exercise successfully demonstrated that DICE simulation can be used to create Markov models that provide the same results as the more traditional implementation but in a simpler, more transparent, more flexible structure. These advantages are not limited to Markov models as DICE simulation can be used to create any model from partitioned survival through hybrids that combine states and events, to full individual discrete event simulations. By providing the much faster engine as an open source platform, it is hoped that the advantages of DICE simulation can be more widely leveraged and modelers are encouraged to develop models with fewer limiting assumptions induced by the way they construct them..

[i] Siebert U, Alagoz O, Bayoumi AM, Jahn B, Owens DK, Cohen DJ, Kuntz KM. State-transition modeling: A report of the ISPOR-SMDM modeling good research practices task force-3. Value in Health 2012; 15:812-20.

[ii] Sonnenberg FA, Beck JR. Markov Models in Medical Decision Making: A Practical Guide. Medical Decision Making 1993; 13:322-38.

[iii] Caro JJ, Moller J. Advantages and disadvantages of discrete-event simulation for health economic analyses. Expert Review Pharmacoeconomics Outcomes Research 2016; 16: 327-9.

[iv] Caro JJ. Discretely Integrated Condition Event (DICE) Simulation for Pharmacoeconomics. PharmacoEconomics 2016; 34:665-72.

[v] Woods B, Sideris E, Palmer S, Latimer N, Soares M. NICE DSU Technical Support Document 19. Partitioned Survival Analysis for Decision Modelling in Health Care: A Critical Review. http://nicedsu.org.uk/technical-support-documents/partitioned-survival-analysis-tsd/. [Accessed January 4, 2020]

[vi] Caro JJ. Pharmacoeconomic analyses using discrete event simulation. PharmacoEconomics 2005; 23:323-32.

[vii] Möller J. Cooking Up a Transparent Model Following a DICE Recipe. PharmacoEconomics 2019; 37:1341-7.

[viii] National Institute for Health and Care Excellence. Addendum to Clinical Guideline 164, Familial breast cancer: Methods, evidence and recommendations. Appendix N Economic modelling report. https://www.ncbi.nlm.nih.gov/books/NBK550790/ [Accessed January 3, 2020]