

Discrete event simulation

Fabian Bastin

`fabian.bastin@umontreal.ca`

Université de Montréal – CIRRELT – IVADO – Fin-ML

Basic concepts

System collection d'entités qui agissent et interagissent afin d'accomplir une certaine fin logique.

L'état d'un système est la collection de variables nécessaires pour décrire un système à un instant particulier.

Model description simplifiée d'un système, dans le but d'évaluer sa performance ou l'effet de certaines décisions.

Simulation faire évoluer le modèle d'un système en fournissant les entrées appropriées, puis observer et analyser les résultats.

Two main system types: discrete and continuous. Deux types principaux de systèmes: discrets et continus.

Models

Physical vs mathematical model.

Analytical model, numerical model, model with simulation.

Reference textbook: A. M. Law, Simulation Modeling and Analysis, fifth edition, McGraw-Hill, USA, 2015.

Event approach

Hypothèse: système dans lequel les variables d'état ne peuvent changer qu'en un nombre dénombrable de points dans le temps.

Suite d'événements $e_0, e_1, e_2 \dots$, survenant aux instants

$$0 \leq t_0 \leq t_1 \leq t_2 \leq \dots$$

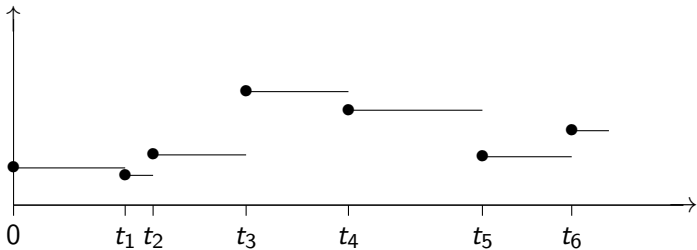
S_i : état du système immédiatement après e_i .

temps de la simulation: valeur courante de t_j .

(t_i, S_i) doit contenir assez d'information pour poursuivre la simulation (sauf les valeurs des variables aléatoires générées lors des événements e_j pour $j > i$).

Discrete events

state

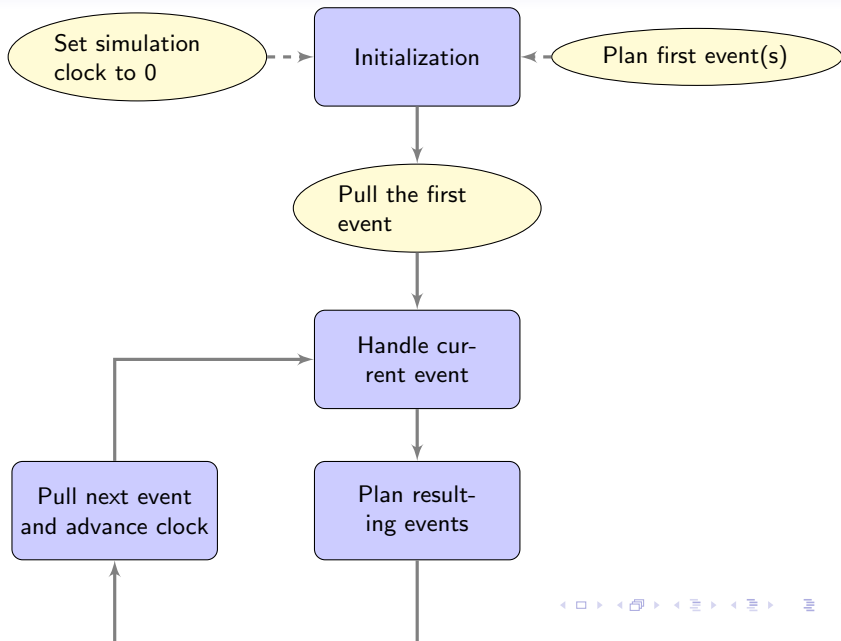


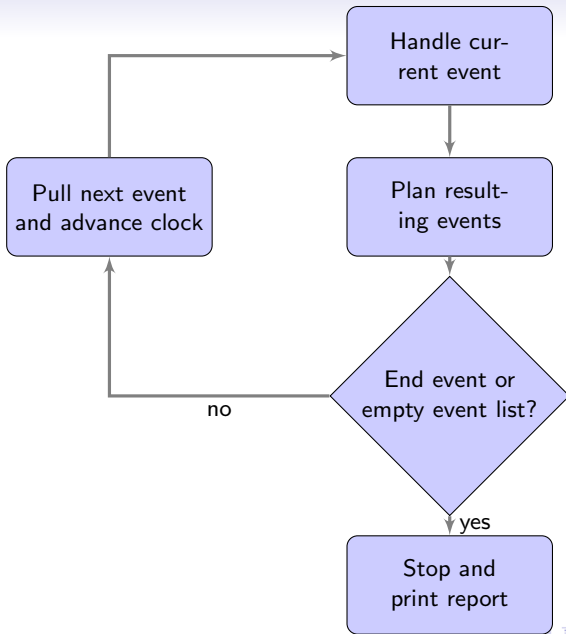
time

General framework

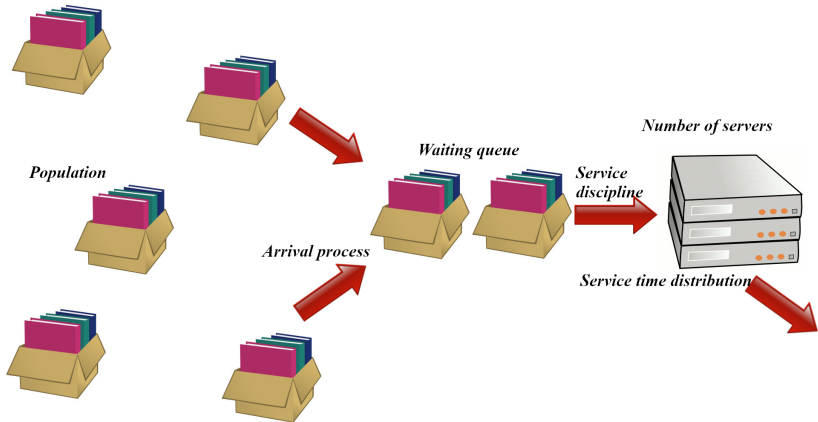
- Programme principal.
- Horloge de simulation.
- Liste d'événements.
- Routines de génération de nombres aléatoires.
- Compteurs statistiques.
- Génération de rapport.

Événement: action, schedule, cancel.





Queueing systems



Kendall notation

$$A/S/c/K/N/SD$$

- A Arrival process
- S Service time distribution
- c Number of servers
- K Waiting list maximum length
- N Population size
- SD Service discipline

Notation

G General

I Independent

M Markovian or memoryless

Defaults:

- Infinite buffer capacity
- Infinite population size
- FCFS (First-Come-First-Served) service discipline

Memoryless property:

$$P[X > x + a \mid X > a] = P[X > x]$$

Continuous: exponential distribution

Discrete: geometric distribution

Example: GI/G/1 queue

$$G/G/1 = G/G/1/\infty/\infty/FCFS$$

We also write $GI/G/1$ to stress that the service times are independent.

The clients arrive one by one, one being served at a time, under a FCFS priority, with

- S_i : service time of client i , with cdf G ;
- A_i : inter-arrival time between clients i and $i + 1$, with cdf F .

We assume S_i and A_i mutually independent.

Example: GI/G/1 queue

The first client arrives at time A_0 in an empty system.

Nous souhaitons simuler ce système pour une durée T et calculer l'attente moyenne par client et la longueur moyenne de la file d'attente. Les types d'événements sont

1. arrivée;
2. départ;
3. fin de la simulation.

Les variables aléatoires (indépendantes) à générer sont A_1, A_2, A_3, \dots et S_1, S_2, S_3, \dots .

Example: GI/G/1 queue

- W_i , le temps d'attente du client i ;
- $Q(t)$, la longueur de la file d'attente au temps t ;
- $N_c(t)$, le nombre de clients ayant débuté leur service durant l'intervalle $[0, t]$.

Supposons que nous voulons calculer, pour l'intervalle $[0, T]$, l'*attente moyenne* par client,

$$\overline{W}_{N_c(T)} = \frac{1}{N_c(T)} \sum_{i=1}^{N_c(T)} W_i;$$

et la *longueur moyenne* de la file d'attente:

$$\overline{Q}_T = \frac{1}{T} \int_0^T Q(t) dt.$$

Example: GI/G/1 queue

Cette dernière intégrale est facile à calculer par simulation.

Pour chaque *client*, nous créons un *objet* contenant l'instant d'arrivée et la durée de service.

Variables d'état: liste des clients en attente et la liste les clients en service.

+ horloge de simulation, compteurs statistiques (au besoin), *liste* des événements futurs prévus, par ordre chronologique, et une procédure pour chaque type d'événement.

Queueing system simulation

Arrivée Générer A selon F et prévoir une autre arrivée dans A unités de temps;

Créer le nouveau client et noter son instant d'arrivée;

Générer sa durée de service S selon G ;

Si (serveur est occupé) **alors**

Insérer ce client dans la liste des clients en attente;

sinon

Insérer le client dans la liste des clients en service;

Prévoir son départ dans S unités de temps;

Mise à jour des statistiques voulues.

Départ Enlever le client de la liste des clients en service;

Si la file d'attente n'est pas vide **alors**

Enlever le premier client de la liste d'attente;

L'insérer dans la liste des clients en service;

Récupérer son S et prévoir son départ dans S unités de temps;

Mise à jour des statistiques voulues.

Simulation

Démarrer la simulation:

- initialiser des variables et compteurs,
- prévoir la “Fin-de-la-Simulation” au temps T
- générer A selon F et prévoir la première “Arrivée” dans A unités de temps,
- lancement de la simulation.

L'exécution de la simulation consiste simplement à répéter la boucle suivante:

Répéter: exécuter le prochain événement dans la liste d'événements

jusqu'à: la liste des événements prévus est vide, ou bien un événement stoppe la simulation.

Simulation sans liste d'événements.

Pas toujours indispensable de recourir à l'approche par événements.

Exemple: récurrence de Lindley:

$$W_1 = 0, \quad W_{i+1} = \max(0, W_i + S_i - A_{i+1}).$$

Nous pouvons ainsi facilement simuler pour un nombre fixe de clients (au lieu d'un horizon de temps T fixe).

Le formulation du programme basée sur la récurrence de Lindsley revient à traiter un problème d'intégration sur $[0, 1]^t$.

Récurrance de Lindley

Supposons que nous souhaitions estimer $E[\overline{W}_{100}]$ par \overline{W}_{100} :

$$\overline{W}_{100} = \frac{1}{100} \sum_{i=1}^{100} W_i.$$

Il nous faut $A_1, S_1, A_2, S_2, \dots, A_{99}, S_{99}$. Si on pose $A_i = F^{-1}(U_{2i-1})$ et $S_i = G^{-1}(U_{2i})$, où $U_j, j = 1, \dots, 99$, sont des variables aléatoires uniformes sur $(0, 1)$, alors $\overline{W}_{100} = f(U_1, \dots, U_{198})$ (f étant de forme inconnue dans le cas présent).

Nous avons donc $t = 198$, i.e.,

$$E[\overline{W}_{100}] = \int_{[0,1]^{198}} f(\mathbf{u}) d\mathbf{u}.$$

Nombre aléatoire de clients

Par contre, si nous voulons simuler $\overline{W}_{N_c(T)}$, le nombre de clients $N_c(T)$ est aléatoire et non borné. Nous devrions donc choisir $t = \infty$ comme dimension.

Approche par processus

Un processus est une séquence temporellement ordonnée d'événements interreliés, séparés par certains intervalles de temps, qui décrit l'expérience entière d'une "entité" comme celle-ci évolue à travers un "système". Un système ou un modèle de simulation peut avoir différents types de processus.

Une routine sera associée à chaque processus du modèle, décrivant son histoire entière à travers le système.

Au contraire de l'approche par événements, une routine de processus contient explicitement le passage du temps simulé et a généralement de multiples points d'entrée.

Processes vs Events

Une simulation utilisant l'approche par processus évolue aussi au cours du temps en exécutant les événements dans l'ordre de leur occurrence.

En interne, les approches par événement et par processus sont donc très similaires.

Pour l'utilisateur, le raisonnement soit différent, et puisse parfois (mais pas toujours) être plus naturel pour l'approche par processus.