Meal Delivery Routing: The Grubhub Instances

Damian Reyes[1], Alan Erera[1], Martin Savelsbergh[1],
Sagar Sahasrabudhe[2], and Ryan O'Neil[2]

[1]H. Milton Stewart School of Industrial Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205
[2]Decision Engineering Department, Grubhub, Chicago, IL 60602

February 25, 2018

1 The Meal Delivery Routing Problem

Let R be a set of restaurants. Each restaurant $r \in R$ has an associated location ℓ_r. Let C be a set of couriers. Each courier $c \in C$ has an on-time e_c (when the courier goes on duty), an on-location ℓ_c (where the courier will be at time e_c), and an off-time $l_c > e_c$ (when the courier goes off duty). Let O be a set of orders. Each order $o \in O$ has an associated restaurant $r_o \in R$, a placement time a_o, a ready time (at the restaurant) e_o, and an order drop-off location ℓ_o. Furthermore, let t_{ij} be the travel time between any two locations i and j, e.g., restaurant locations, courier on-locations, and order drop-off locations, let s^r be the service time associated with the pickup of an order at a restaurant, i.e., the time a courier needs to park his vehicle, walk to the restaurant, pick up an order, and walk back to his vehicle, and let s^o be the service time associated with the delivery of an order at a customer location, i.e., the time a courier needs to park his vehicle, walk to the customer, drop off an order, and walk back to his vehicle. Couriers are compensated as follows. A courier receives p_1 per delivered order, but is guaranteed a minimum compensation of p_2 per hour. Thus the courier's compensation is $\max\{p_1 n,\ p_2(l_c - e_c)\}$, where n is the number of orders delivered during his duty period.

Orders from the same restaurant may be combined into "bundled-orders" with multiple drop-off locations. (The ready time of a bundled-order is the latest ready time of the orders in the bundle.) There is no limit to the number of orders that may be combined into a bundled-order.

The pickup time of an order at a restaurant is not smaller than the maximum of a) the order ready time and b) the courier arrival time to the restaurant plus half of the service time at a restaurant. (The departure time is the pickup time plus half of the service time.) The drop-off time of an order is the arrival time of the courier at the customer location plus half of the service time at a customer location. (The departure time is the drop-off time plus half of the service time.)

Couriers cannot pick up any orders after their off-time, but are allowed to drop off orders after their off-time.

Couriers do not execute any autonomous decisions while on duty. In particular, they always accept any instruction handed to them, and they always wait for (new) instructions at their on-location and at the last location of their active assignment. Note that this implies that couriers cannot receive instructions while executing an assignment and, thus, cannot be diverted while en-route to a restaurant.

We consider the following variations in operating environments:

- *Prepositioning.* Without prepositioning, the only instruction that can be sent to a courier is to pick up and deliver an order (possibly a bundled-order). With prepositioning, either the instruction to move to a restaurant (a prepositioning move) or the instruction to pick up and deliver an order can be sent to a courier.

- *Assignment updates.* Without assignment updates, once the instruction to pick up and deliver an order (possibly a bundled-order) has been sent to a courier, this assignment has to be completed before a courier can receive a new instruction. With assignment updates, when a courier arrives at a restaurant to pick up an order (possibly a bundled-order), an instruction can be sent to the courier updating the order to be picked up. For example, the initial instruction may have indicated that order o_1 had to be picked up and delivered, the update instruction may indicate that order o_1 and order o_2 have to be picked up and delivered (a bundled-order). Note that the only assignment update allowed is an update to the set of orders picked up at a restaurant.

1.1 Performance metrics

A number of performance measures are of interest. Some, like click-to-door, which is the difference between the drop-off time of an order and the placement time of an order, involve a target value, τ, and a maximum allowed value, τ_{max}. The primary performance measures for the meal delivery problem are:

1. Number of orders delivered.

2. Total courier compensation.

3. Fraction of couriers receiving guaranteed minimum compensation.

4. Click-to-door time: the difference between the drop-off time and the placement time of an order.

5. Click-to-door time overage: the difference between the drop-off time of an order and the placement time of an order plus the target click-to-door time.
6. Ready-to-door time: the difference between the drop-off time of an order and the ready time of an order.
7. Ready-to-pickup time: the difference between the pickup time of an order and its ready time.
8. Courier utilization: the fraction of the courier duty time that is devoted to driving, pickup service, and drop-off service (as opposed to time spent waiting).
9. Courier delivery earnings: courier earnings when considering only the number of orders served.
10. Courier compensation: the maximum of the guaranteed minimum compensation (based on the length of the duty period) and the delivery earnings.

Relevant summary statistics for measures 4-10 are: average, standard deviation, minimum, 10-th percentile, median, 90-th percentile, and maximum.

2 Instances

To facilitate the study of different algorithms and solution strategies, we have prepared a set of meal delivery problem instances as well as a small program to evaluate the feasibility and calculate the performance metrics of a solution to an instance.

Instances have been crafted based on historical data provided by our industry partner, with the intention to resemble realistic conditions in order and courier patterns. A sampling procedure on the historical record of orders is used to create instances with sizes ranging from 200 orders to 3000 orders. To ensure the anonymity of customers and restaurants, while also preserving some of the geographic patterns observed, all location coordinates from historical instances are perturbed (substituted by a uniform-random point sampled in a square region around them) and transformed to UTM (Universal Transverse Mercator) coordinates (while discarding the grid zone reference information).

Instances are derived from a singe day in history. Times are represented as non-negative integers (with zero representing the start of the day). All locations are represented as (x, y) coordinates. The travel time from $\ell_1 = (x_1, y_1)$ to $\ell_2 = (x_2, y_2)$ is the product of their Euclidean distance and a multiplier γ, rounded up to the next integer: $t_{\ell_1,\ell_2} = \left\lceil \gamma\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \right\rceil$. Note that this may lead to cases where the triangle inequality for travel times does not hold.

For each "base instance" generated, we include a set of variations:

Varying size of order and courier sets. Given an order set, two instance variations can be obtained by sampling 50% of the orders directly (uniform sampling with repetitions) or indirectly, through sampling of restaurants (where including a restaurant means including all the orders placed at the restaurant). Courier sets can be reduced in a similar fashion, sampling roughly the same proportion of shifts at each start time (rounding up to the nearest integer number of shifts when necessary). These variations preserve some of the geographical and temporal distribution of orders.

Varying travel times. "Short-distance" and "Long-distance" geographies can be induced by using different multipliers. This preserves the relative spatial distribution of orders. (Note that performance metrics associated with bundled-orders may deteriorate if travel times are longer.) Specifically, to build these instance variations, the original multiplier of an instance is reduced by 25%, producing a "Short-distance" version.

Varying structure of courier schedules. Schedules can follow a simple "rule-of-thumb" structure (4 shift start-times and 3 shift durations) or a more "flexible" structure (frequent start-times and many possible shift durations). In the instance set provided, to encourage fair comparisons, the total number of courier-hours in each schedule variation is kept constant.

Varying preparation times. Given an instance, we can modify the degree of "visibility" of upcoming orders by reducing or increasing the time it takes to prepare each meal, i.e. the difference between the time when an order is ready and its placement time. For given order placement times, longer preparation times increase the level of anticipation in the dispatching process. At the same time, given a target click-to-door time, longer preparation times increase the "urgency" of order assignments, as less time is available for a timely pickup once the order becomes ready. Specifically, to build these instance variations, the original preparation times are increased by 25%, producing instances with low and high levels of urgency.

In summary, from a given historical instance, a family of 24 instances is generated. Ten base instances have been derived from different days and cities, chosen to illustrate the various conditions found in realistic environments, providing a total of 240 meal delivery problem instances. Half of them have been produced with direct sampling on the order set, and the rest have been produced by sampling the set of restaurants.

Information for an instance is provided in four text files, with lines formatted as follows (in all files the first line contains helpful information on its content and can be ignored):

Listing 1: Restaurant set

```
Restaurant ID, x, y
```

Listing 2: Order set

```
Order ID, x, y, placement time, Restaurant ID, ready time
```

Listing 3: Courier set

```
Courier ID, x, y, on−time, off−time
```

Listing 4: Time and compensation parameters

```
Travel time multiplier, service time at pickup,
service time at drop−off, target click−to−door time,
maximum click−to−door time, per−order pay, per−hour pay
```

2.1 Solution evaluation

A solution to an instance can be summarized by a list of assignments of orders to couriers. A full specification also includes pickup and delivery times for individual orders, as well as the sequence of locations visited and the corresponding departure times for each courier.

Our solution evaluator receives as input three files. The first file contains one line for each pickup and delivery assignment (which represent decisions and corresponding decision times). The format of each line is as follows (in all files the first line is assumed top contain helpful information on its content and will be ignored):

Listing 5: Assignment information

```
assignment time, pickup time, Courier ID, Order ID, ..., Order ID
```

The second file contains one line per individual order delivered, formatted as follows:

Listing 6: Order delivery information

```
Order ID, placement time, ready time,
pickup time, delivery time, courier ID
```

The third file contains one line per movement of a courier. Moves corresponding to each courier are presented in blocs and, within each bloc, lines are written following the sequence in which they are executed. A block corresponds to the execution of an assignment, i.e., the move to a restaurant followed by the move to one or more delivery locations. In general, each line is formatted as follows:

Listing 7: Courier dispatch information

```
Courier ID, departure time, origin, destination
```

where ORIGIN is either 0, indicating the courier's on-location, RESTAURANT ID, or ORDER ID, and DESTINATION is either RESTAURANT ID or ORDER ID.

The evaluator checks that the following conditions are satisfied:

1. each order is assigned at most once;
2. assignments are not made before orders are placed;
3. a courier completes all the pickup tasks before the end of his duty period.

4. the pickup time of a bundle is at or after the latest ready time of the orders in the bundle;

5. orders are delivered in the sequence prescribed by the assignment;

6. the sequence of courier movements is feasible (no tele-transporting), and all arrival and departure times are consistent.

7. at the time of the pickup of an order or a bundle, the courier is in the corresponding restaurant.

8. at the time of the drop off of an order, the courier is in the corresponding diner location.

If a given solution satisfies these conditions, *i.e.* it is feasible, the evaluator returns a file with values for the aforementioned performance metrics. If the solution is infeasible, the evaluator returns a file indicating the ID of couriers and orders involved in an infeasible action.