ISPOR 2022
May 15-18, 2022
Washington, DC, USA Area and Virtual

Converting discrete-event simulation networks (DESnets) into DICE models

Carmen M. Yago, Francisco J. Díez
Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain
cyago@dia.uned.es, fjdiez@dia.uned.es

UNED

1 Introduction

- Discrete-event simulation (**DES**) is useful when:
  - time should be explicitly modelled
  - heterogeneous population
  - continuous scale for disease progression
  - past events must be remembered.

  Used for **cost-effectiveness analysis** (CEA) and multicriteria decision analysis (MCDA).
- **OpenMarkov**: open-source tool for probabilistic graphical models (Bayesian networks, influence diagrams…)
- Our goal: editing DESnets with OpenMarkov's graphical user interface, and converting them into **DICE** models

2 Discrete-event simulation networks: DESnets

Causal graph with four types of **nodes**:

- Decision ➢ choice among several alternatives
- Chance ➢ properties that are not under the direct control of the decision maker
- Payoff ➢ decision maker's values, such as costs and health outcomes
- Event ➢ their occurrence drive the evolution of the model

Links represent causal influences

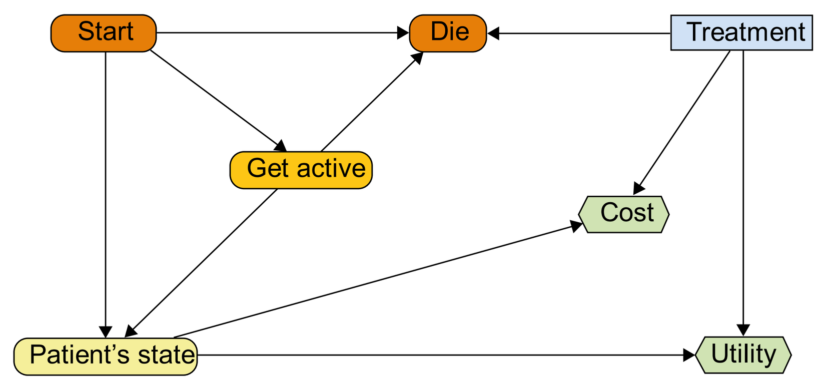
Potentials (functions): quantitative relations

- Value of the variables, depending on other variables
- Time of occurrence of events

❖ DESnets can be edited and evaluated with a plug-in that we implemented for OpenMarkov.

Example: disease progressing from latent to active and death (Soares and Canto e Castro, 2012)

- Patients may die at any time; the probability depends on health state
- Each transition is an event; time of occurrence follows an exponential distribution
- Two decision criteria: cost (£) and effectiveness (QALY)
- Cost and an effectiveness (per time unit) depend on health state and treatment applied.



Node Potential: Utility

Relation Type: Exact

Patient's state	latent	latent	active	active
Treatment	standard	new	standard	new
Utility	0.9	0.9	0.5	0.75

Node Potential: Die

Relation Type: TimeToEventTable   Distribution: Exponential

Treatment	standard	standard	new	new
Events	Start	Get active	Start	Get active
lambda	0.09	0.41	0.09	0.328

3 DICE

Discretely Integrated Condition Event (DICE) simulation:
a framework for cost-effectiveness analysis

Conditions

- Aspects that persist over time
- Have *levels*, which can change
- Affect events

Events

- Happen at a point in time
- Affect *conditions*
- Affect other events

Several implementations are possible. Most common: Excel add-on.

Widely accepted because it is transparent and simplifies the modeling process.

```
Do                                        'Loop through all events until End-event
    ThisEventName = UCase(EventTypes(rgeNextEvent, 4))
    ErrMsg = "Setting ThisEventName to " & ThisEventName
    ThisEvent = shEvents.ListObjects(ThisEventName).DataBodyRange.Value2    'changed for Tables
    ThisEventTime = rgeNextEventTime.Value 'For log only, adjust time after first row setting
    For i = 1 To UBound(ThisEvent, 1)
        If ThisEvent(i, 3) <> "" Then
            ErrMsg = "Setting " & ThisEvent(i, 2) & " to " & ThisEvent(i, 3)
            rgeCalcCell = "=" & ThisEvent(i, 3)
            Range(ThisEvent(i, 2)) = rgeCalcCell.Value
```

All Events

Name	CurEventTime	Initial Time To Event	Table
Die	99999999	Never	tblDie
GetActive	99999999	Never	tblGetActive
Start	99999999	Now	tblStart
End	5.678417052	TimeHorizon	tblEnd

Event: Start

Type	Name	Expression	Notes
Condition	Time	Start	To reset the clock to zero
Event	Start	Never	To avoid infinite loop
Output	Treatment	CHOOSE(IntervNum,"standard","new")	
Condition	TimeHorizon	50	
Condition	CostAccr	0	
Condition	CostPrevTime	0	
Condition	UtilityAccr	0	
Condition	UtilityPrevTime	0	
Output	dcCost	0	
Output	dcEffectiveness	0	
Condition	PatientState	VLOOKUP(RAND(), {0,"latent";1,"active"},2,TRUE)	
Condition	Parents	Treatment&PatientState	
Condition	Cost	SWITCH(Parents, "standardlatent",	
Condition	Utility	SWITCH(Parents, "standardlatent",	
Event	GetActive	(LN(1-RAND())/-0.12)+Time	
Event	Die	(LN(1-RAND())/-0.09)+Time	
Event	End	TimeHorizon	
Condition	NextEventTime	Min(CurEventTime)	Find next event time
Condition	NextEvent	Match(NextEventTime,CurEventTime,0)	Find next event

4 Method: conversion algorithm

First stage: set up DICE elements from the DESnet

- One *output* for each decision criterion and for each decision node
- One *condition* for each chance or payoff node
- One *event table* for each event node
- One *condition* for the time horizon
- Auxiliary *conditions* for cumulative values and flow control
- One *event table* "End"

Second stage: lay out the simulation flow

- For each event in the DESnet, add to the *event table* the expressions that, when the event is fired,
  - accrue the payoffs
  - update the *conditions*, and
  - compute the time of occurrence of child events

Third stage: assemble the *outputs*

- Add expressions to the "End" *event table* for computing the final outputs by summing up the payoffs for each decision criterion

❖ This algorithm is available in the plug-in for OpenMarkov.

5 Example: converting the former DESnet

First stage

Setting up the *outputs*:

- Decision node
- Decision criteria

Setting up the *conditions*:

- Time horizon
- Chance and payoff nodes
- Auxiliary *conditions*

Setting up the *event tables*:

- Event nodes
- End event

Name
Treatment
dcEffectiveness
dcCost
RepNum

Name
Die
GetActive
Start
End

Name
ID
Time
TimeHorizon
IntervNum
PatientState
Cost
CostAccr
CostPrevTime
Utility
UtilityAccr
Parents
UtilityPrevTime
NextEventTime
NextEvent

Second stage

- Accrue the payoffs
- Update simulation state
- Child events

Event: GetActive (name: tblGetActive)

Type	Name	Expression
Condition	Time	GetActive
Event	GetActive	Never
Condition	PatientState	VLOOKUP(RAND(), {0,"latent";0,"active"},2,TRUE)
Output	Treatment	CHOOSE(IntervNum,"standard","new")
Condition	Parents	Treatment&PatientState
Condition	CostAccr	CostAccr + (Time -CostPrevTime)*Cost
Condition	CostPrevTime	Time
Condition	Cost	SWITCH(Parents, "standardlatent", 4000,"standardactive",7000,"newlatent", 4000,"newactive",8500)
Condition	UtilityAccr	UtilityAccr + (Time -UtilityPrevTime)*Utility
Condition	UtilityPrevTime	Time
Condition	Utility	SWITCH(Parents, "standardlatent", 0.9,"standardactive",0.5,"newlatent", 0.9,"newactive",0.75)
Event	Die	SWITCH(Treatment, "standard", (LN(1-RAND())/-0.41)+Time, "new", (LN(1-RAND())/-0.328)+Time)
Condition	NextEventTime	Min(CurEventTime)
Condition	NextEvent	Match(NextEventTime,CurEventTime,0)

Third stage

Event: End

Type	Name	Expression
Condition	Time	End
Condition	CostAccr	CostAccr + (Time - CostPrevTime)*Cost
Condition	UtilityAccr	UtilityAccr + (Time - UtilityPrevTime)*Utility
Output	**dcCost**	**dcCost + CostAccr**
Output	**dcEffectiveness**	**dcEffectiveness + UtilityAccr**

ICER comparison (evaluated for N=100,000 patients)	
DESnet	8,285 £/QALY
DICE	8,270 £/QALY

References

1. Caro JJ (2016) Discretely Integrated Condition Event (DICE) simulation for pharmacoeconomics. *PharmacoEconomics* 34:665–672. https://doi.org/10.1007/s40273-016-0394-z
2. Soares MO, Castro LC e (2012) Continuous time simulation and discretized models for cost-effectiveness analysis. *PharmacoEconomics* 30:1101–17. https://doi.org/10.2165/11599380-000000000-00000
3. www.cisiad.uned.es
4. www.openmarkov.org
5. www.evidera.com/dice

Acknowledgments

- Grant PID2019-110686RB-I00. Spanish Ministry of Science and Innovation, European Regional Development Fund
- C.M.Y. is supported by the Ministry of Science and Innovation, and UNED.