Machine Learning and Optimization in Tourism and Hospitality

Roberto Battiti and Mauro Brunato

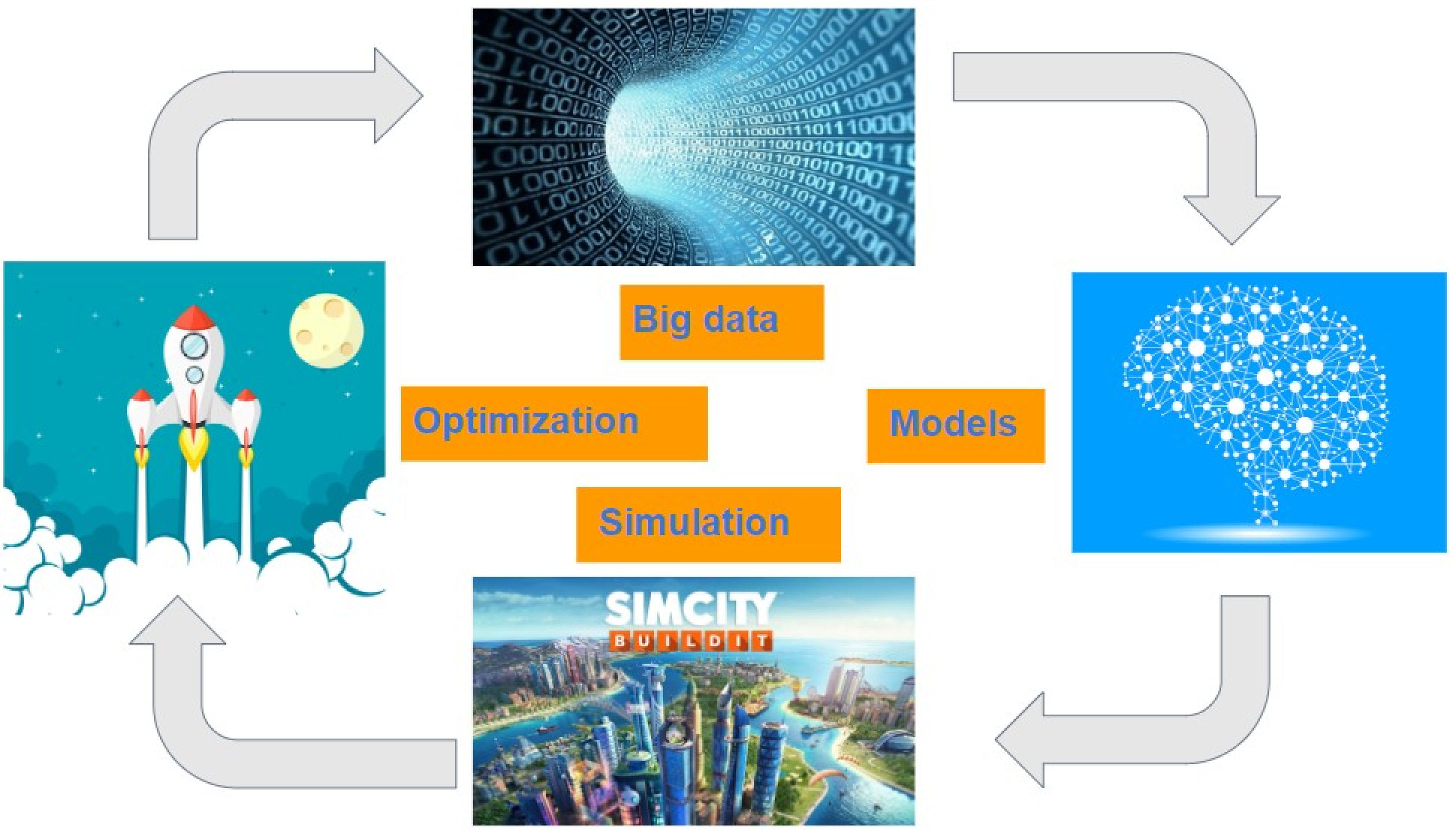
LION-lab

University of Trento

LION

intelligent-optimization.org

UNIVERSITY OF TRENTO - Italy



Part 1
Machine learning:
From Data to Models

Data mining

- Internal
  - Databases
  - Web logs
  - Application logs
- External
  - Social network APIs
  - Data brokers
  - Web scraping

Data mining: databases

Highly structured

Relatively fast, can respond to complex queries by means of a fairly structured language (SQL)

Designed to store information related to normal operativity: property/accommodation info, prices, availability, reservations, customers...

Data mining: databases

“For each property and accommodation type, how many reservations involving August dates have been confirmed?”

```
SELECT
    `a`.`property_id`, `r`.`accommodation_id`, COUNT(`r`.`id`)
  FROM
    `reservation` AS `r`
    JOIN `accommodation` AS `a`
        ON `a`.`id`  = `r`.`accommodation_id`
  WHERE
    `r`.`checkin` < "2019-09-01"
    AND `r`.`checkout` >= "2019-08-01"
    AND `r`.`confirmed` = 1
  GROUP BY
    `r`.`accommodation_id`
  ORDER BY
    `a`.`property_id`,
    `r`.`accommodation_id`;
```

```
+-------------+------------------+-----------------+
| property_id | accommodation_id | COUNT(`r`.`id`) |
+-------------+------------------+-----------------+
|       14185 |            14282 |               2 |
|       14185 |            14299 |               6 |
|       14185 |            14493 |               7 |
|       14497 |            14507 |               2 |
|       14497 |            14516 |               4 |
|       14497 |            14529 |              21 |
|       14497 |            14542 |              15 |
|       14497 |            14555 |              12 |
+-------------+------------------+-----------------+
```

Data mining: web logs

Quite structured

Record interaction events between customers and reservation portal

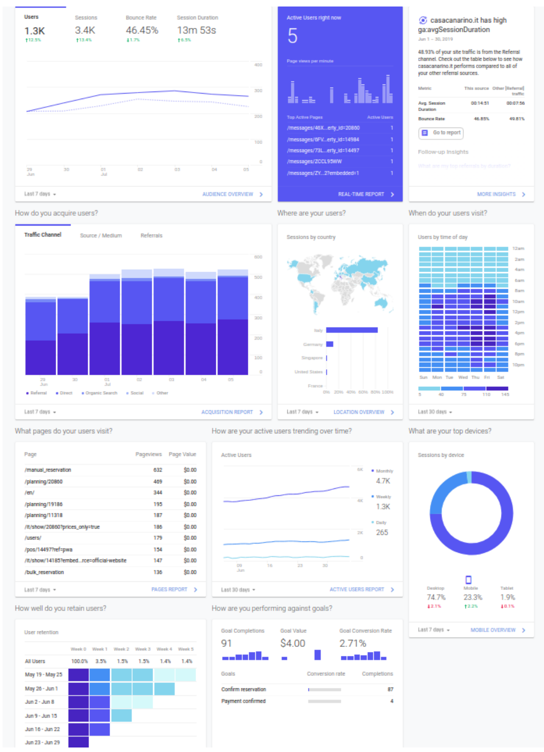
Not all details available (e.g., request variables, detailed responses)

However, they can fully track a specific user’s interaction with the website by means of the “referer” field

Data mining: web logs

Initial contact with the reservation portal by user referred by the official hotel site in English

```
11.22.33.44 - -
  [20/Jun/2019:07:11:09 +0000]
  "GET /en/show/12345?
  embedded=1 HTTP/1.1"
  200 41295
  "https://official_hotel_site
  .com/en/book-now"
  "Mozilla/5.0 (Macintosh;
  Intel Mac OS X 10_14_5)
  AppleWebKit/537.36 (KHTML,
  like Gecko)
  Chrome/74.0.3729.169
  Safari/537.36"
```



Data mining: application logs

- (Almost) unstructured text files, may contain every detail the application designer (or, more often, the actual coder) deems to be relevant for debugging/tracing.
- Possibly hard to parse, structure changes over time depending on new developers being added to the project, new functions…
- Often provide precious information about customer interaction.

Data mining: application logs

Example: free-form debug information about a potential guest asking for quotes in specific dates

```
2019-07-03T05:08:54.758Z INFO Computing price for property 20860-a2aa11f0-9d50-11e9-
  98d2-3ba9bed9a43e, accommodation 1
FREE 16 0 17 0 0 0
AGE DELTAS [ 0, 0 ]
LAST 2019 6 16 2019-07-03T05:08:54.759Z 348.8514558333333 0 0
2019-07-03T05:08:54.759Z INFO Proposed price: 20860-a2aa11f0-9d50-11e9-98d2-
  3ba9bed9a43e 20870 2019-07-17T00:00:00.000Z 2019-07-18T00:00:00.000Z 1 0 1 true 17
  80
2019-07-03T05:08:54.759Z INFO Computing price for property 20860-a2aa11f0-9d50-11e9-
  98d2-3ba9bed9a43e, accommodation 2
FREE 16 0 17 0 0 0
AGE DELTAS [ 0, 0 ]
LAST 2019 6 16 2019-07-03T05:08:54.760Z 348.85145555555556 0 0
...
```

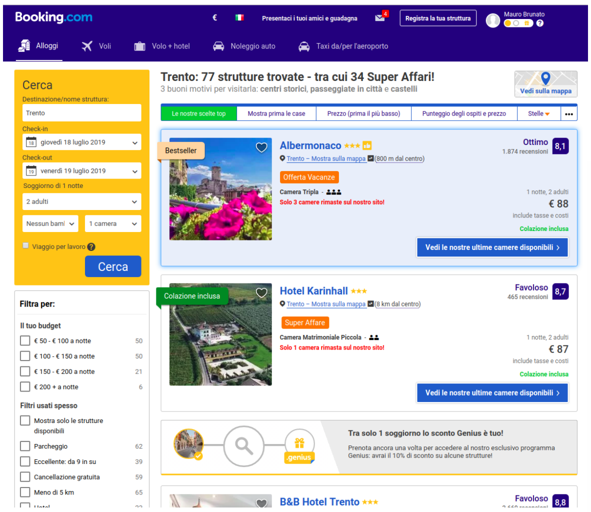
Data sources: external providers

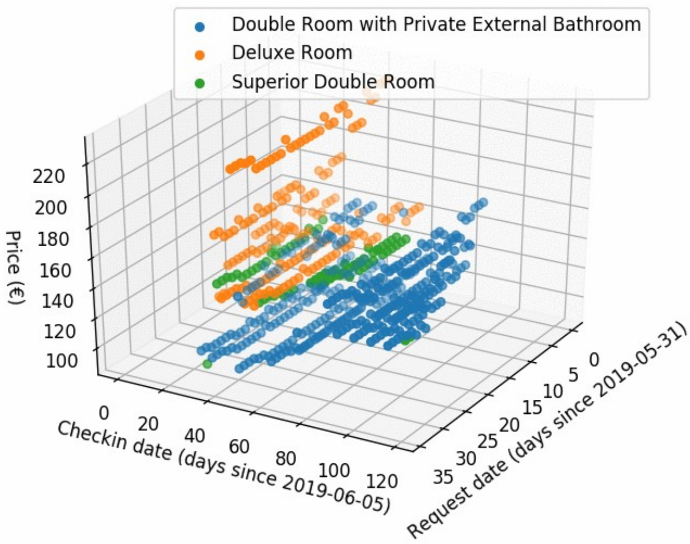
Facebook/Twitter/... APIs (with limitations)

Data brokers: collecting and selling information

Websites: scraping data without annoying the webmaster (and keeping below the radar of DoS detectors).

External sources: web scraping





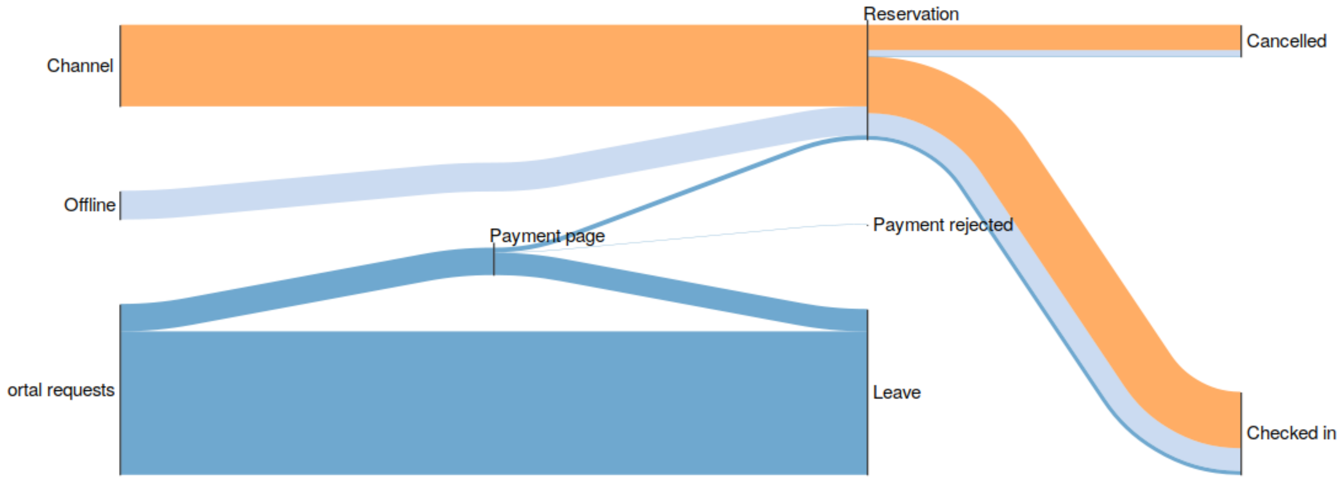
Example: tracking a guest's reservation

Weblog: locate early access, referrer and landing page.

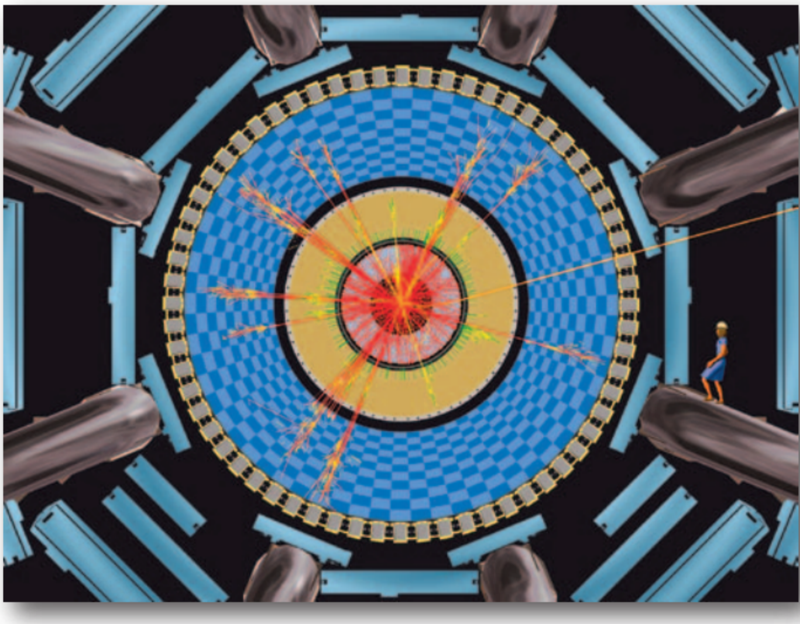
Application log: see history of queries (different dates/options), confirmation attempts, credit card acceptance/refusal...

Database: further reservation details, possible cancellation, checkin/out, purchases...

Example: user tracking



Detour: "big" data?



CERN ATLAS experiment

Trigger and Data Acquisition (TDAQ)

The trigger system selects 100 interesting events per second out of 1000 million total.
The data acquisition system channels the data from the detectors to storage.

Bunches of protons cross 40 million times a second.
Each bunch contains 10^{11} protons.
Number of proton-proton collisions in the detector: 1 billion per second.
When any of the protons collide, the process is called an "event".
A given bunch crossing sometimes has particles from more than one proton-proton collision.

If all data would be recorded, this would fill 100 000 CDs per second. This would create a stack of CDs 150 m (450 ft) high every second, which could reach to the moon and back twice each year.
This data rate is also equivalent to making 50 billion telephone calls at the same time.

TDAQ has a 3 level Trigger system (reduction in three steps).
Total event reduction factor by the trigger system: 200 000.

- 1st level trigger: Hardware, level 1 is done using special-purpose processors.[1]
- 2nd level trigger: Software, large computing farms with ~ 500 dual pc processors.
- 3rd level trigger: Software, large computing farms with ~ 1700 dual pc processors.

The rates and reduction factors at 14 TeV are summarized as:

	Incoming event rate per second	Outgoing event rate per second	Reduction factor
Level 1	40 000 000	100 000	400
Level 2	100 000	3 000	30
Level 3	3 000	200	15

TDAQ records 320 Mbytes per second, which would fill more than 27 CDs per minute.

Source: atlasexperiment.org – ATLAS 18-page fact sheet

Computing

Analysing 1000 Million events recorded per year

Data recording:

- The raw data are recorded after 3rd level Trigger (see previous page).
- The raw data are analysed in terms of particles produced in the collision (tracks, shower in the calorimeters etc.) and this "reconstructed" data is recorded.
- From this the physics data is extracted with specialized software and recorded.

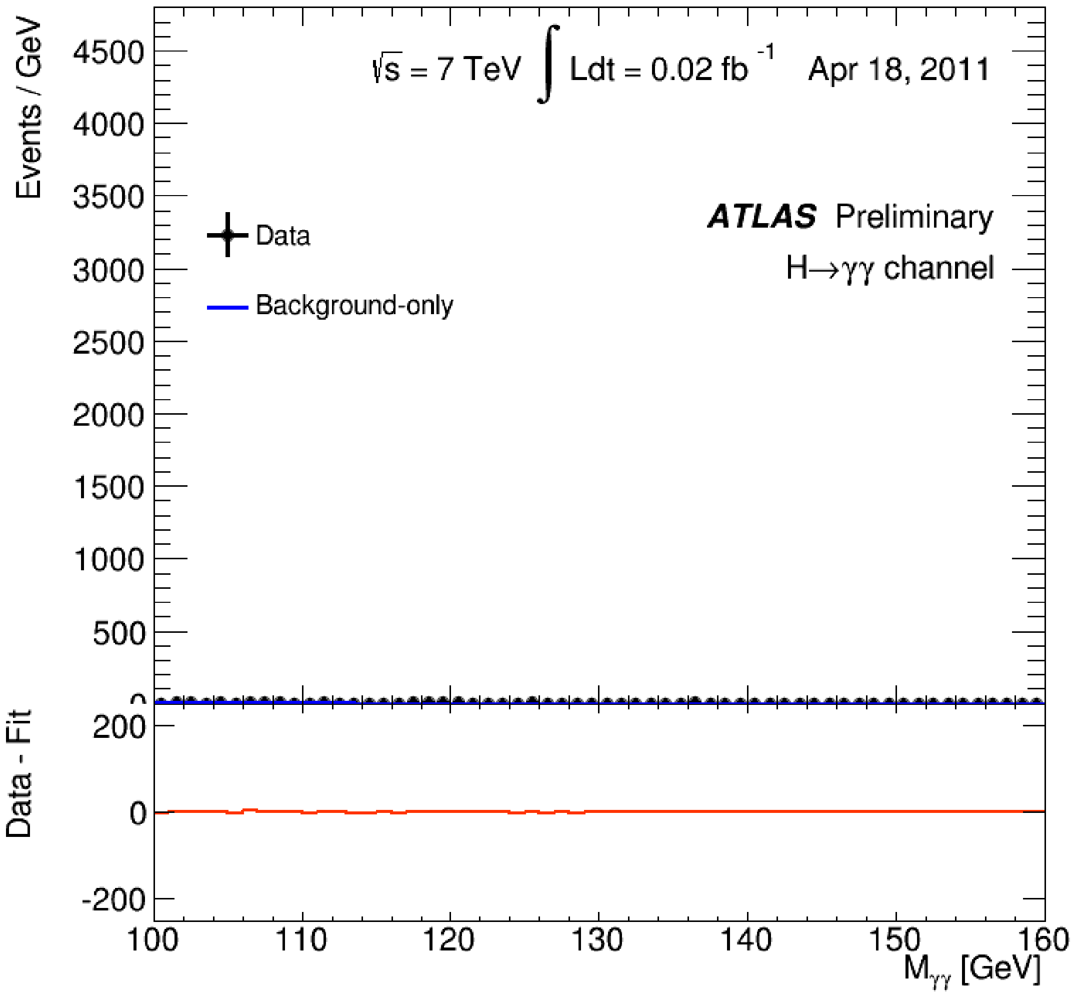
Recorded	per event	per year
raw data	1.6 Mbytes	3 200 Tbytes
reconstructed data	1 Mbytes	2 000 Tbytes
physics data	0.1 Mbytes	200 Tbytes

(A terabyte is a million megabytes)

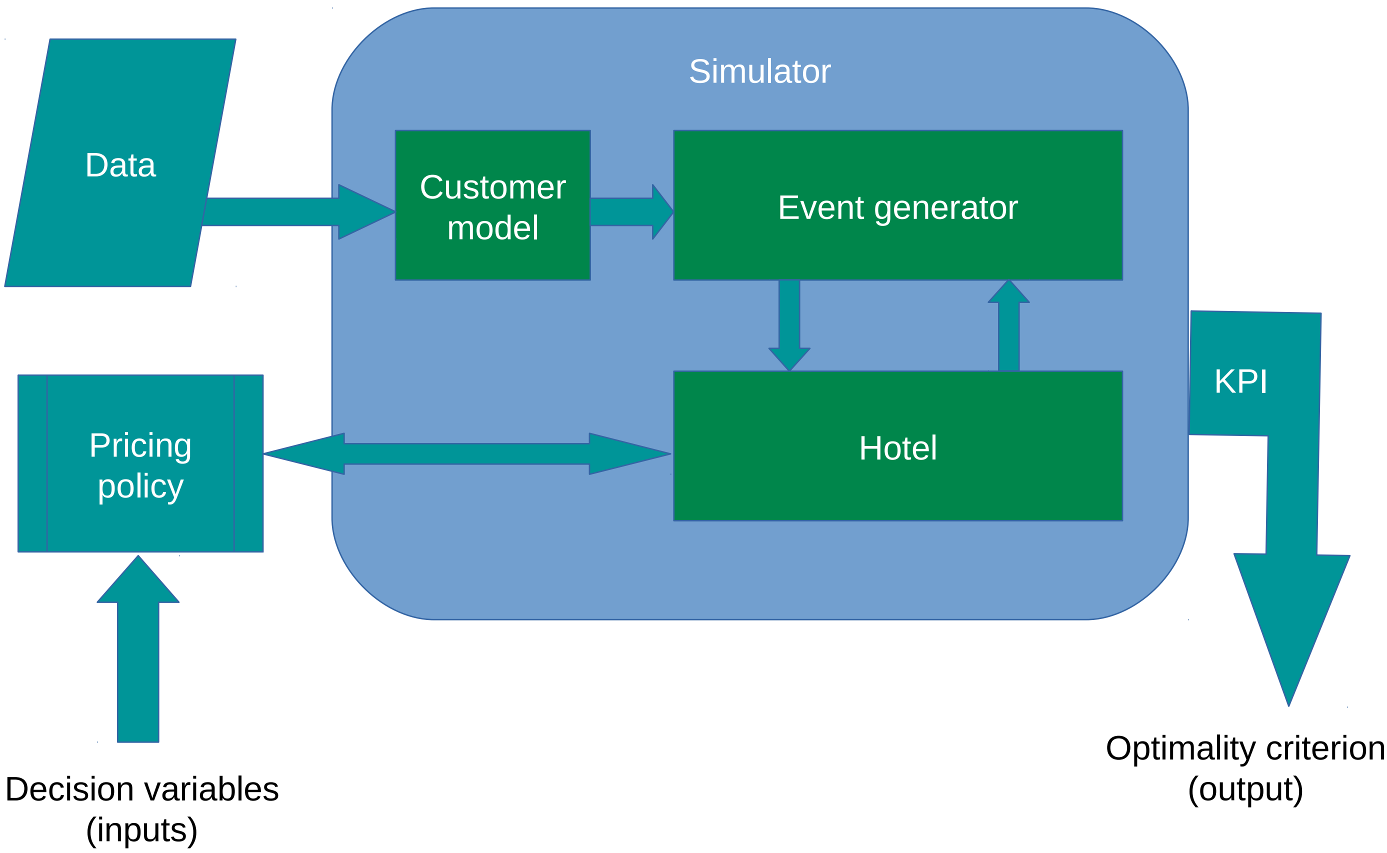
The 3200 terabytes of data that will be seen by ATLAS each year are the equivalent of the content in:

- 160 million trees made into books.
- 7 km (4 miles) of CD-ROMs stacked on top of each other.
- 600 years of listening to songs.
- 160 US Library of Congress (3 billion books).

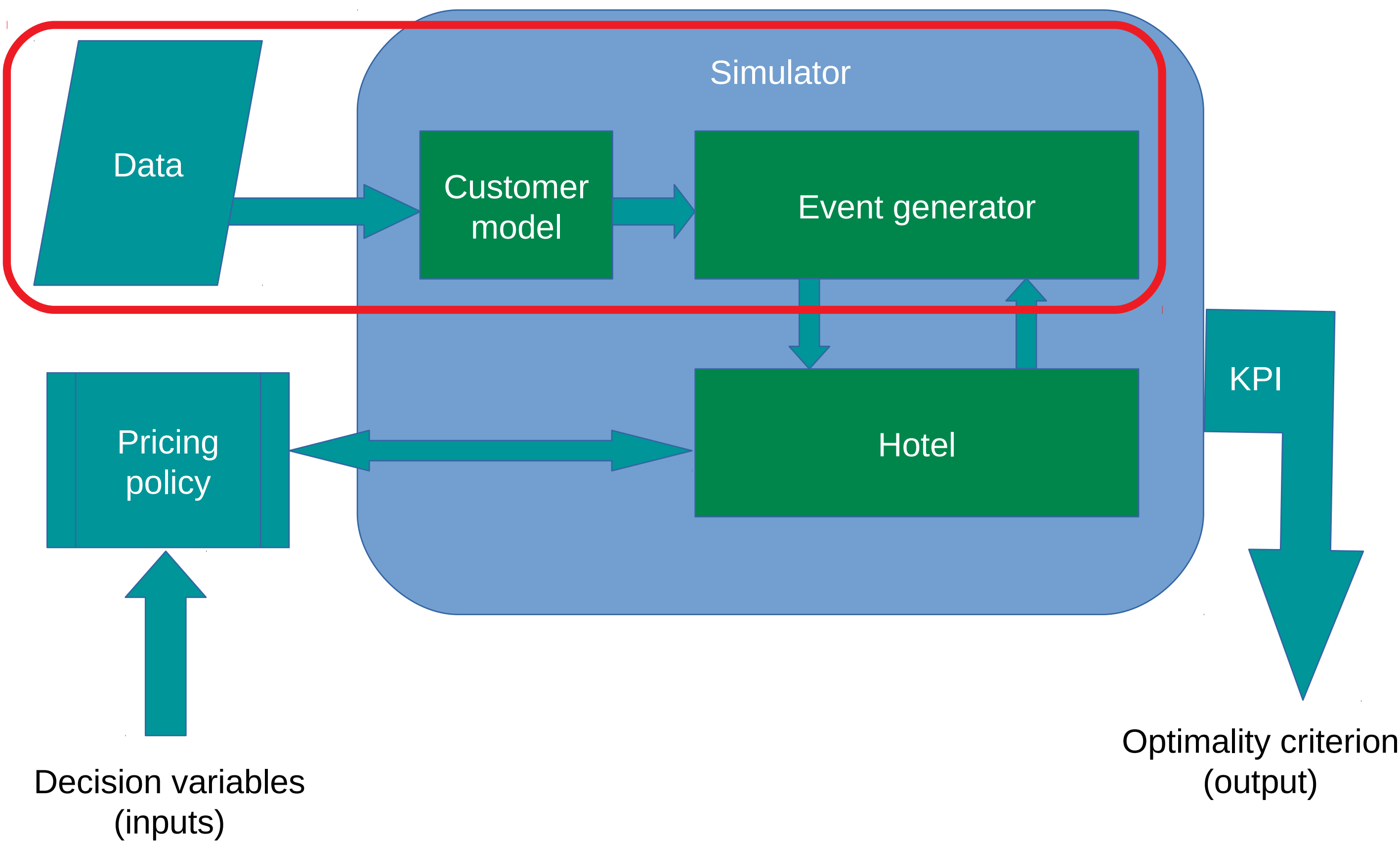
Source: atlasexperiment.org – ATLAS 18-page fact sheet



Generative models

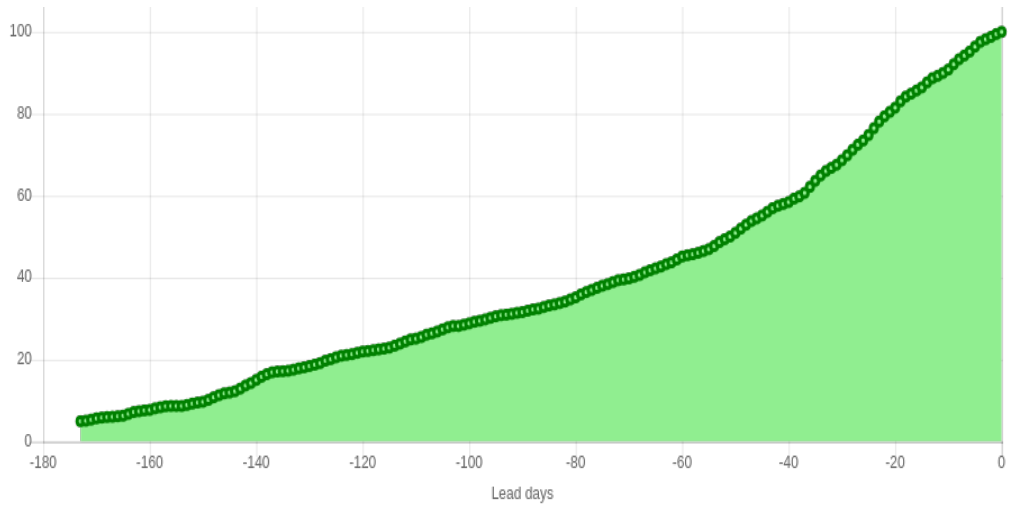


Modeling the customer



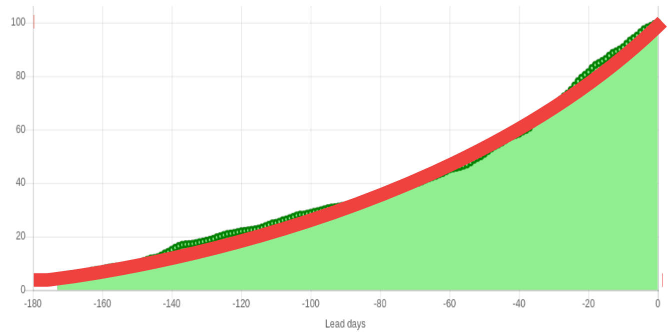
Modeling the customer

Cumulate number of reservations *d* days before check-in
(100 = #reservations @check-in)



Modeling the customer

Model: number *k of* events taking place *d* days before check-in:

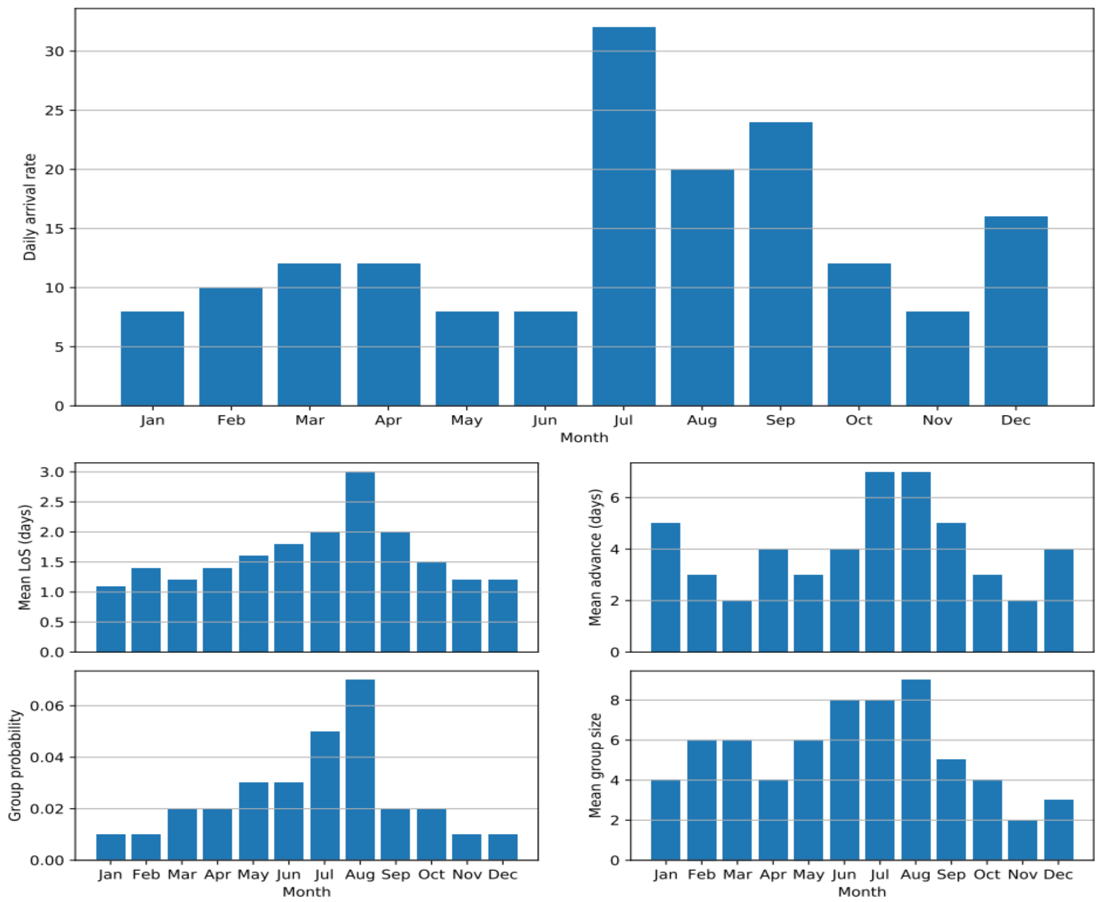


$$Pr(k) = \frac{e^{-\lambda_d} \lambda_d^k}{k!}$$

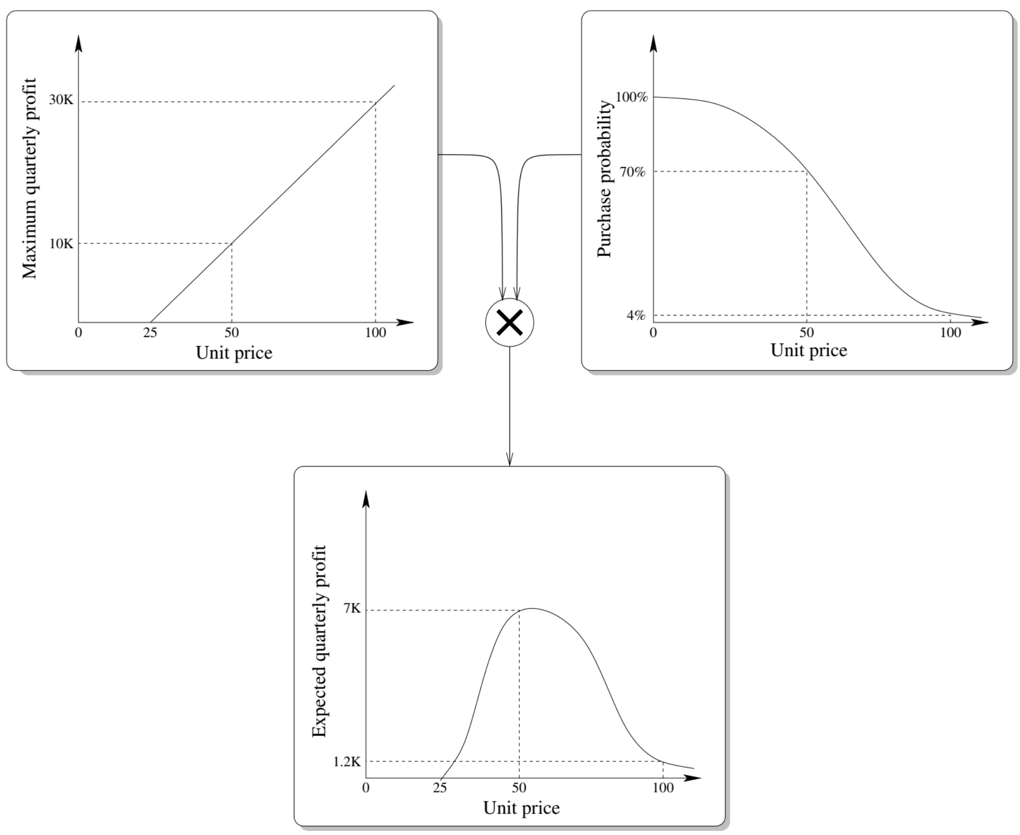
Standard non-homogeneous Poisson process: every day is characterized by a possibly different parameter λ_d.

$$\lambda_d = e^{\alpha_0 + \alpha_1 t + \alpha_2 I_w + \alpha_3 I_{day0} + \alpha_4 I_{week0}}$$

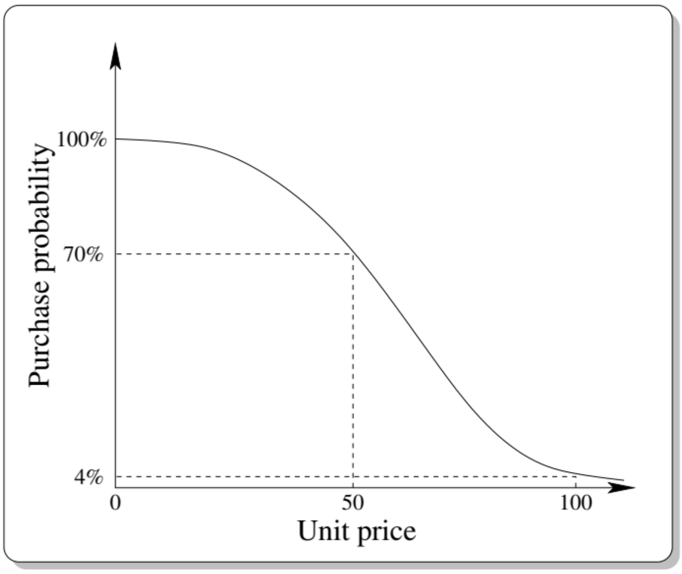
Parameters based on historic data

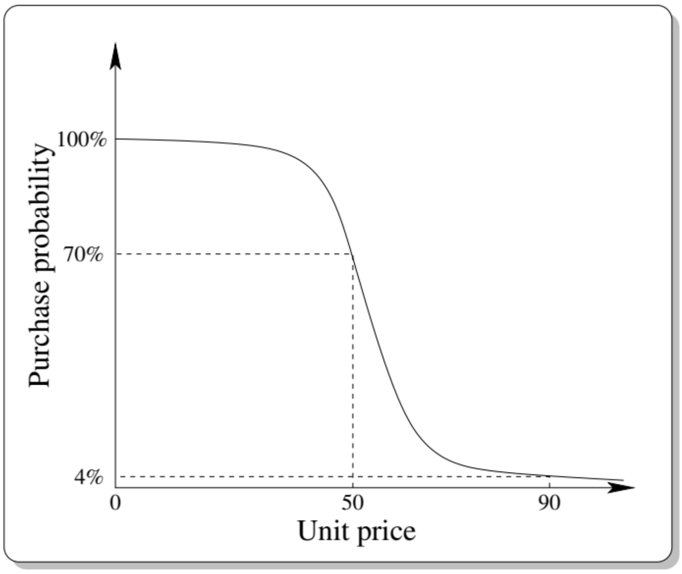


Modeling the customer

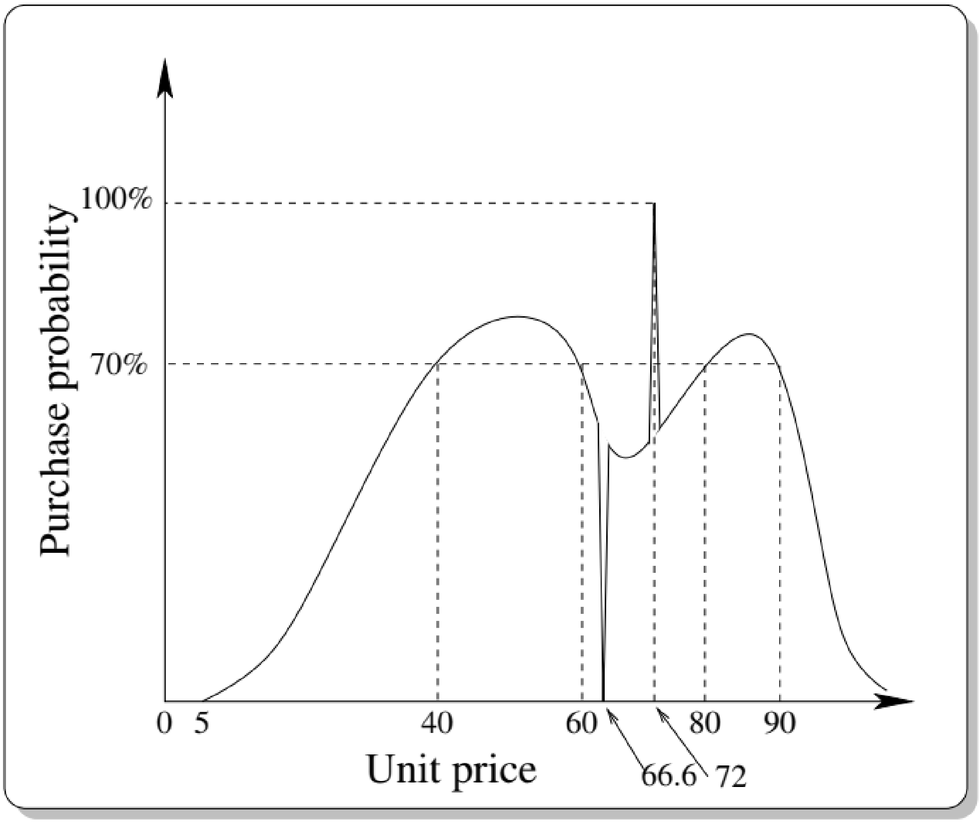


Modeling the customer





Modeling the customer



Modeling the customer

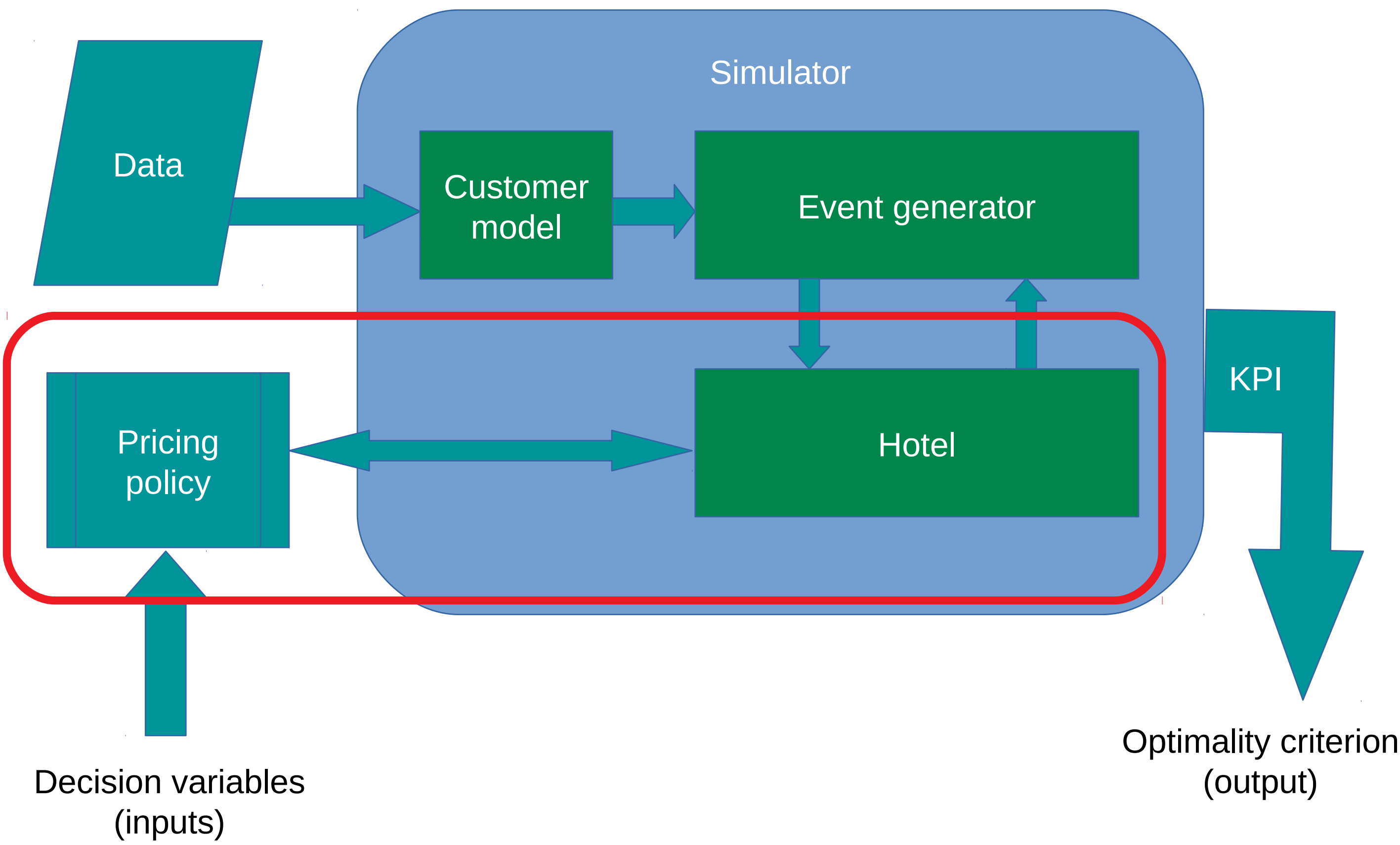
$$Pr(u) = 1 - \sigma\left(\frac{u-\mu}{\eta}\right)$$

where

$$\sigma(x) = \frac{1}{1-e^{-x}}$$

- μ is the median acceptance price
- η controls the slope (rate of decay) – the smaller, the sharper

Pricing policies

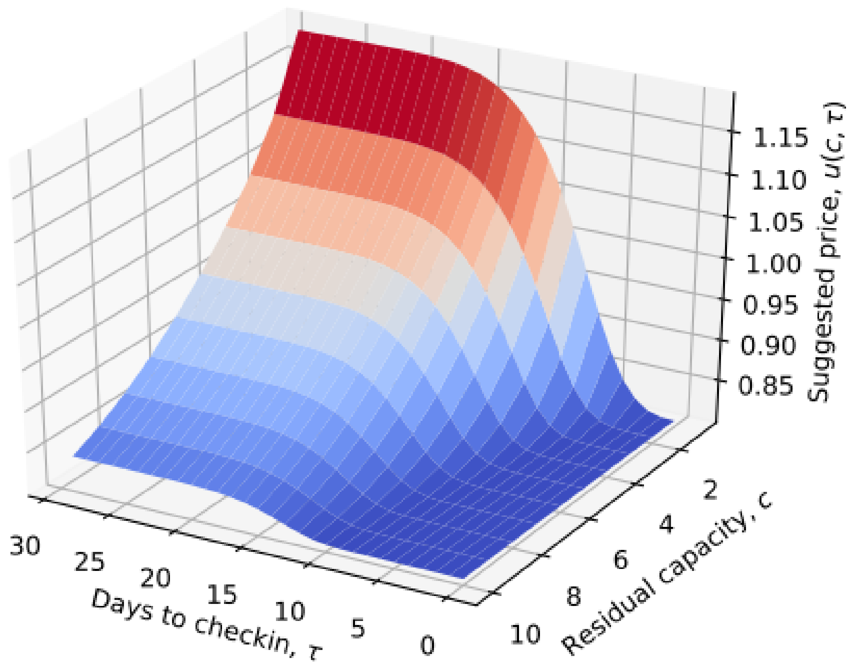


Pricing policy: dynamic programming

$$u = u(c, \tau)$$

$$V(c,\tau) = p(u,\tau)(u + V(c-1, \tau-1)) + (1 - p(u,\tau))V(c, \tau-1)$$

c Residual capacity
τ Days in advance of checkin

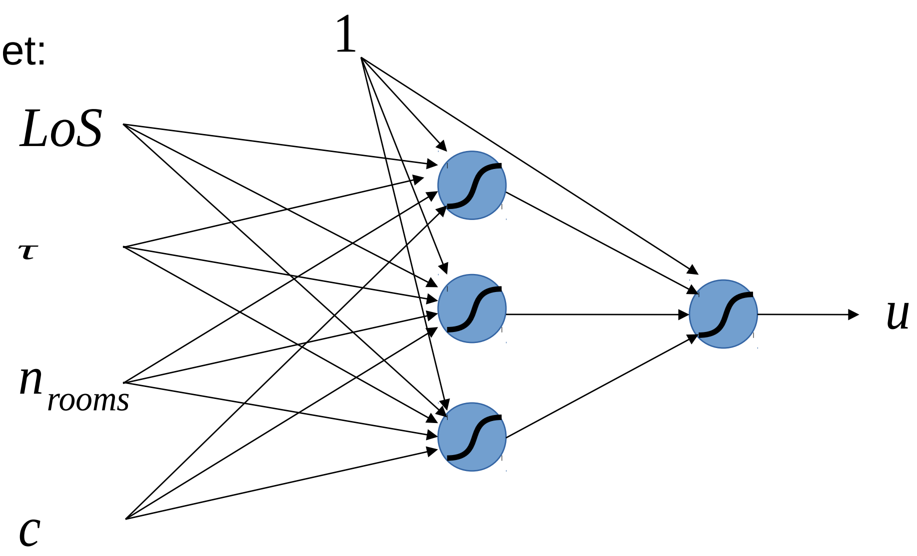


Pricing policy

Factored:

$$u = u_0 \times f_1(LoS) \times f_2(\tau) \times f_3(n_{rooms}) \times f_4(c)$$

Neural net:

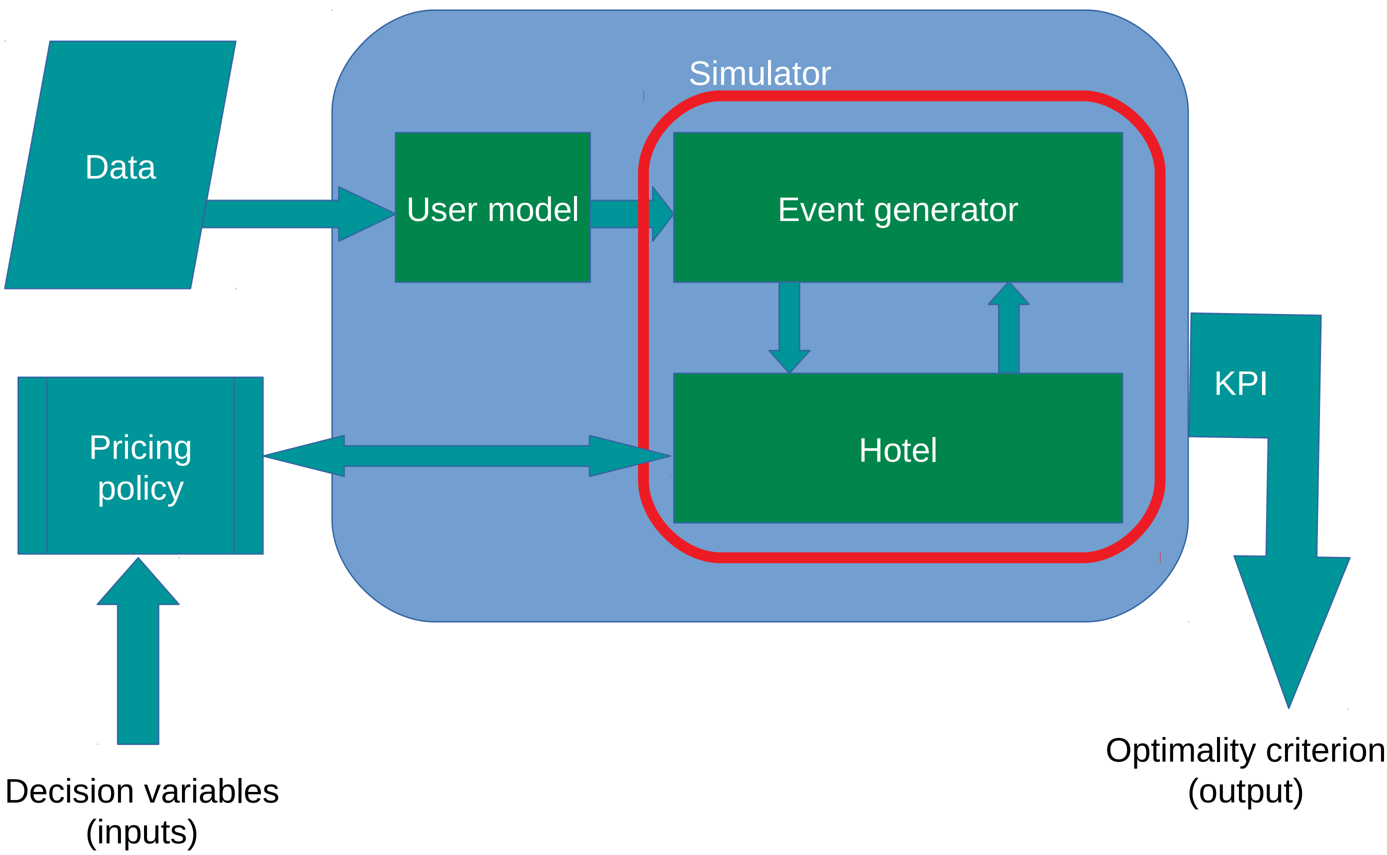


Pricing policy

Price may be a function of:

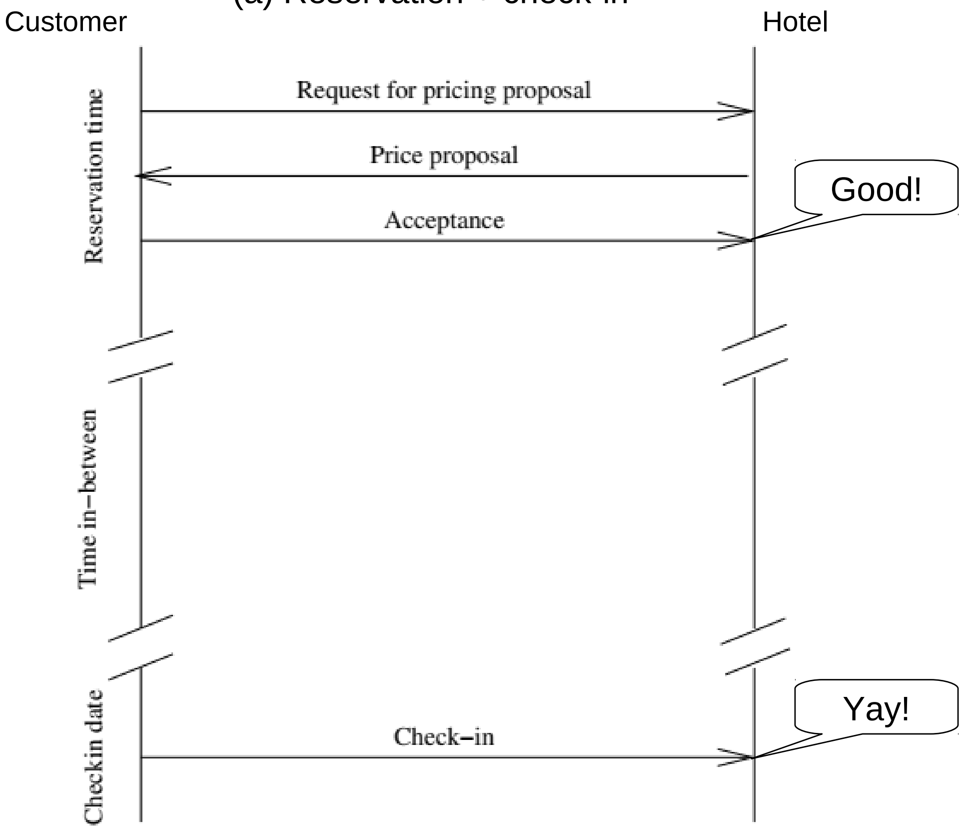
- Capacity
- Predicted (un)constrained demand
  - OTB demand
  - Similar day previous years
- Request features (LoS,…)
- Customer’s predicted reference price, elasticity...

The protocol



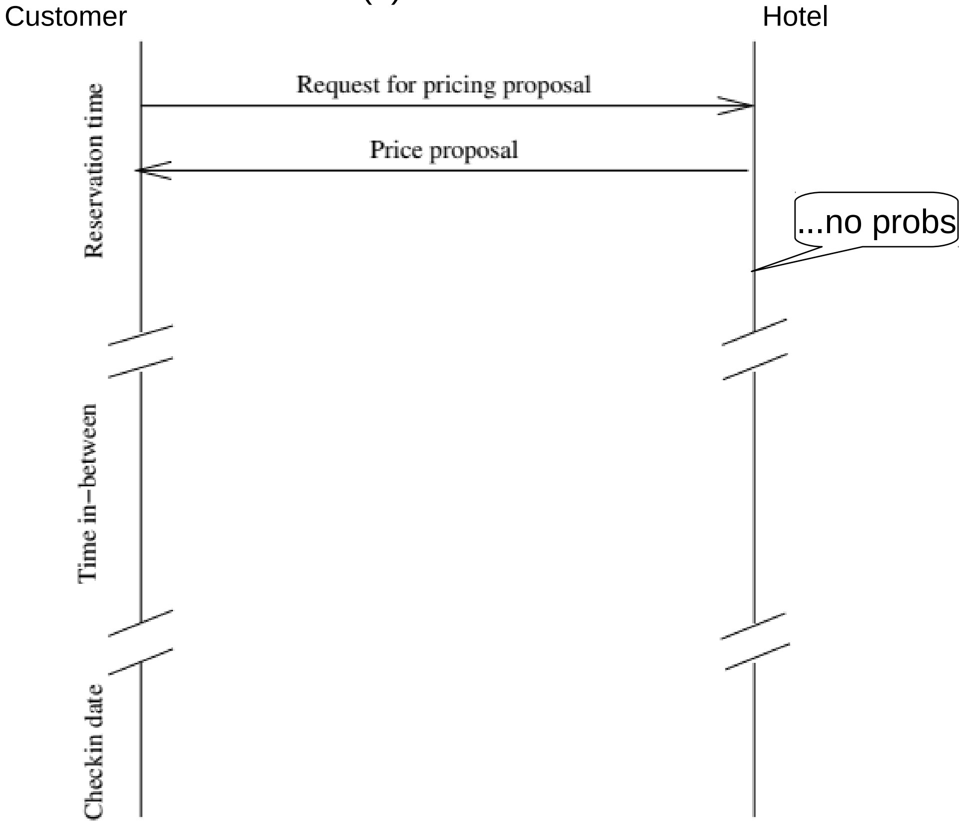
The protocol

(a) Reservation + check-in



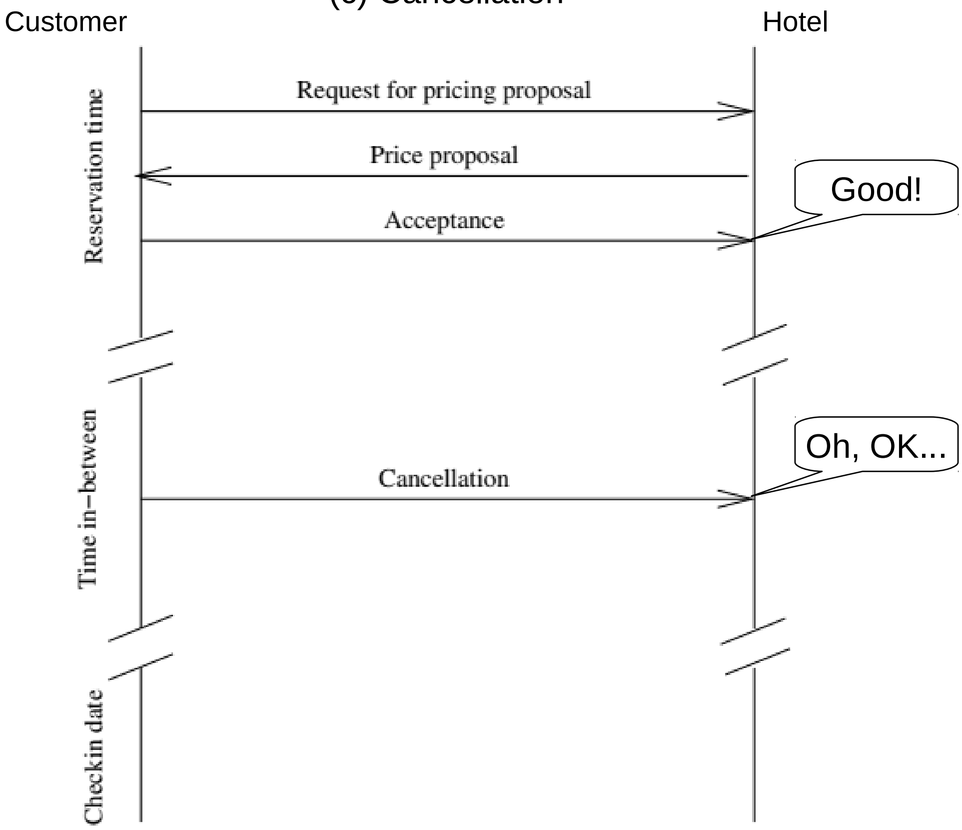
The protocol

(b) Refusal



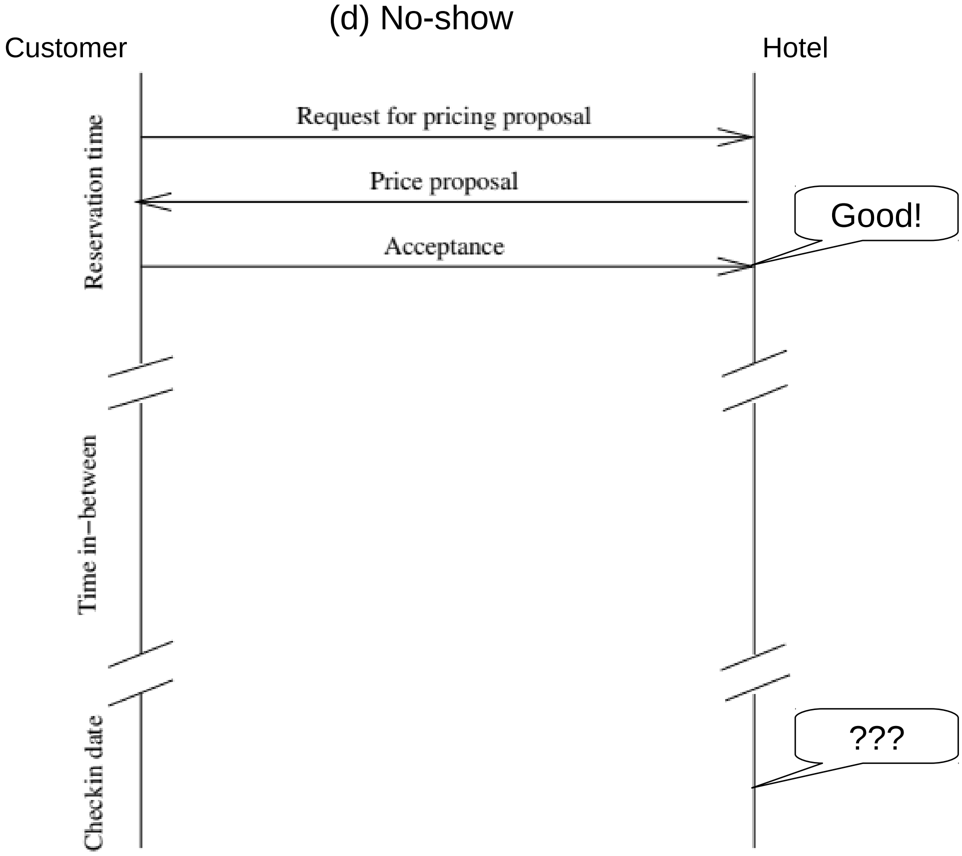
The protocol

(c) Cancellation

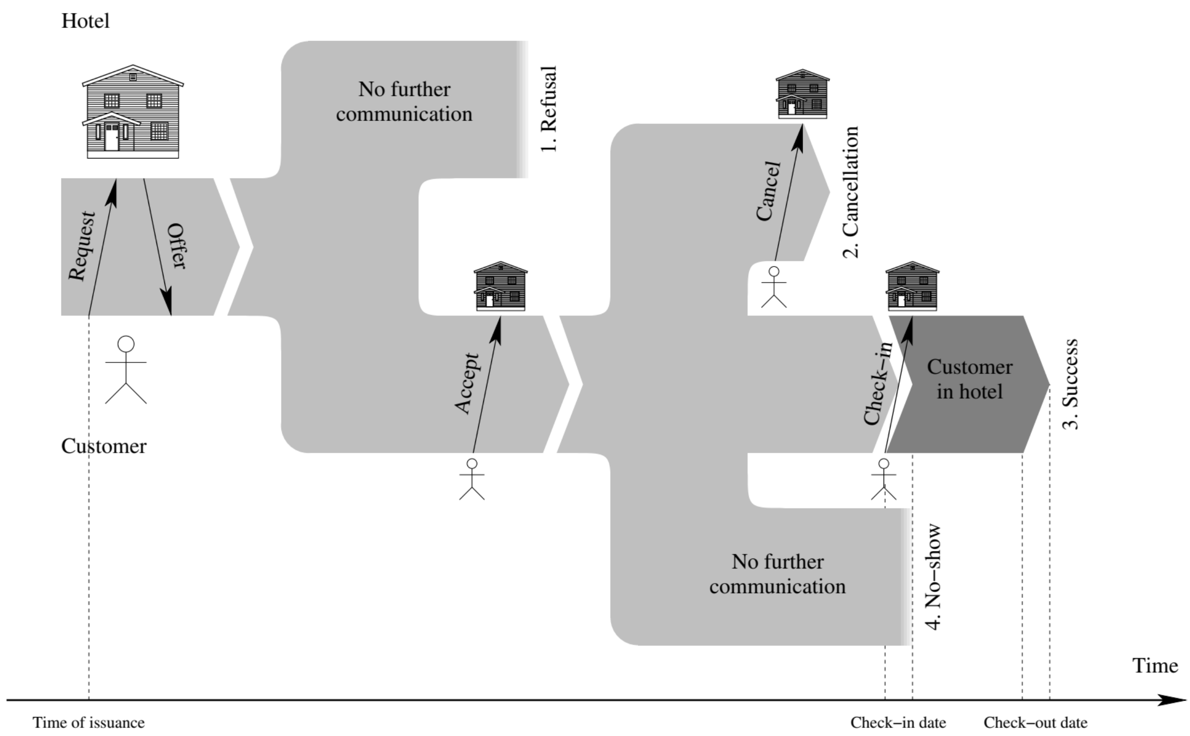


The protocol

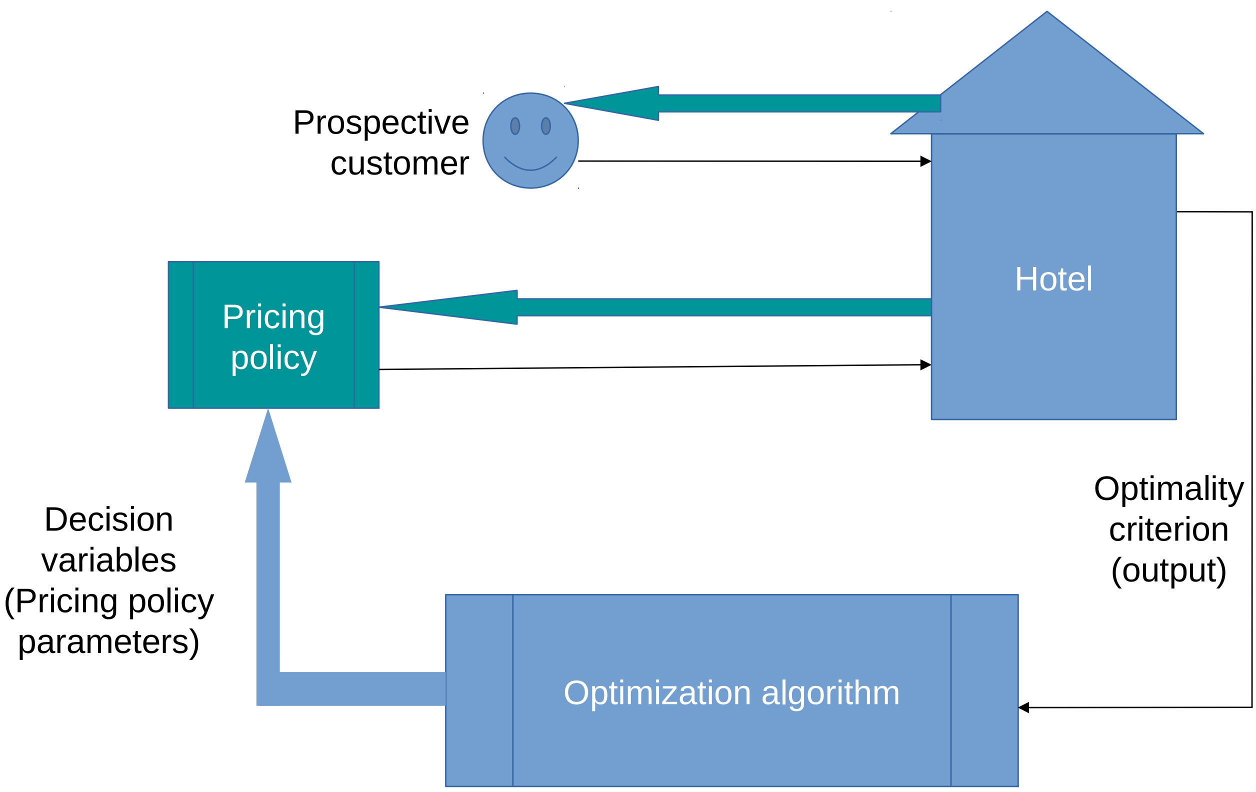
(d) No-show



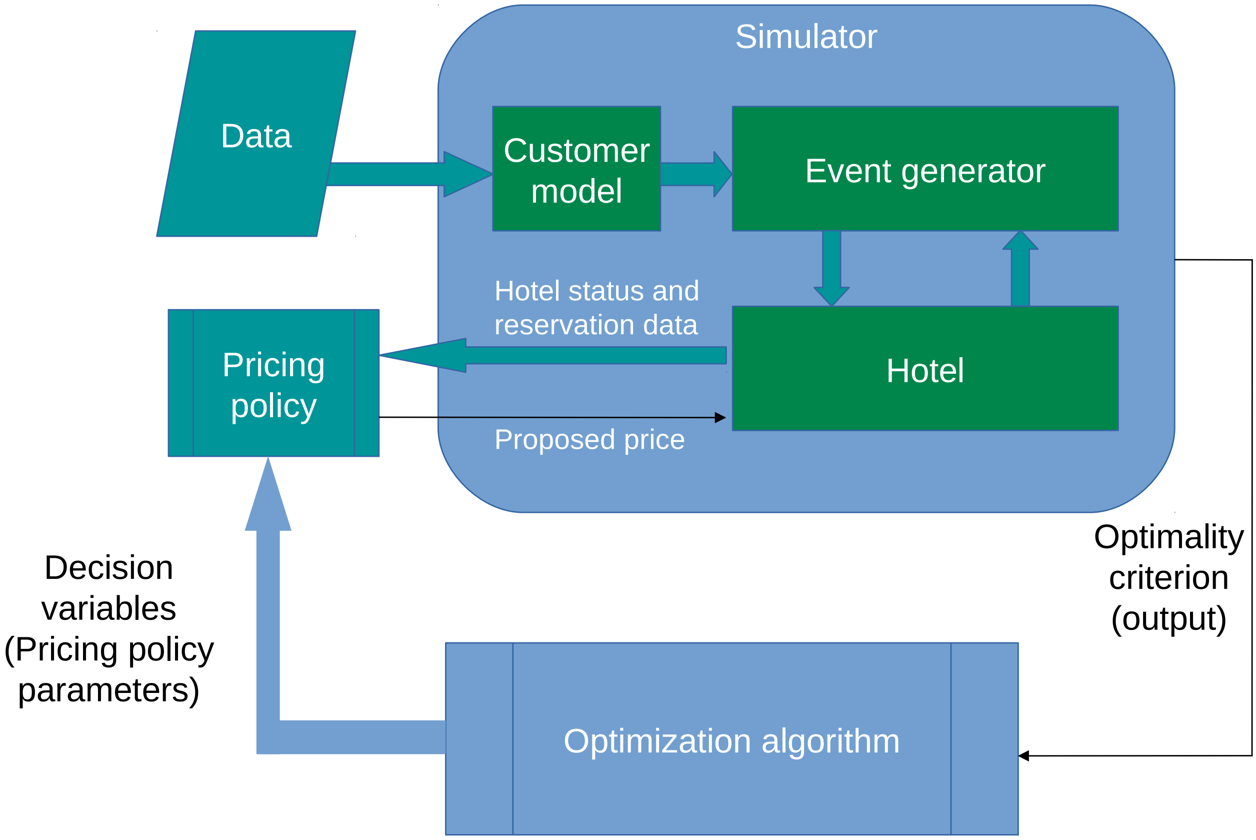
The protocol



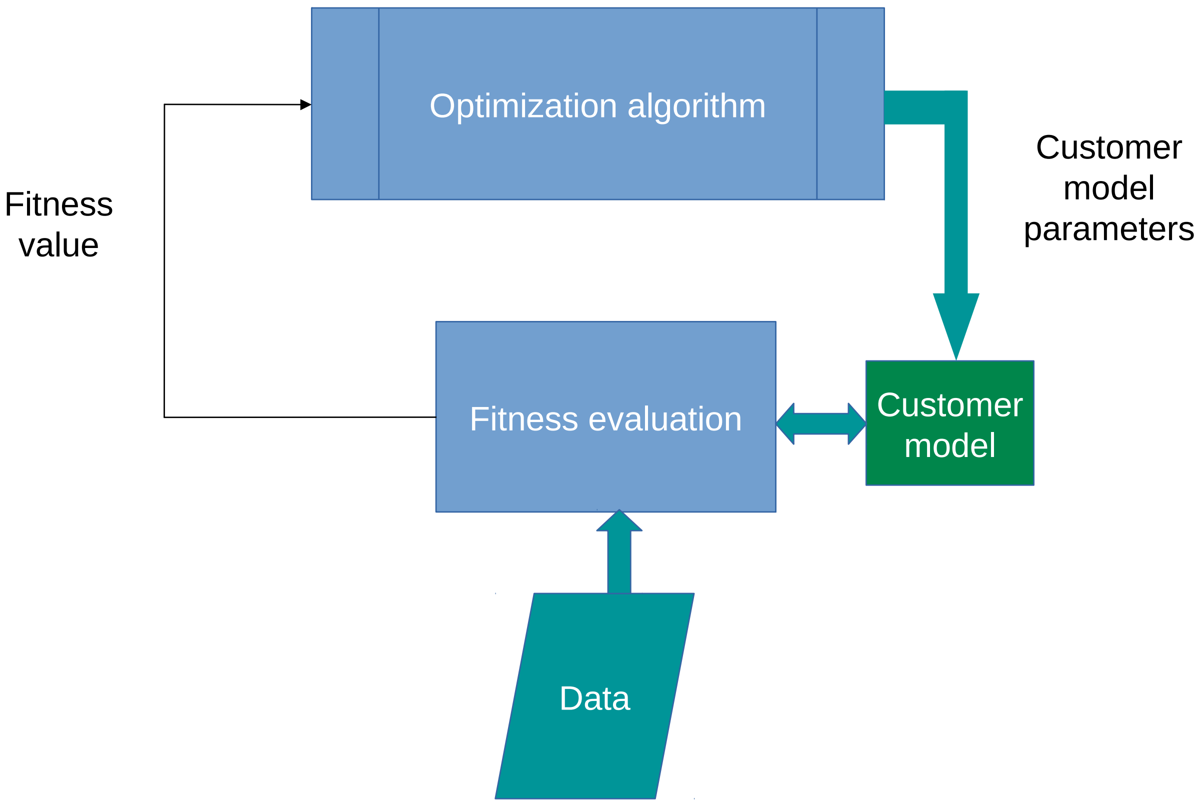
Optimization loop



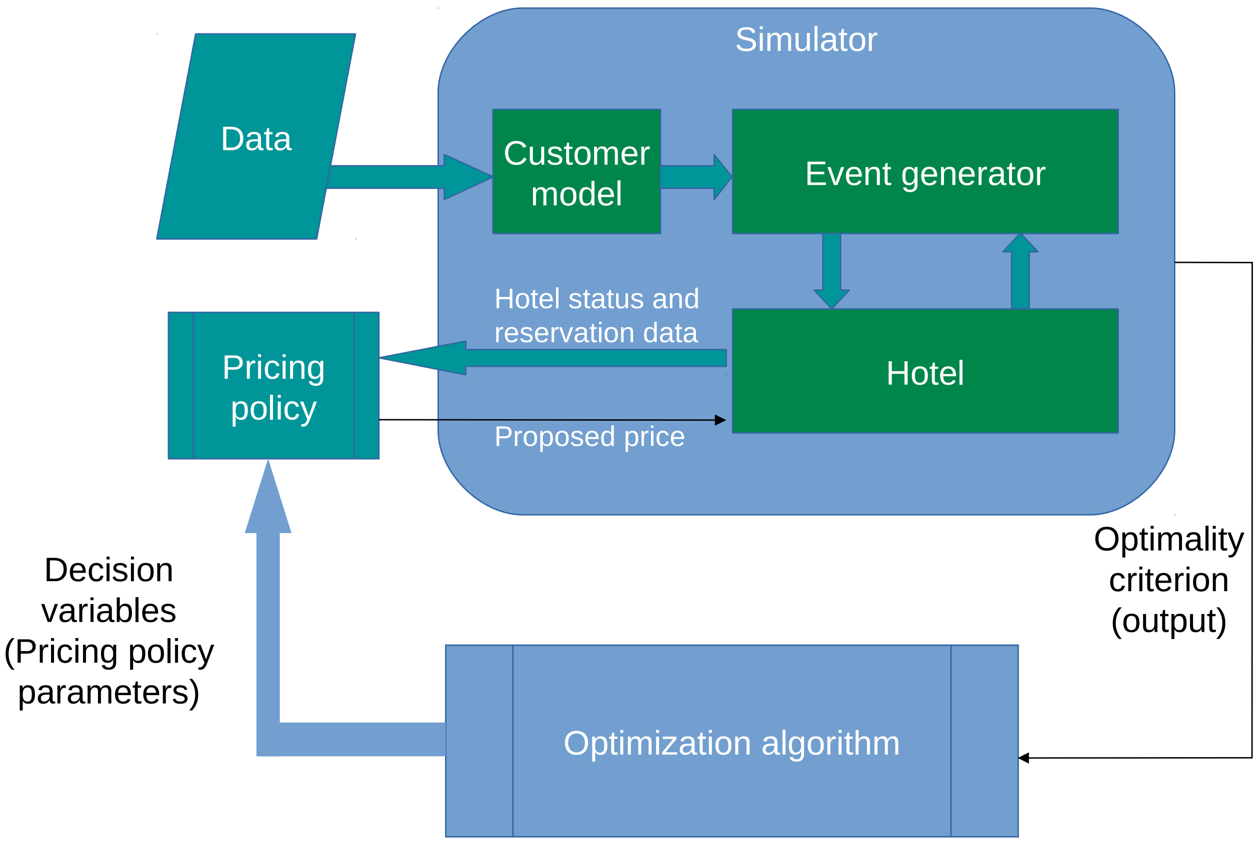
Simulator / optimizer loop



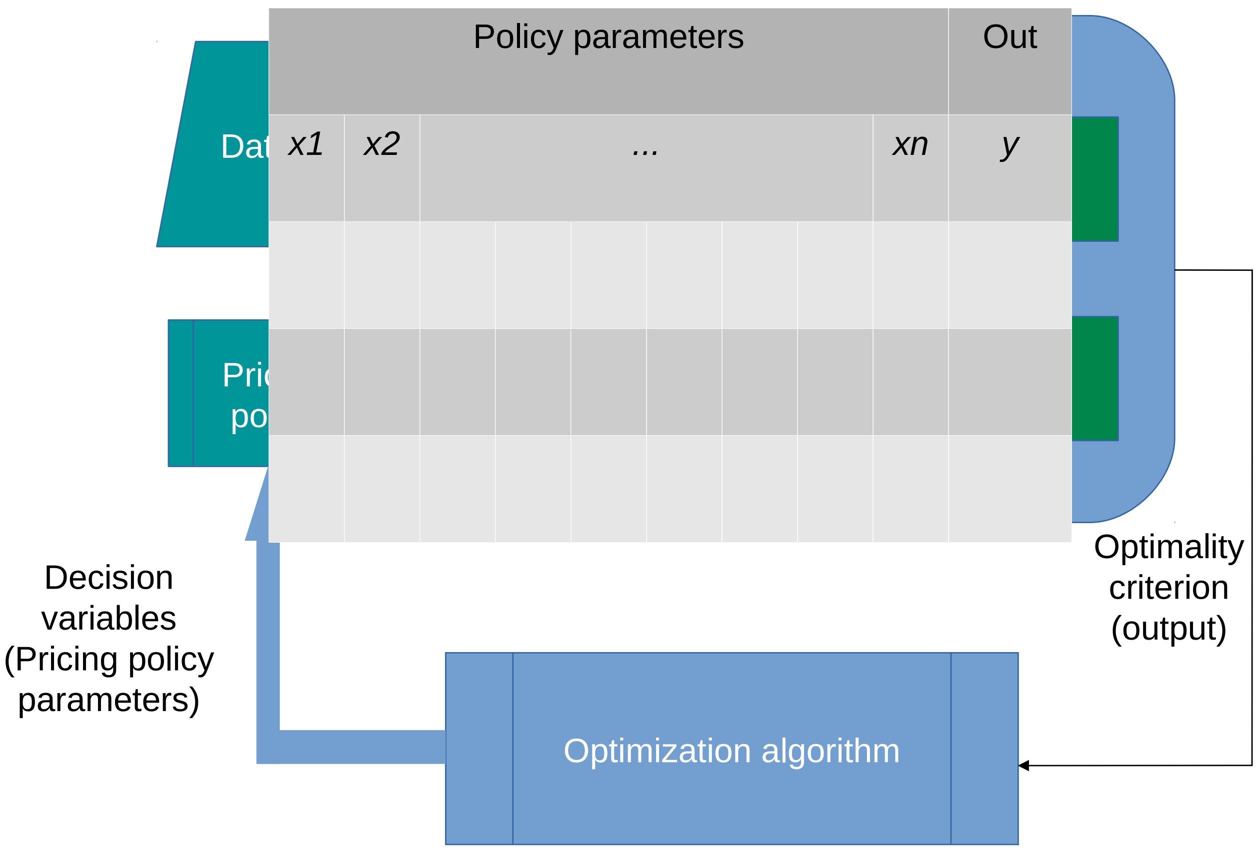
Building the customer model



Learning the simulator?

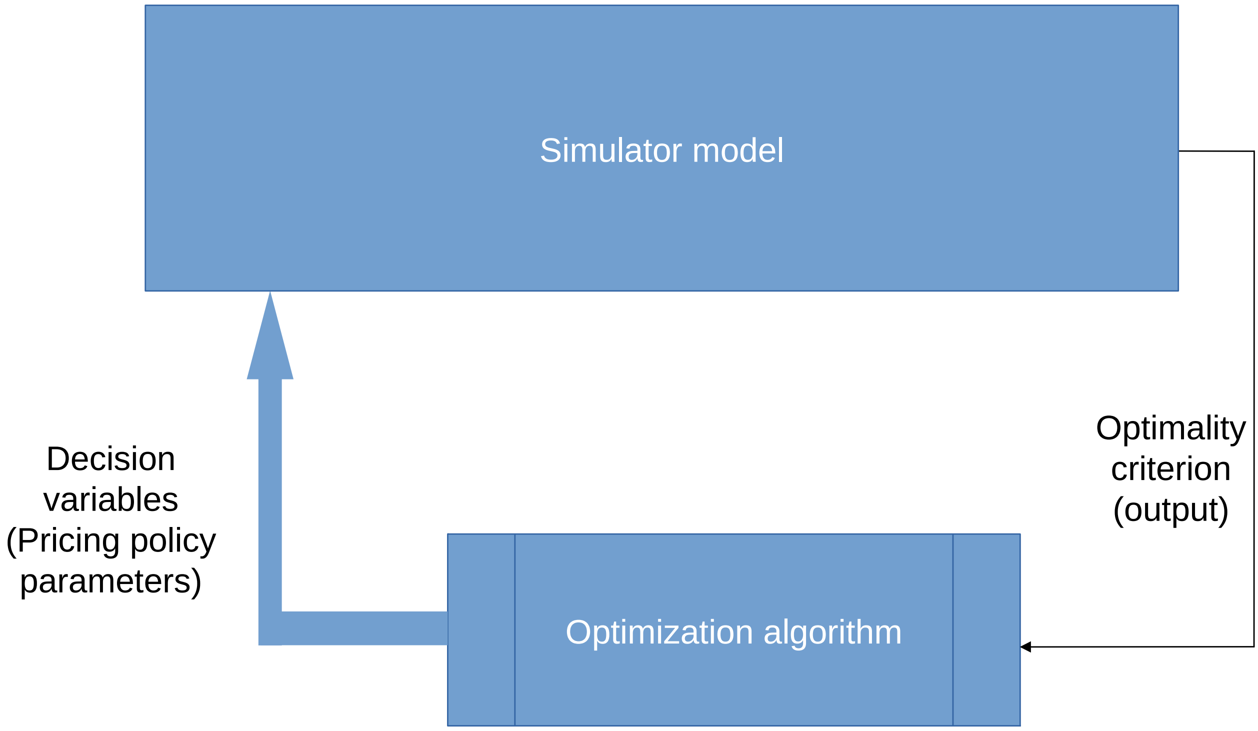


Learning the simulator?

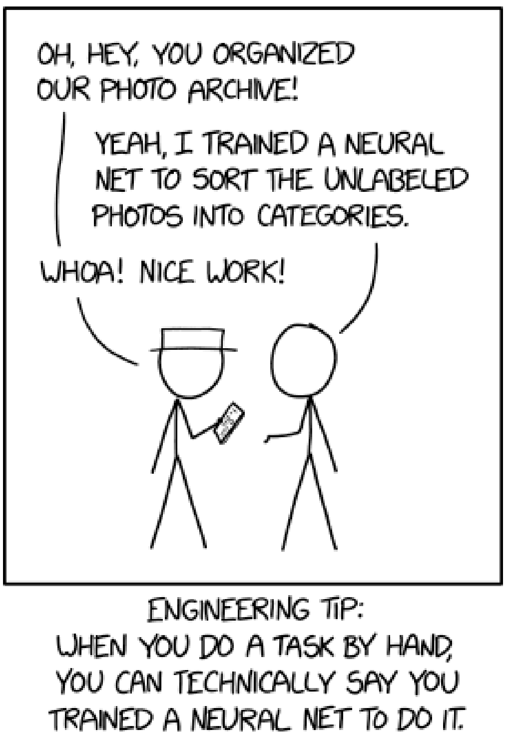


Policy parameters									Out
x1	*x2*	...						*xn*	*y*

Learning the simulator?



Wetware

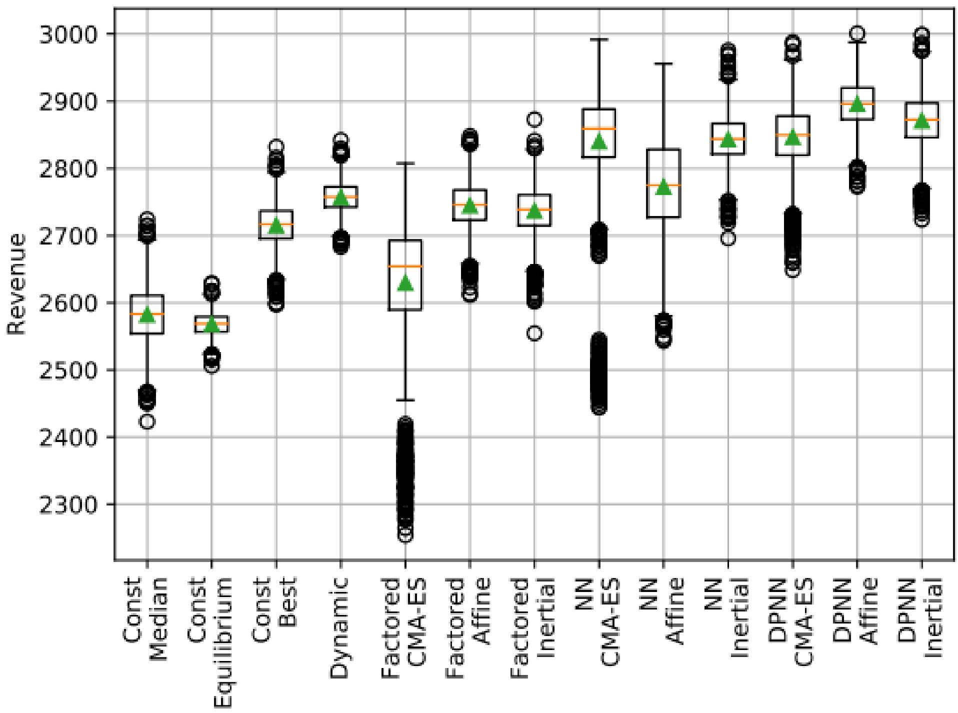


xkcd.com, July 8, 2019

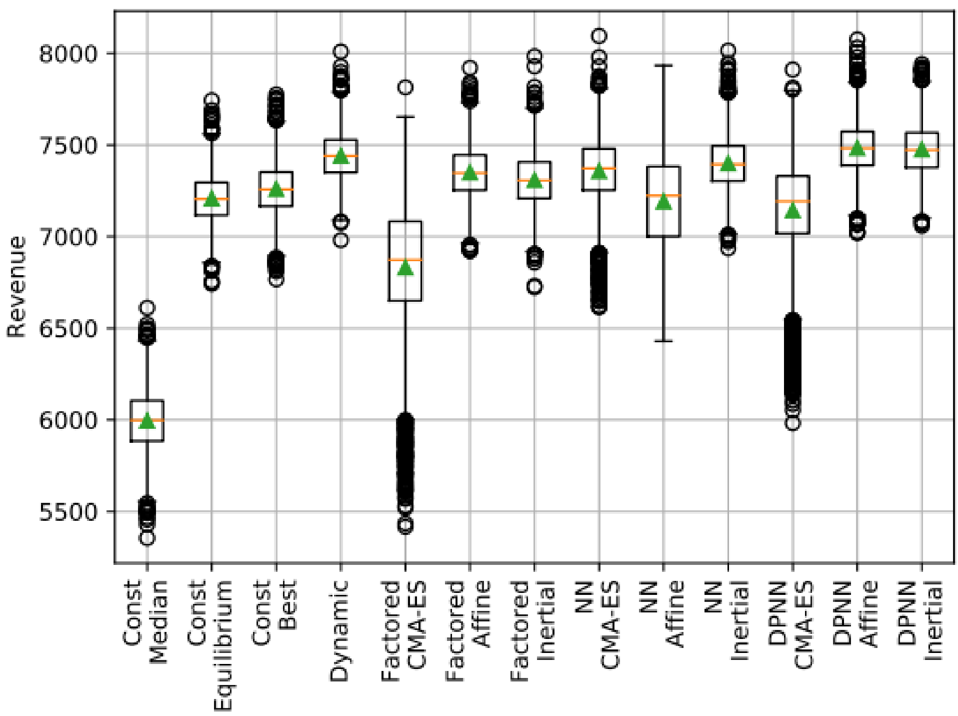
Restaurant owner:

I use a stack of Neural Networks.
I trained a simple NN, "Cindy", on basic tasks, such as rejecting unprofitable couples on a date.
Then a more powerful NN overrides it for more complex situations and outliers.

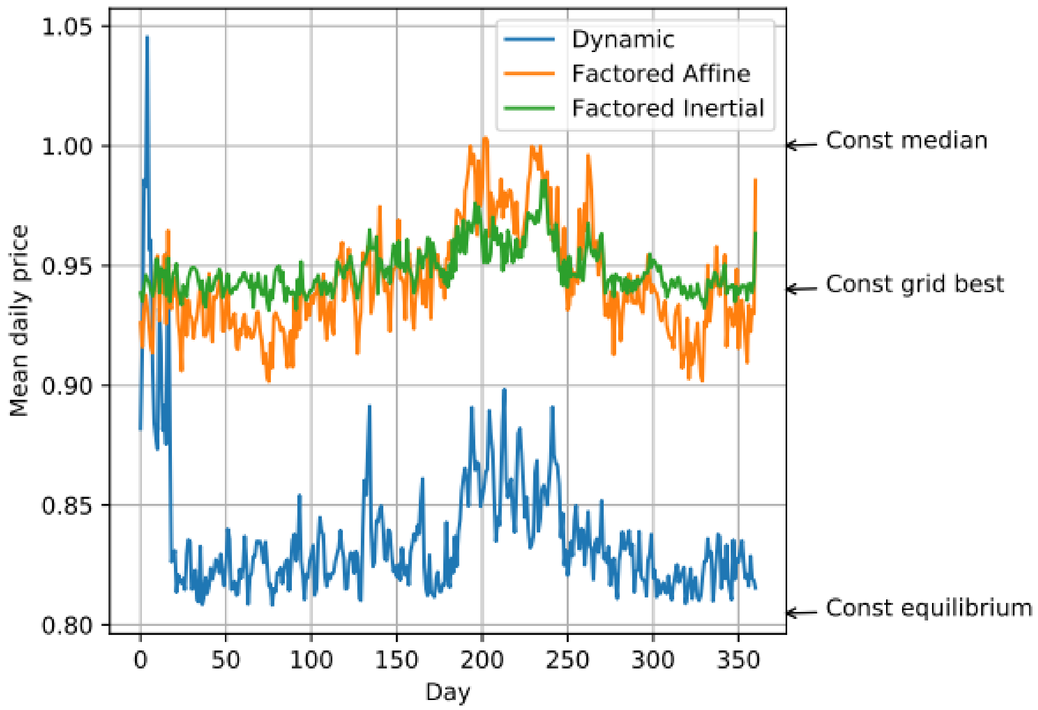
Pricing policy comparison: small hotel



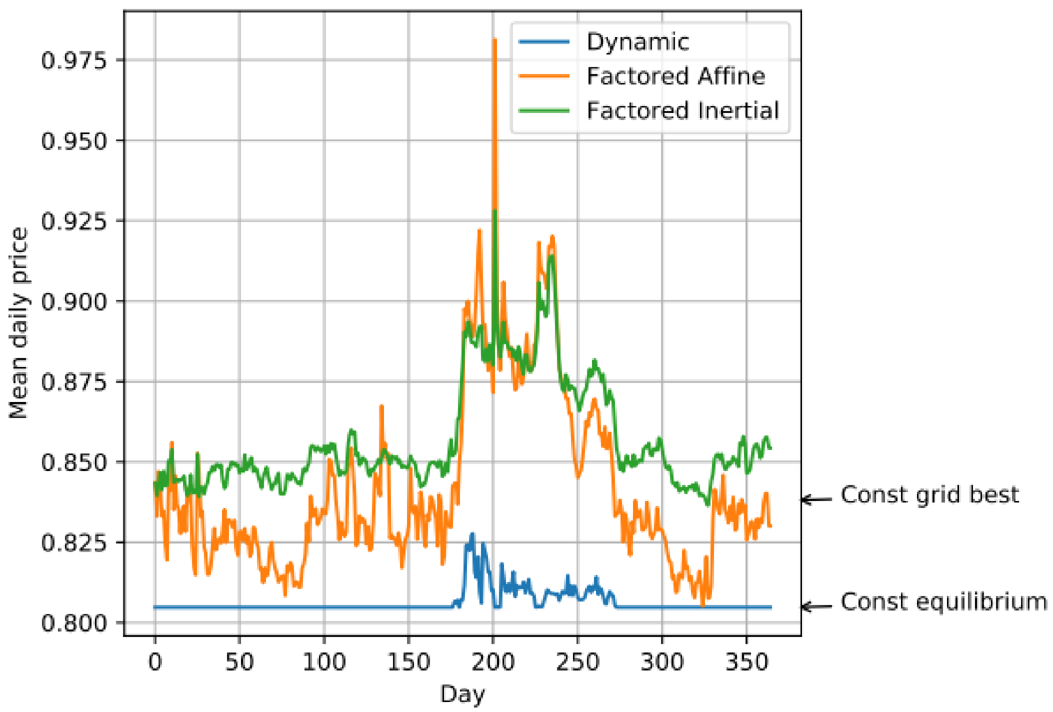
Pricing policy comparison: small hotel



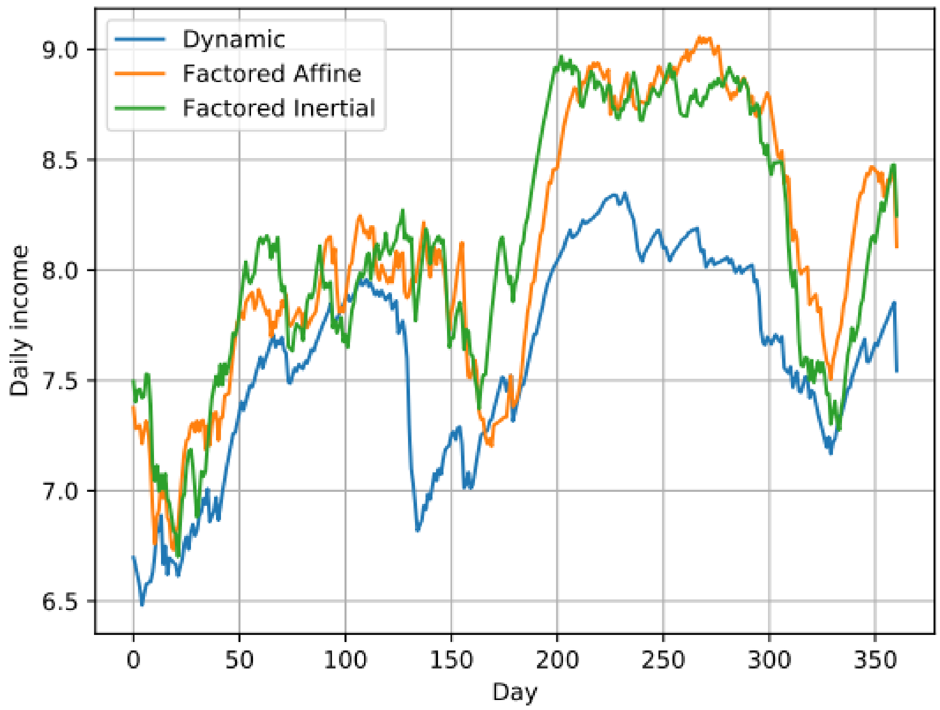
Pricing policy in action: small hotel



Pricing policy in action: large hotel



Pricing policy in action: small hotel



Pricing policy in action: large hotel