



MAXIMIZING SORTER THROUGHPUT

September 29, 2022



ABOUT US



Ari Siesser
Co-Founder and Chief Scientist



Felipe Haro
Founder and Simulation Consultant

OUR CUSTOMERS

Since 2019, Automation Intelligence has developed relationships with world-class movers, producers, and OEMs.





THE PROBLEM

TILT TRAY SORTERS

Commonly used in parcel sorting, distribution / order fulfillment centers, and baggage handling.



TILT TRAY SORTERS



THE PROBLEM

The way parcels are assigned to trays causes a

10%

decrease in throughput!

POOR SPACE UTILIZATION

Suboptimal greedy FIFO assignment uses more trays than necessary.

Induction Lanes Assignments

Lane 1



Lane 2



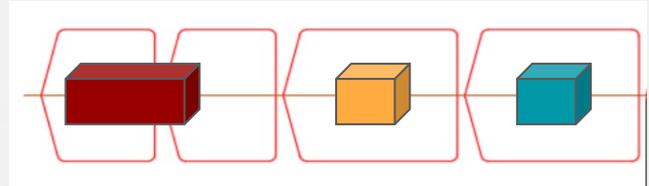
Lane 3



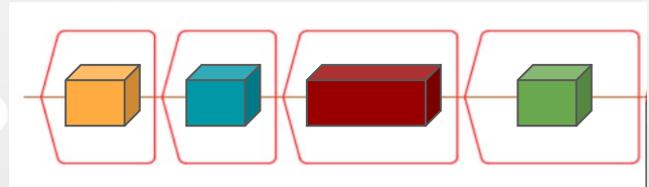
Lane 4



Naive:



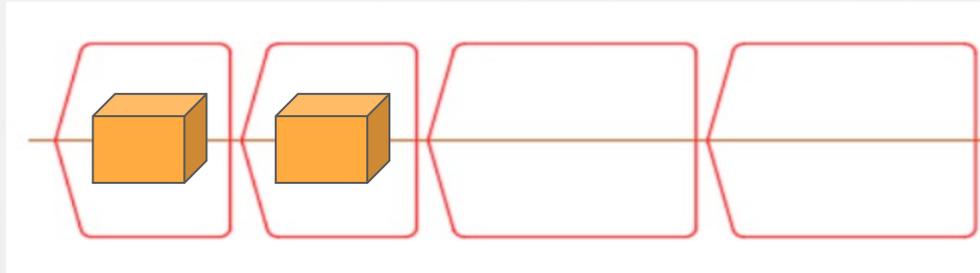
Optimized:



RECIRCULATION

Recirculating parcels prevent new parcels from entering the system.

- Cause: two parcels bound for the same destination are placed back-to-back.
- Output lanes need 3 trays to pass before consecutive discharges.





THE SOLUTION

MATHEMATICAL OPTIMIZATION

- A.K.A. Mixed integer linear programming, mathematical programming
- Translates a business / decision problem into a math problem.
- Each decision is encoded as a **decision variable**.
- **Constraints** and **objective** are expressed as linear functions of decision variables.
- Methodology comes with a guarantee!
- Problems can get very large and take a long time to solve.
- Modern solvers / algorithms much faster than enumeration.

FORMULATION

Decision Variables: which tray to assign each parcel to

$$x_{p,t} = \begin{cases} 1 & \text{if parcel } p \text{ is assigned to, tray } t \\ 0 & \text{otherwise} \end{cases} \quad \forall p \in P, t \in T$$

- Example: 20 parcels & 50 trays → 1000 decision variables

Objective Function: Maximize throughput (parcel density)

Constraints:

$$\sum_{p \in P} X_{p,t} \leq 1 \quad \forall t \in T$$

- No more than 1 parcel per tray
- Parcel longer than a tray → Needs to use the following tray as well
- Induction lane sequence
- Induction space
- Recirc prevention / discharge capacity



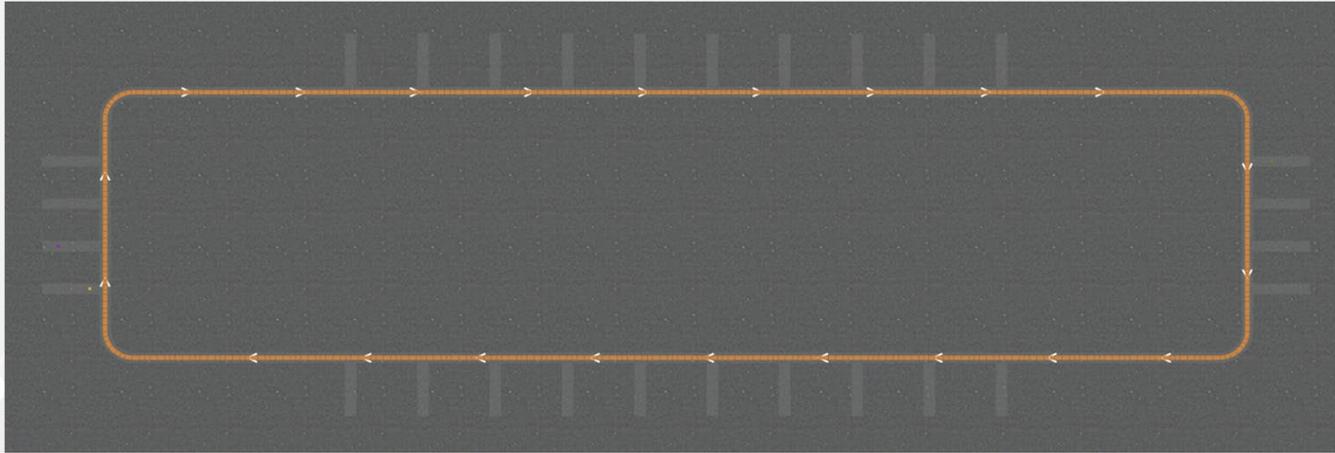
A

SOLUTION ARCHITECTURE

ARCHITECTURE



GUROBI
OPTIMIZATION



REST REQUEST



GUROBI
OPTIMIZATION



Parcels						
Arrival Sequence	Lane	Lane Position	Length	Destination	Parcel ID	First Possible Tray
1	1	1	1	2	P1	7
2	1	2	2	5	P2	8
3	3	1	1	8	P3	7

Trays				
Arrival Sequence	Tray ID	Length	Occupied?	Destination
1	T1	1	0	-1
2	T2	1	0	-1
3	T3	2	1	5

REST RESPONSE



GUROBI
OPTIMIZATION



Assignments	
Parcel ID	Tray ID
P1	T1
P2	T3
P3	T2

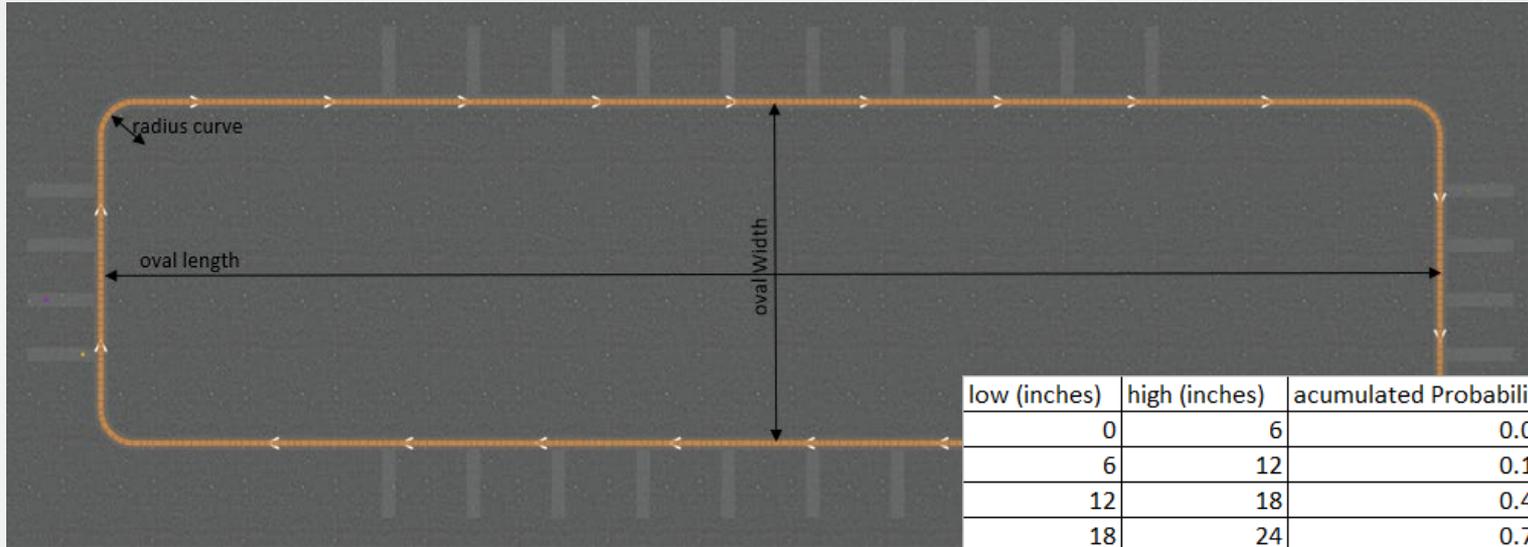


SIMULATION

USING ANYLOGIC

- The ability to make a system as flexible as we want
 - Data object input
- Making functionalities as customized as we want
 - Agent-Based to develop discrete-events models
- Communicating agnostically with other systems
 - UDP or REST API
 - JSON

FLEXIBILITY



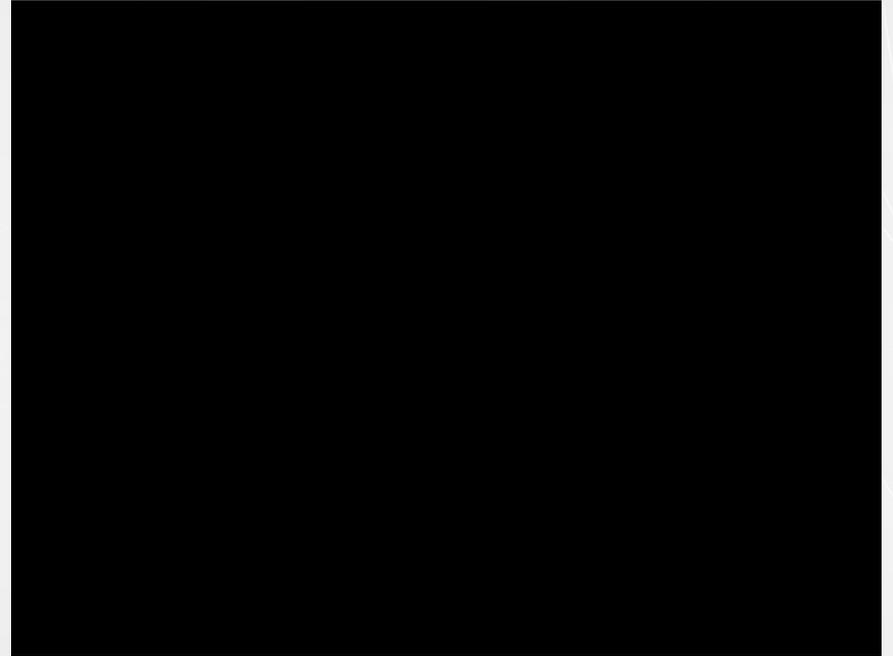
low (inches)	high (inches)	acumulated Probability	probability
0	6	0.0004	0.0004
6	12	0.1443	0.1439
12	18	0.4886	0.3443
18	24	0.7553	0.2667
24	30	0.8808	0.1255
30	36	0.9262	0.0454
36	37.5	0.9334	0.0072
37.5	42	0.955	0.0216
42	48	0.9691	0.0141
48	54	0.9826	0.0135
54	60	1	0.0174



Data Object Contains
All Characteristics
Of the System



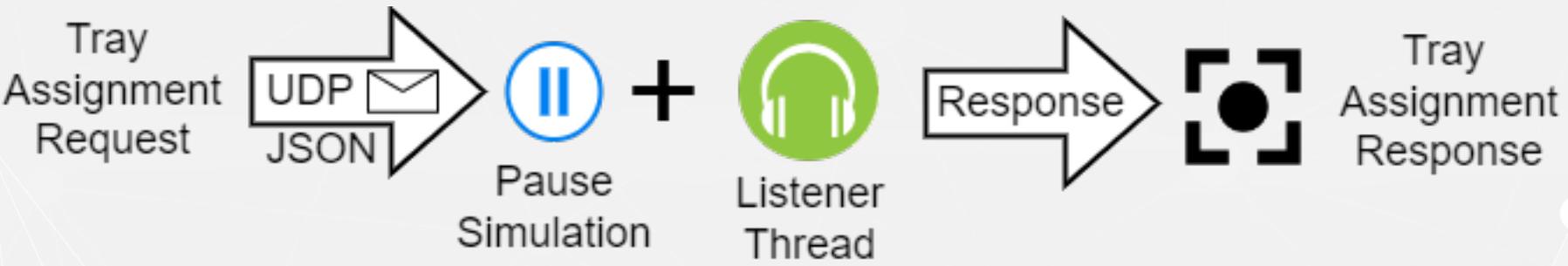
FLEXIBILITY



CUSTOMIZATION



UDP - JSON



REST API - JSON



Tray
Assignment
Request



POST



Tray
Assignment
Response

```
String uri = "http://127.0.0.1:5000/assign-parcels?algorithm="+main.algorithm=="0 ? "naive" : (
HttpRequest request = HttpRequest.newBuilder()
    .POST(HttpRequest.BodyPublishers.ofString(requestBody))
    .uri(URI.create(uri))
    .setHeader("User-Agent", "Java 11 HttpClient Bot")
    .header("Content-Type", "application/json")
    .build();

HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.ofString());
```



RESULTS

RESULTS



QUESTIONS + ANSWERS