

# Progetto di Sistemi Complessi: Modelli e Simulazione

## Supermarket Simulation

Tonelli Lidia Lucrezia (m. 813114)  
Grassi Marco (m. 830694)  
Giudice Gianluca (m. 829664)

University of Milano Bicocca

Settembre 2021

- 1 Introduzione
- 2 Stato dell'arte
- 3 Descrizione del modello
  - Overview
  - Agenti del modello
- 4 Implementazione dei comportamenti e delle strategie
- 5 Simulazione
- 6 Conclusioni

# Introduzione

Il nostro progetto è un modello basato su agenti che **simula** il comportamento dei clienti in un supermercato durante le fasi di scelta della coda relativa a diversi tipi di casse.

Prenderemo in considerazione 3 tipi di casse diverse (standard, self-service e self-scan) e 2 tipi di scelte fatte dai clienti (scelta della coda, jockeying). L'utilizzo di code è fondamentale per gestire le grandi quantità di clienti.

## Obiettivo

Sperimentare diverse **configurazioni di casse** e **strategie di scelta della coda** per gestire in modo ottimale il flusso di clienti e ridurre al minimo il tempo d'attesa passato in coda.

Un supermercato è un **sistema complesso** in cui agiscono diverse entità, come clienti e casse.

Si verificano aspetti emergenti difficilmente prevedibili dovuti a diversi aspetti:

- Flusso di clienti in ingresso variabile
- Numero di prodotti che un cliente acquista
- Numero di casse aperte contemporaneamente
- Strategia di scelta della coda dei clienti
- Strategia di cambio della coda dei clienti

# Stato dell'arte

Il principale spunto per il modello è stato l'articolo *Data-driven simulation modeling of the checkout process in supermarkets: Insights for decision support in retail operations*<sup>1</sup>, che utilizza 5 strategie di scelta della coda confrontando i **tempi d'attesa medi** dei clienti.

---

<sup>1</sup>Antczak, Tomasz and Weron, Rafał and Zabawa, Jacek, 2020

# Estensioni

Abbiamo voluto estendere il modello introducendo nuovi concetti:

- **Jockeying**<sup>2</sup>: quando un cliente sta attendendo in coda, confronta i tempi d'attesa della propria coda con quelli delle code vicine e può decidere di spostarsi di conseguenza.
- **Code parallele e N-Fork**<sup>3</sup>: vogliamo indagare sull'effetto della disposizione delle code sui tempi di attesa medi, questa può essere **parallela**, se ogni cassa ha una coda dedicata, oppure **N-Fork**, se c'è un'unica coda condivisa.

---

<sup>2</sup>*On jockeying in queues*, E. Koenigsberg, 1966

<sup>3</sup>*Methods for improving efficiency of queuing systems*, Yanagisawa, D and Suma, Y and Tanaka, Y and Tomoeda, A and Ohtsuka, K and Nishinari, K, 2011

# Estensioni

- **Casse self-scan:** l'articolo sopra citato prende in considerazione solo 2 tipi di casse, le casse **standard** e le casse **self-service**. Nel modello sono presenti le casse **self-scan**, presenti attualmente in molti supermercati, che permettono di scannerizzare i prodotti in fase di spesa e rendere la fase di pagamento molto più veloce.
- **Simulazione non deterministica:** per far emergere comportamenti non banali nel supermercato e rendere più realistiche le simulazioni sono state aggiunte alcune variabili probabilistiche.

# Approccio ad agenti

- **Sistema multiagente** sviluppato in Python con il framework Mesa
  - **Ambiente:** supermercato
    - Il supermercato è una struttura divisa in zone, composta da code e casse
    - I clienti entrano nel supermercato per fare la spesa minimizzando il tempo impiegato
  - **Agenti:** clienti e casse
- I clienti sono agenti intelligenti (pianificano e decidono) con una componente imprevedibile che fa emergere un comportamento complesso interagendo con gli altri agenti



# Ambiente

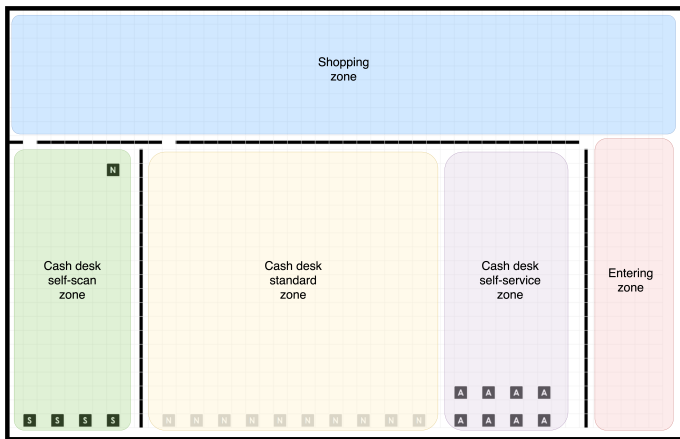


Figure: Stuttura del supermercato diviso in zone.

# Ambiente

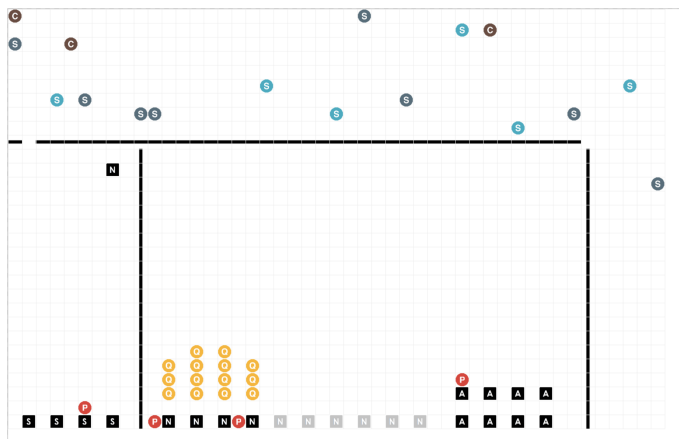


Figure: Interazione tra agenti e ambiente.

# Descrizione del modello

Cose teoriche su ambiente, agenti, interazione #TODO: Marco

# Agente Cliente

- I clienti sono gli agenti principali, si muovono nel supermercato con hanno l'obiettivo di fare la spesa e attendere il minimo tempo possibile in coda.
- Per minimizzare il tempo in coda il cliente usa strategie di **scelta della coda** e di **jockeying**, quindi ha bisogno di una pianificazione
- Il cliente è un agente di tipo *utility-based* perchè per la pianificazione e la valutazione dei tempi d'attesa utilizza una utility function.

## Agente Cliente - workflow

- 1 Attesa all'entrata del supermercato: l'agente è in attesa fino a che non può mettersi nella zona d'entrata.
- 2 Fase di shopping: si muove nella zona di shopping e inizia a raccogliere elementi fino a raggiungere il *basket-size* desiderato; questa fase dipende dalla velocità di shopping, un parametro del modello.
- 3 Scelta della coda: finita la spesa, deve scegliere la cassa in base alla utility function; viene scelta la coda  $q^*$  tale che

$$q^* = \operatorname{argmin}_{q \in Q} f(q)$$

dove  $Q$  è l'insieme delle code dedicate e  $f$  varia con la strategia.

## Agente Cliente - workflow

- 1 Attesa in coda e jockeying: mentre il cliente è in coda può decidere di lasciarla per una coda migliore. Considera le 2 code adiacenti (parametro del modello) alla propria e per ognuna calcola la coda migliore secondo la sua strategia di jockeying. Se il "guadagno" risulta maggiore di un certo **threshold**, allora il cliente può decidere di cambiare coda. Si estrae quindi un numero casuale che determina se cambiare coda o no (perchè non tutte le persone fanno jockeying).
- 2 Attesa alla cassa: il cliente viene servito dalla cassa e deve attendere la fine del pagamento per uscire dal supermercato.

# Agente Cassa

- I clienti, una volta conclusa la fase si di spesa, scelgono una coda in attesa di essere serviti in cassa per il pagamento.
- Ogni cassa ha al più una coda associata.
- La cassa è un'agente di tipo *model-based reflex*, in cui lo stato è il cliente che si sta processando in un determinato momento.
- Il comportamento di una cassa è piuttosto semplice:
  - 1 Prendi un cliente dalla coda (se disponibile)
  - 2 Processa il cliente
  - 3 Ripeti
- Le code (FIFO) ammissibili per ogni cassa sono di 2 tipi:
  - 1 Coda dedicata: ogni cassa ha una coda dedicata
  - 2 Coda condivisa: una coda è associata a più casse, tutte le casse serviranno i clienti che si sono accodati alla coda condivisa
- Nel modello sono state modellate 4 tipi di casse diverse.

# Agente Cassa - Tipo 1: Standard

- Rappresenta la classica cassa di un supermercato.
- Questa cassa può avere una coda dedicata o condivisa, in entrambi i casi:
  - 1 Il cliente si accoda
  - 2 La cassa prende il primo cliente dalla coda
  - 3 Il cassiere processa gradualmente tutti gli articoli del cliente
  - 4 Ripeti



## Agente Cassa - Tipo 2: Self-service

- Cassa in cui non è presente un cassiere ma è il cliente stesso a dover passare uno alla volta gli articoli acquistati.
- Tutte le casse self-service hanno una coda condivisa
  - ① Il cliente si accoda alla coda condivisa
  - ② La cassa prende il primo cliente dalla coda
  - ③ Il cliente processa ogni articolo
  - ④ Il cliente lascia il supermercato

## Agente Cassa - Tipo 3: Self-scan

- Il cliente scannerizza gli articoli durante la spesa mediante un dispositivo fornito dal supermercato
- Avendo già scannerizzato gli articoli a priori non è necessario farlo in cassa
- Vengono effettuati controlli a campione per verificare il corretto comportamento dei clienti (tutti gli articoli devono essere stati effettivamente scannerizzati durante la spesa)
- Tutte le casse hanno una coda condivisa
  - ① Il cliente si accoda alla coda condivisa
  - ② La cassa prende il primo cliente dalla coda
  - ③ Nel caso in cui il cliente è stato estratto per una rilettura della spesa si reca ad una cassa riservata, altrimenti paga ed esce dal supermercato

## Agente Cassa - Tipo 4: Riservata

- Comportamento analogo alla cassa standard.
- Ha una coda dedicata
- Nel caso di rilettura alla cassa "self-scan", il cliente viene normalmente processato in questa nuova cassa
- La rilettura può essere **parziale** (vengono controllati solo 10 elementi), o **totale** (viene controllata tutta la spesa)
- Nessun cliente può venire processato nella cassa riservata a meno di una rilettura

# Considerazioni

Il nostro modello è stra flessibile per le strategie e per gli stati e anche per i parametri che mo descriviamo #TODO: Marco

# Parametri

- Configurazione del supermercato:
  - Dimensione delle zone (entering zone, shopping zone)
  - Numero di casse (standard, self-service o self-scan + 1 riservata obbligatoria)
  - Code N-fork o parallele per le casse standard
- Jockeying:
  - Numero di code adiacenti considerate
  - Threshold
  - Probabilità di fare jockeying
- Distribuzione dei clienti in entrata (presa dai dati di Antczak e altri<sup>4</sup>)  
# TODO: qui manca la spiegazione come nel caso del basket-size, che però manca anche nella relazione
- Distribuzione dei basket-size (presa dai dati di Antczak e altri<sup>4</sup>)

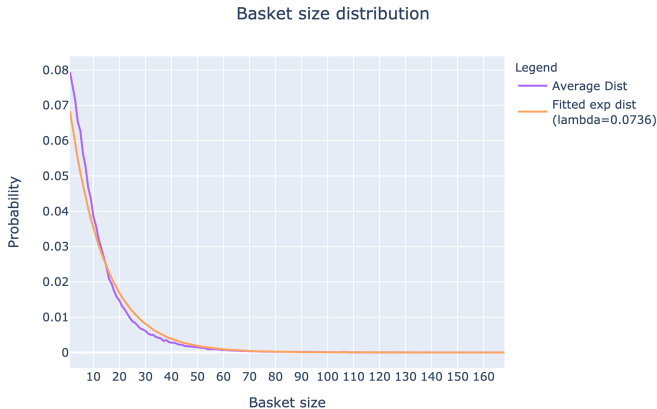
---

<sup>4</sup>*Data-driven simulation modeling of the checkout process in supermarkets: Insights for decision support in retail operations*, Antczak, Tomasz and Weron, Rafał and Zabawa, Jacek, 2020

# Parametri - distribuzione dei clienti

## Parametri - distribuzione dei basket size

Nella figura viene mostrata la distribuzione reale dei dati e la distribuzione esponenziale derivata aggregandoli, la quale viene usata nel modello per generare il basket size di ogni cliente.



# Parametri

- Parametri di tempo:
  - Durata di uno step (attualmente 30 secondi)
  - Velocità di shopping del cliente: numero di articoli messi nel carrello ad ogni step
  - Tempo di elaborazione della spesa da parte delle casse: per le self-scan è 1 step, per le standard e le self-service è governato dai parametri  $a, b, \alpha, \beta \in \mathbb{R}$ , presi dall'articolo di Antczak e altri sopra citato.



# Implementazione dei comportamenti e delle strategie

- Il cliente una volta presi tutti gli articoli si reca alla coda.
- La cassa self-scan deve essere scelta prima di fare la spesa e non è possibile cambiare.
- Il cliente vuole minimizzare il tempo speso all'interno del supermercato tramite:
  - **Scelta iniziale della coda:** il cliente sceglie la coda ottima rispetto ad una determinata strategia
  - **Fase di jockeying:** una volta in coda il cliente può scegliere di cambiarla se ne esiste una migliore

## Scelta della coda

definizione di strategia (con formula argmin) - 4 strategie - formule delle strategie #TODO: Marco

# Jockeying

- Un cliente fa **jockeying** se calcola che nelle code adiacenti a quella in cui è in attesa c'è un tempo di attesa minore, e quindi si sposta
- Il parametro di adiacenza determina il numero di code adiacenti che il cliente prende in considerazione per il suo calcolo
- Il cliente calcola un *guadagno* di tempo nel cambiare coda, se questo supera un certo threshold, allora fa jockey, altrimenti no perchè per lui "non ne vale la pena"
- Anche se esistono code migliori di altre, può non avvenire il jockey: viene estratto un parametro che rende il jockey aleatorio, in quanto non tutte le persone lo fanno

# Jockeying - strategie

- Sono 2 le strategie per fare jockeying:

- 1 **Minimo numero di elementi:** è scelta la coda con il minor numero di elementi nei carrelli di tutti i clienti. Il guadagno è:

$$g = \# \text{elementi nei carrelli nella coda pivot} - \min_{q \in Q_{adj}} \# \text{elementi nei carrelli}$$

- 2 **Minimo numero di persone:** è scelta la coda con il minor numero di persone accodate. Il guadagno è:

$$g = i - \min_{q \in Q_{adj}} |q|$$

dove  $i$  è la posizione del cliente nella coda pivot

# Simulazione

Introduciamo le definizioni di densità e flusso di clienti ad ogni step al fine di avere una misura della gestione dell'affluenza di clienti nel negozio per le simulazioni.

La **densità** di clienti per step corrisponde al numero di clienti medio per ogni cassa; la densità allo step  $i$  è:

$$\text{density}_i = \frac{\# \text{ customers in the supermarket}}{\# \text{ cashdesks}}$$

Il **flusso in entrata** di clienti per step corrisponde al numero di clienti che entrano ad ogni step per ogni cassa in media; il flusso allo step  $i$  è:

$$\text{flow}_i = \frac{\# \text{ customers entering in the supermarket}}{\# \text{ cashdesks}}$$

# Validazione

Stessa simulazione dei polacchi con le 4 strategie

## Simulazione con jockey

Uguale a sopra aggiungendo il jockey (4 strategie + 2 strategie di jockey)

## Simulazione con coda condivisa

Uguale a sopra ma con coda condivisa (quindi senza scelta della coda e senza jockey)



## Simulazione con casse self-scan

Non importa che sia uguale a sopra perchè tanto è come se fosse una simulazione a parte, si può fare anche con 0 standard e 0 self-service, giusto per non avere mille robe in mezzo

# Simulazione non deterministica

Uguale con tutto (quindi a sx abbiamo i self scan e a dx le casse normali con coda condivisa o meno, 4 strategie di scelta della coda, 2 strategie di jockey) però accendiamo i parametri probabilistici

# Conclusioni

DAJE RAGA