How it works

How On-Off Works

The On-Off Algorithm works in such a way that it assigns aggression states/levels to the warehouses when:

- Utilization
- Queued Order Ratio
- Order per Courier
- Red Basket Count
- Available Courier Count are larger/less than certain thresholds or satisfy the combinations of certain logical conditions for those metrics.

These thresholds are re-calculated every day by analyzing past weeks' data for each warehouse.

There are 5 Aggression States: 1,2,3,4 and 5

Every warehouse has an Aggression Level that is assigned and updated by the On-Off Algorithm every 5 minutes

The Algorithm checks congestion metrics such as Utilization, Queued Order Ratio, Order per Courier, Available Courier Count and Red Basket Count every 5 minutes and compares it with the historic thresholds of each warehouse. If the momentary metrics are larger/less than certain thresholds or satisfy the combinations of certain logical conditions, then the algorithm assigns a larger/smaller Aggression Level to that warehouse.

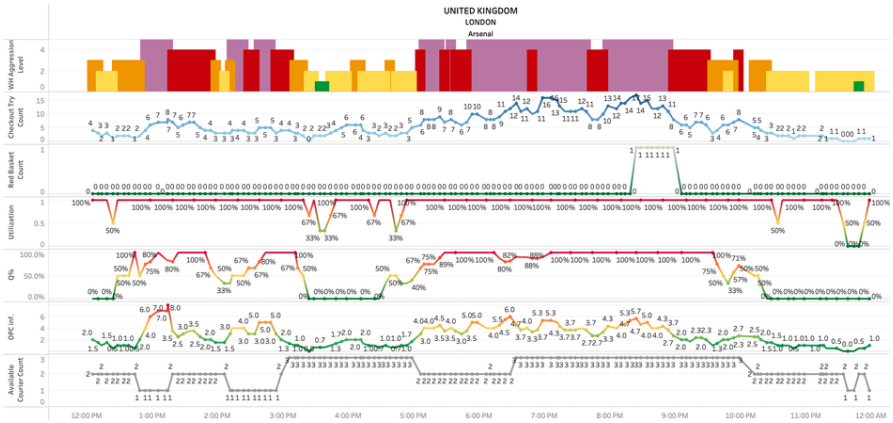
The orders can be broken down into 3 in terms of promo usage:

1. **Organic Orders**: Orders without a promo
2. **Orders with "Never Off Promos"**: These promos are visible to customers at a warehouse regardless of the aggression level of the warehouse
3. **Orders with "On-Off Promos"**: These are the Promos which have Open Aggression State Info on them. If the Open Aggression State of a promo is 2,3; then that promo is visible to the customer only if the aggression level of the warehouse is 2 or 3. Promos with lower Open Aggression States usually have lower minimum checkout amounts so that they are more attractive and desired more by the customer.

The On-Off Algorithm assigns aggression levels to warehouses and shows/hides (regulates) the On-Off Promos only.

- With **lower Aggression States** the On-Off Algorithm shows **more aggressive promos with lower minimum checkout amounts** to the customers and **steps on the gas.**

- With **higher Aggression States** the On-Off Algorithm shows **less aggressive promos with larger minimum checkout amounts** to the customers and **goes safe.**



A dashboard using which the performance of the algorithm is monitored

Notions & Abbreviations:

Available Courier Count: Utilized Courier Count + Idle Courier Count

RB Count: Red Basket Count (orders that received their first No Available Courier Error in the last 45 minutes and haven't been delivered yet)

Checkout Try Count/Demand/Potential Order Count: Active Order Count at That Moment + Orders that tried to checkout but received an error (e.g., No Available Courier Error) in the last 45 minutes and haven't been delivered yet

Util: Utilization (Utilized Courier Count / Available Courier Count)

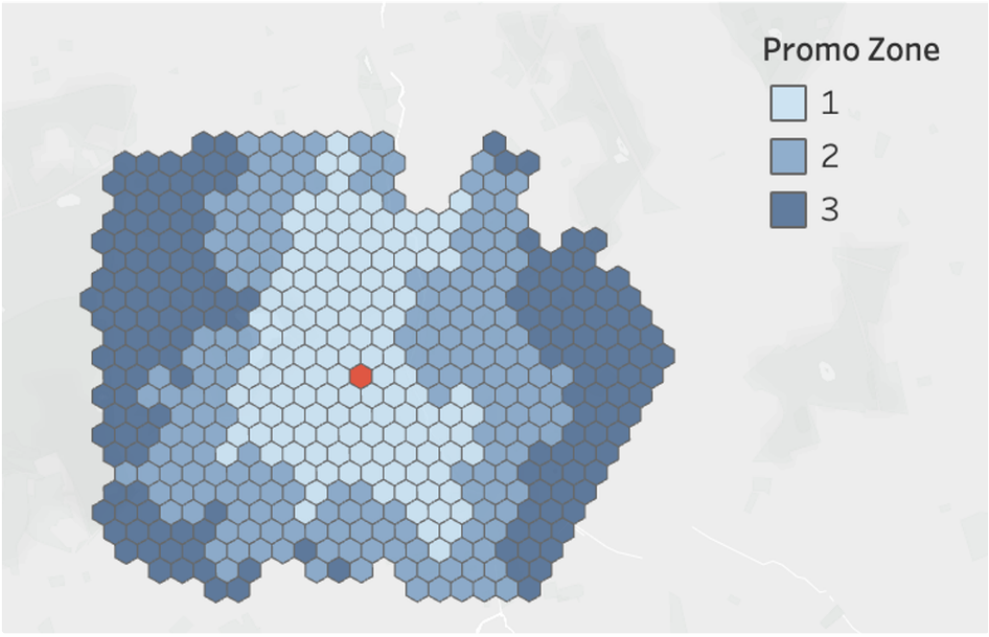
Q%: Queued Order Ratio (Queued Order Count / Active Order Count)

OPC: Order per Courier (Checkout Try Count / Available Courier Count)

Hexagon Based Development for On-Off Algorithm

With the Hexagon Based development now the customers at different zones of the service area (polygon) of a warehouse are subject to different aggression levels whereas in the old system the whole warehouse had a single aggression level. This enables us to have a varying and optimized aggression level distribution within the service area of a warehouse.

Users that are closer to the warehouse will be subject to lower aggression levels and see more aggressive promos with lower minimum checkout amounts whereas users that are far from the warehouse will be subject to larger aggression levels and see promos with larger minimum checkout amounts.



Red Hexagon is the warehouse

Delivering orders far from the warehouse is costly as they have high fuel/electricity cost, have high durations and lower the throughput. With the Hexagon Based system we aim:

- not to burn that much money with promos for the orders that are already more costly to deliver
- to slightly promote orders that are closer to warehouse as they are less costly to deliver with high throughput and low fuel/electricity cost

FAQ:

1. **How is it possible to see an order that checks out with a State 1,2 Promo while the warehouse is at State 3,4 or 5?**

Answer: Let's say a user opens the app when the warehouse aggression level is 2 and as a result sees a Promo whose Open Aggression State is 1,2. Once the user sees this promo he/she can continue seeing it for 15 minutes even if the On-Off Algorithm updates and increases the aggression level of the warehouse to 3,4 or 5. This 15-minute countdown restarts every time the users taps on the promos tab. This is a hard-coded business decision which ensures that the user does not lose that promo between "App Open" and "Checkout" events even if the On-Off Algorithm updates and changes aggression level of the warehouse in between (which happens every 5 minutes) to improve customer experience. So it is possible to see cases where an order checks out with a state 1,2 promo when the warehouse aggression level is 3,4 or 5 etc.