Pemodelan Kejadian Diskrit Dinamis

Simulasi kejadian diskrit memodelkan sistem yang berubah sesuai waktu melalui suatu representasi dimana variabel status berubah secara langsung pada titik terpisah dalam waktu. Titik terpisah dalam waktu adalah keadaan dimana suatu kejadian terjadi. Kejadian didefinisikan sebagai kejadian langsung yang dapat mengubah status sistem. Meskipun simulasi kejadian diskrit dapat dilakukan secara manual, jumlah data yang harus disimpan dan dimanipulasi dalam dunia nyata mengharuskan penggunaan komputer digital.

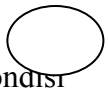
Simulasi jarum Buffon adalah simulasi kejadian-diskrit statis dalam artian bahwa simulasi itu terdiri dari serangkaian kejadian acak dimana setiap kejadian tidak dipengaruhi oleh kejadian sebelumnya. Waktu bukan bagian dari simulasi. Menjatuhkan jarum dilakukan berulang-ulang, memberikan perkiraan yang lebih baik akan probabilitas jarum menyentuh atau memotong garis, tapi simulasi akan tetap sama jika ke 3000 jarum dijatuhkan secara bersama-sama atau dijatuhkan satu demi satu sebanyak 3000 kali.

Lebih sering, simulasi bersifat dinamis, dimana interaksi antara kejadian acak dan waktu adalah bagian dari simulasi. Karena sifat dinamis ini, kita harus mengikuti nilai waktu tersimulasi selama simulasi dijalankan, dan kita juga perlu mekanisme mengembangkan waktu tersimulasi dari satu nilai ke nilai lainnya. Kita sebut variabel model simulasi yang memberikan nilai waktu tersimulasi saat ini dengan *simulation clock*. Unit waktu simulation clock tidak pernah dinyatakan secara eksplisit ketika pemrograman model dibuat dengan bahasa pemrograman umum seperti FORTRAN, Pascal atau C, dan diasumsikan dalam unit yang sama dengan parameter input. Juga, secara umum tidak ada hubungan antara waktu tersimulasi dengan waktu menjalankan simulasi dalam komputer.

Dua pendekatan prinsipal disarankan untuk menjalankan simulation clock yaitu next-event time advance dan fixed-increment time advance. Pendekatan pertama digunakan hampir semua bahasa simulasi dan bahasa umum (general purpose language), karena itu kita akan menggunakan pendekatan ini.
Dalam next-event time advance simulation clock diinisiasi dengan 0 dan waktu terjadinya kejadian di masa mendatang ditentukan. Simulation clock kemudian bertambah (maju) dengan waktu terjadinya kejadian berikutnya yang pertama, dimana pada suatu titik status sistem diperbaharui setelah terjadinya suatu kejadian, dan pengetahuan kita akan waktu kejadian berikutnya juga diperbaiki. Proses penambahan simulation clock berlanjut terus dari satu kejadian ke kejadian lainnya sampai kondisi penghentian yang sudah didefinisikan dipenuhi. Karena semua status berubah hanya pada waktu kejadian model simulasi kejadian-diskrit, periode tidak aktif diloncat dari waktu kejadian ke waktu kejadian. Harus diperhatikan bahwa loncatan berurutan simulation clock secara umum bervariasi dalam ukuran (tidak sama).

Representasi kejadian sistem

Model simulasi kejadian-diskrit dapat digambarkan sebagai sebuah model interaksi kejadian diskrit yang terjadi dalam sistem dan variabel status sistem. Interaksi ini dapat ditunjukkan dengan graf dimana simpul (verteks) menunjukkan kejadian dan cabang berarah (ruas) menunjukkan penyebab langsung terjadinya suatu kejadian hanya jika kondisi dipenuhi.

 Kejadian,  penghubung tidak terkondisi

 penghubung terkondisi

jika ruas yang menghubungkan dua kejadian adalah garis terputus, itu menunjukkan bahwa terjadinya satu kejadian bisa menyebabkan pembatalan kejadian lainnya. Jika ada penundaan antara dua kejadian terhubung, penundaan ditunjukkan pada ruas di antara kedua kejadian. Jika terjadinya suatu kejadian bersifat kondisional, referensi terhadap kejadian penting ditunjukkan ruas penghubung. Contoh :

diagram di atas menunjukkan bahwa kejadian i akan mengarah ke kejadian j setelah penundaan selama t dan kondisi 1 terpenuhi.
Sebagai contoh sistem kejadian-diskrit, perhatikan sekumpulan mesin yang ditangani sekelompok operator. Setiap mesin rusak perlu diperbaiki oleh operator. Setelah diperbaiki, mesin akan berfungsi kembali. Variabel status sistem adalah sebagai berikut:

M(i) status mesin i
0 = menunggu perbaikan
1 = sedang diperbaiki
2 = beroperasi
O(j) status operator
0 = menganggur
1 = sibuk

Kejadian diskrit yang terjadi adalah:

1(i) mesin i menunggu diperbaiki
M(i) diatur jadi 0
2(ij) operator j mulai memperbaiki mesin i
M(i) bernilai 1

O(j) bernilai 1

3(ij) operator j menyelesaikan perbaikan mesin i
O(j) bernilai 0

4(i) mesin i mulai beroperasi
M(i) bernilai 2

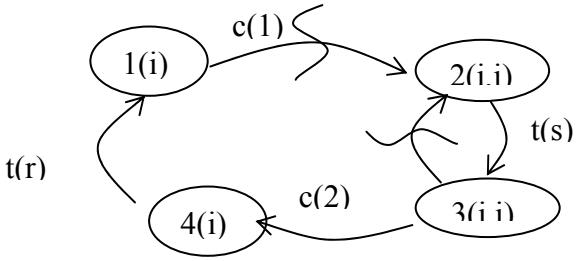
Kondisinya adalah:

C(1) beberapa O(j) = 0 (ada operator menganggur)
C(2) beberapa M(i) = 0 (ada mesin sedang mengunggu diperbaiki)

Penundaan kejadian :

T(r) waktu mesin dijalankan di antara panggilan perbaikan
T(s) waktu yang dibutuhkan untuk memperbaiki mesin

Graf kejadian sistem tersebut adalah:



Gambar 4. Graf kejadian sistem perbaikan mesin

Simulasi Monte Carlo

Simulasi Monte Carlo adalah proses menurunkan secara acak nilai variabel tidak pasti secara berulang-ulang untuk mensimulasikan model. Metode Monte Carlo karena itu merupakan teknik stokastik. Kita dapat menemukan metode Monte Carlo diaplikasikan dalam berbagai bidang, mulai dari ekonomi sampai fisika nuklir untuk pengaturan lalu lintas aliran. Tentu saja cara aplikasinya berbeda dari satu bidang ke bidang lainnya, dan ada banyak sekali himpunan bagian Monte Carlo meskipun dalam satu bidang yang sama. Hal yang menyamakan semua itu adalah bahwa percobaan Monte Carlo membangkitkan bilangan acak untuk memeriksa permasalahan.

Metode Monte Carlo dianggap sebagai penemuan dari Stanislaw Ulam, seorang matematikawan cemerlang yagn bekerja untuk John Von Neumann di proyek United State's Manhattan selama perang dunia II. Ulam adalah orang utama yang diketahui merancang bom hidrogen dengan Edward Teller tahun 1951. Dia menemukan metode

Monte Carlo tahun 1946 sewaktu memikirkan peluang memenangkan permainan kartu soliter. Percobaan jarum di atas merupakan contoh metode Monte Carlo.

Yang kita lakukan dalam lembar kerja adalah mendefinsikan nilai yang mungkin dengan distribusi peluang untuk setiap varaibel tidak tentu. Tipe distribusi yang dipilih didasarkan pada kondisi di sekeliling variabel.

Metode Monte Carlo, sebagaimana yang dipahami saat ini, melingkupi sampling statistik yagn digunakan untuk memperkirakan solusi permasalahan kuantitatif. Ulam tidak menciptakan sampling statistik. Metode ini sebelumnya digunakan untuk menyelesaikan permasalahan kuantitatif dengan proses fisik, seperti pelemparan dadu atau pengocokan kartu untuk menurunkan sampel. W.S. Gosset, yang mempublikasikan karyanya dengan nama "Student", secara acak menarik sampel ukuran jari tengah dari 3000 kriminal untuk mensimulasikan dua distribusi normal berhubungan. Dia mendiskusikan metode Monte Carlo dalam dua publikasinya (1908a dan 1908b). Kontribusi Ulam diakui dalam potensi penemuan baru komputer elektronik untuk mengotomasi penarikan sampel. Bekerja dengan John von Neuman dan Nicholas Metropolis, dia mengembangkan algoritma untuk implementasi komputer, juga mengeksplor alat transformasi permasalahan tidak acak ke dalam bentuk acak yang akan memfasilitasi solusinya melalui penarikan sampel acak. Nama Monte Carlo diberikan oleh Metropolis, dipublikasikan pertama sekali tahun 1949.

Nama simulasi Monte Carlo diberikan sesuai dengan nama salah satu kota di Monaco, yaitu Monte Carlo, tempat utama kasino yang mengandung permainan peluang (kesempatan). Permainan peluang seperti roda rolet, dadu dan mesin slot menunjukkan perilaku acak.

Perilaku acak dalam permainan peluang adalah sama dengan bagaimana simulasi memilih nilai variabel secara acak untuk mensimulasikan model. Ketika kita melempar dadu, kita tau bahwa yang akan muncul mungkin 1, 2, 3, 4, 5 atau 6, tapi kita tidak tau yang mana pastinya untuk lemparan tertentu. Hal itu sama dengan variabel yang mempunyai kisaran nilai diketahui tapi tidak diketahui nilai pasti untuk waktu atau kejadian tertentu.

Pemahaman metode Monte Carlo dapat dilakukan dengan memikirkan bahwa itu merupakan teknik umum integrasi numerikal. Setiap aplikasi metode Monet Carlo dapat direpresentasikan sebagai integral terbatas.

Misalkan kita ingin mengevaluasi integral terbatas multi-dimensional dari:

$$\psi = \int_0^1 \int_0^1 \cdots \int_0^1 f(u_1, u_2, \ldots, u_n) du_1 du_2 \cdots du_n = \int_{(0,1)^n} f(\boldsymbol{u}) d\boldsymbol{u} \qquad [1]$$

Kebanyakan integral dapat dikonversi ke dalam bentuk ini dengan perubahan variable yang sesuai, sehingga kita dapat mempertimbangkan ini menjadi aplikasi umum yang sesuai untuk metode Monte Carlo.

Integral mereprensentasikan permasalahan bukan acak, tetapi metode Monet Carlo memperkirakan solusi dengan memperkenalkan vector acak U yang berdistribusi normal dapa area integrasi. Dengan mengaplikasikan fungsi f ke U, kita mendapatkan variable acak f(U). Variabel acak f(U) mempunyai fekspektasi:

$$E[f(\boldsymbol{U})] = \int_{(0,1)^n} f(\boldsymbol{u})\ \phi(\boldsymbol{u})\ d\boldsymbol{u} \qquad [2]$$

Dimana ϕ adalah fungsi densitas peluang U. Karena ϕ sama dengan 1 pada area integrasi integration, maka ekspektasi di atas menjadi:

$$E[f(\boldsymbol{U})] = \int_{(0,1)^n} f(\boldsymbol{u})\ d\boldsymbol{u} \qquad [3]$$

Dengan membandingkan [1] dan [3] didapatkan ekspresi peluang untuk integral Ψ:

$$\Psi = E[f(\boldsymbol{U})] \qquad [4]$$

Sehingga variable acak $f(\boldsymbol{U})$ mempunyai rata-rata Ψ dan standar deviasi σ. Kita definisikan:

$$H = f(\boldsymbol{U}^{[1]}) \qquad [5]$$

Sebagai penduga tidak bias untuk Ψ dengan kesalahan standar (standar error) σ. Ini kurang konvensional, karena [5] adalah penduga yang tergantung pada sampel $\{\boldsymbol{U}^{[1]}\}$ dengan ukuran 1, tetapi itu tetap sebagai penduga valid.

Untuk memperkirakan Ψ dengan kesalahan standar kurang dari σ, mari kita generalisasikan penduga kita untuk mengakomodasi sample yang lebih besar $\{\boldsymbol{U}^{[1]}, \boldsymbol{U}^{[2]}, \ldots, \boldsymbol{U}^{[m]}\}$. Aplikasikan fungsi f ke setiap sample ini, akan menghasilkan sejumlah m variable acak independent and identically distributed (IID) $f(\boldsymbol{U}^{[1]}), f(\boldsymbol{U}^{[2]}), \ldots, f(\boldsymbol{U}^{[m]})$, dimana masing-masingnya mempunyai ekspektasi Ψ dan standar deviasi σ. Generalisasi [5]:

$$H = \frac{1}{m}\sum_{k=1}^{m} f(\boldsymbol{U}^{[k]}) \qquad [6]$$

Adalah penduga tidak bias untuk Ψ dengan kesalahan standar:

$$\frac{\sigma}{\sqrt{m}} \qquad [7]$$

Jika kita mempunyai realisasi $\{\boldsymbol{u}^{[1]}, \boldsymbol{u}^{[2]}, \ldots, \boldsymbol{u}^{[m]}\}$ untuk sample kita, kita dapat memperkirakan Ψ sebagai:

$$h = \frac{1}{m}\sum_{k=1}^{m} f(\boldsymbol{u}^{[k]}) \qquad [8]$$

Kita sebut [6] sebagai estiamtor Monte Carlo kasar. Persamaan [7] adalah penting karena dua alas an. Pertama, persamaan itu menunjukkan bahwa kesalahan standar analisis Monte Carlo menurun sebesar pangkat dua dari ukuran sampel. Kedua, kesalahan standar tidak tergantung pada dimensionalitas integral. Kebanyakan teknik integrasi numerical, seperti aturan trapezoidal atau metode Simpson, tergantung pada dimensionalitas. Ketika digeneralisasikan ke dimensi jamak, jumlah perhitungan yang dibutuhkan meningkat secara eksponensial sesuai dengan dimensionalitas integral. Metode Monet Carlo tidak tergantung dengan dimensionalitas.

Dalam analisis Monte Carlo, peningkatan jumlah sample akan mengurangi kesalahan standar, tapi itu akan bernilai mahal. Teknik reduksi ragam dapat digunakan untuk memperbaiki solusi. Teknik ini menggabungkan informasi tambahan tentang analisis secara langsung kedalam penduga. Hal ini memungkinkan penduga Monte Carlo lebih deterministik, dan karenanya mempunyai kesalahan standar lebih rendah. Teknik standar pengurangan ragam termasuk **antithetic variates**, **control variates**, **importance sampling**, dan **stratified sampling**.