Riders Dispatch Algorithm for food ordering industry

The aim of Dispatch Algorithm is to decrease delivery time while increasing the efficiency of food delivery riders. We need to make sure that the delivery time is as low as possible and to guarantee a fast and reliable delivery experience to the customers. Restaurants are partners on this operation, it is important that making sure the riders are sent on time. Riders are also a great asset: they are the face of the delivery company to the customer and it's important to keep them engaged and busy, making sure we use their time efficiently and cost-effective. (see *fig.1*)

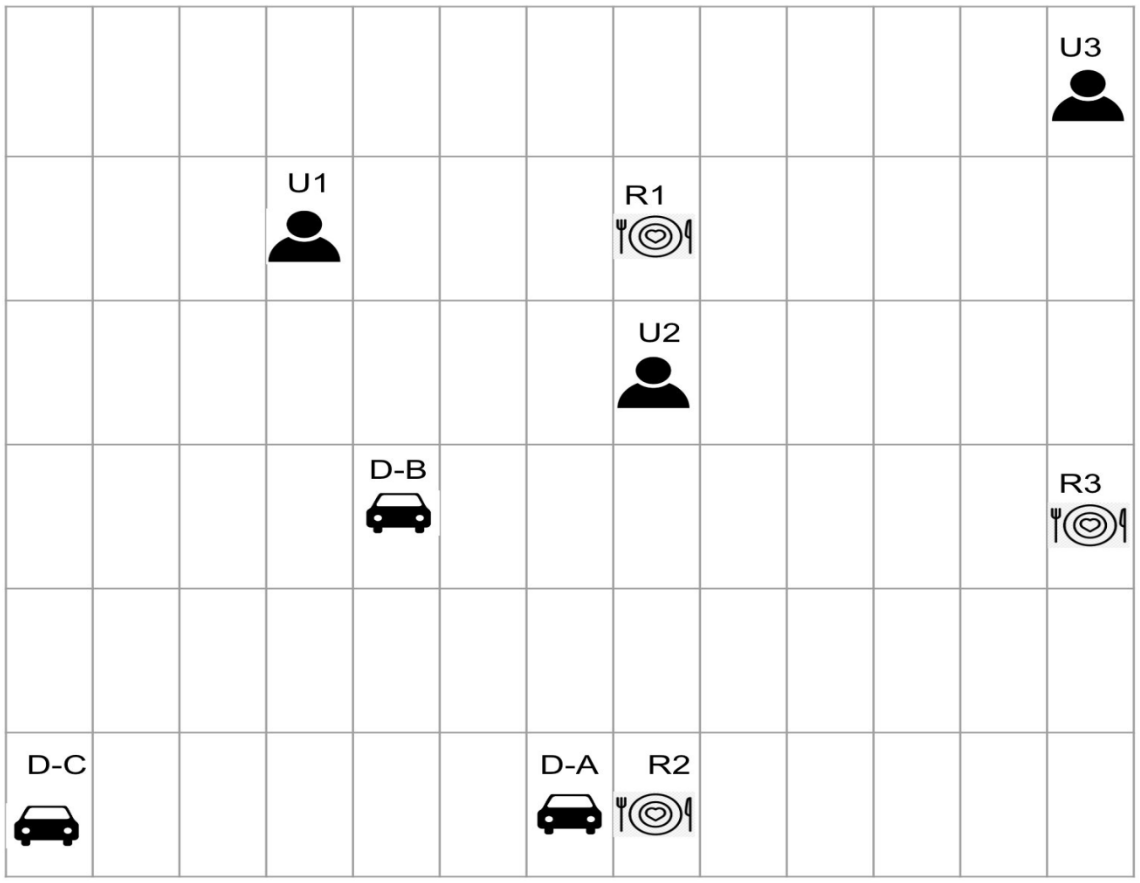

Figure 1: Map of customers, riders, and restaurants (Source: Delivery Hero)

Optimization of riders dispatch is very important because big companies play at large scale by dealing with millions of daily orders to deliver. A such bigness means that tens of orders comes every minute, so, the dispatching algorithm should be robust enough to provide good solutions in few of seconds.

The purpose of the dispatching problem is to assign the set of rider to a set of orders O (see *fig.2 and fig.3*) in the best way to minimize the sum of travelled time /or distance. Also, the customers' delivery time expectations must be respected or at least the tardiness made must be minimized. If a customer expects the order o to be delivered at the due date d_o, then it is delivered at the moment t_o. The tardiness equals then to $max(0, t_o - d_o)$. The sum of all tardiness made by all the orders is expressed as follow:

$$minimize \sum_{o \in O} max(0, t_o - d_o)$$

Business strategy has not to exclude any of the KPIs defined above, because the first KPI (the travelled time) make the drivers all the time engaged and busy, the second one (the tardiness) is important for the company's reputation. Therefore, the dispatch algorithm takes two KPIs into consideration. This called by bi-objective optimization algorithm.

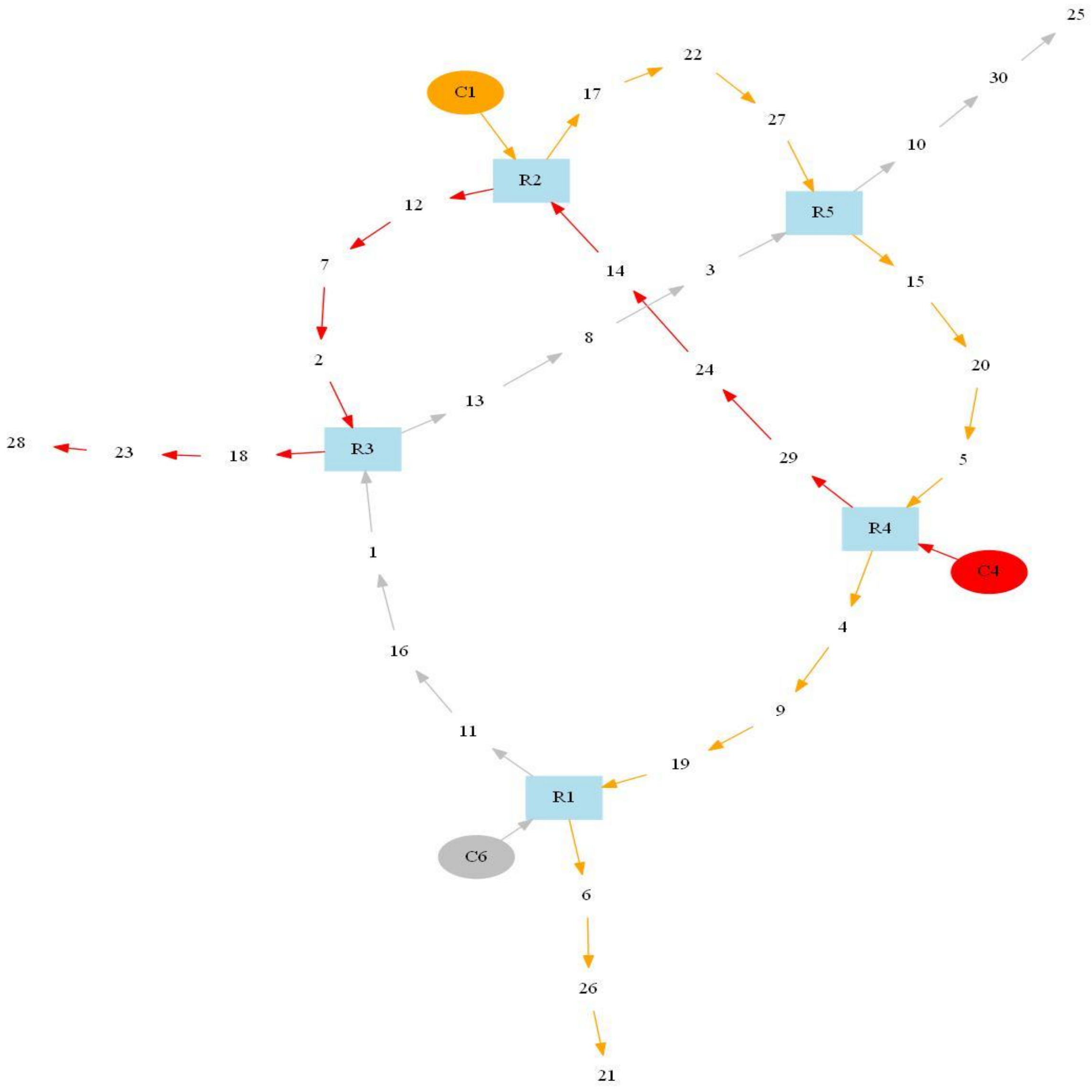

Figure 2: Riders Traveling Graph: three riders, five restaurants and thirty costumers

(Source: Java code)

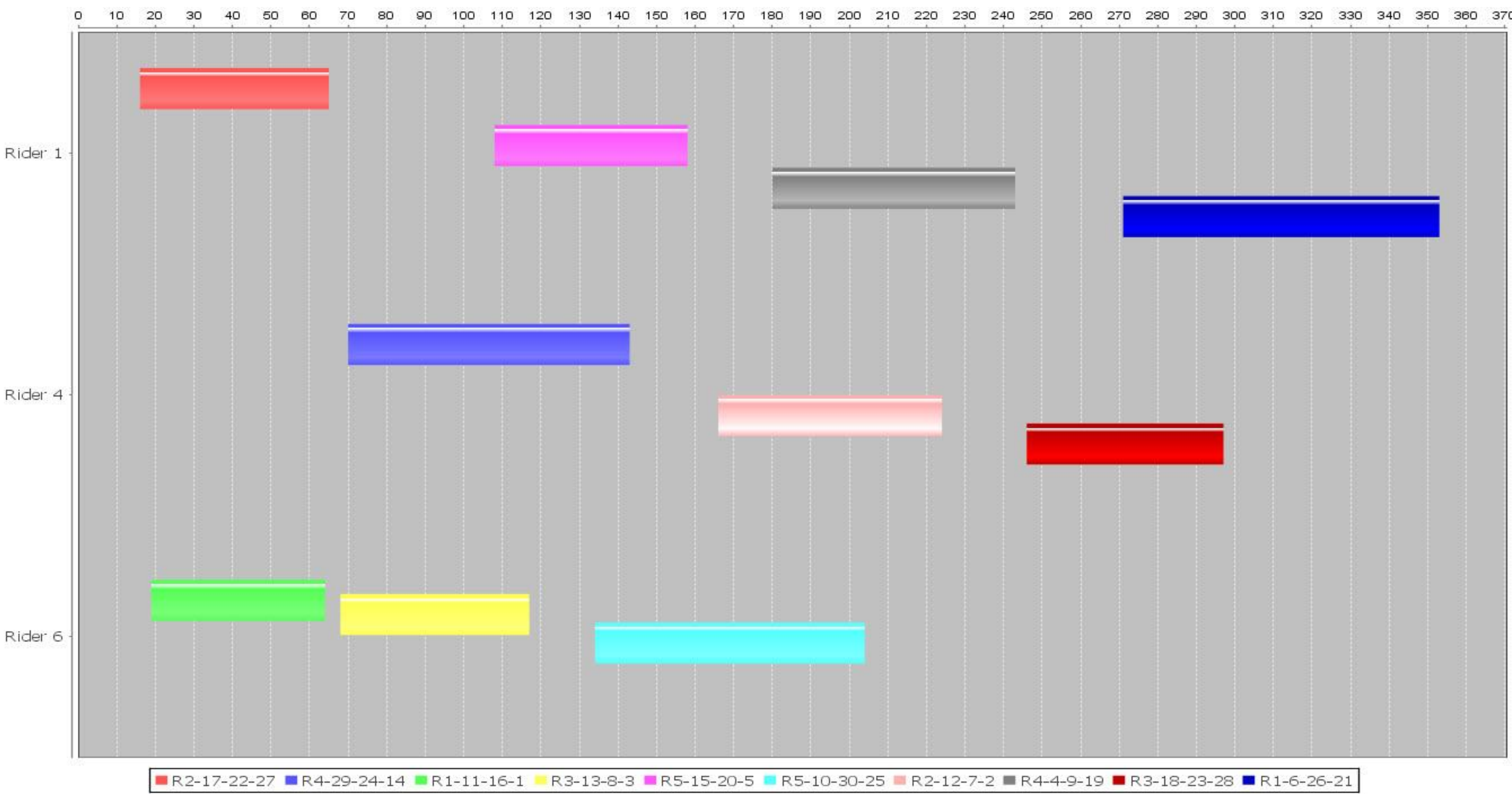

Figure 3: Riders Scheduling (Source: Java code)

The proposed to model to solve this problem is to consider it as a variety of Vehicle Routing Problem VRP (see *fig.4*). Such problem is known also as NP-hard: there is no known algorithm for solving this problem in polynomial time, and even small instances may require long computation time. Therefore researchers use all the time heuristics and metaheuristics algorithms to solve NP-hard problems in acceptable computation times. As instance genetic algorithm is the most famous metaheuristic.

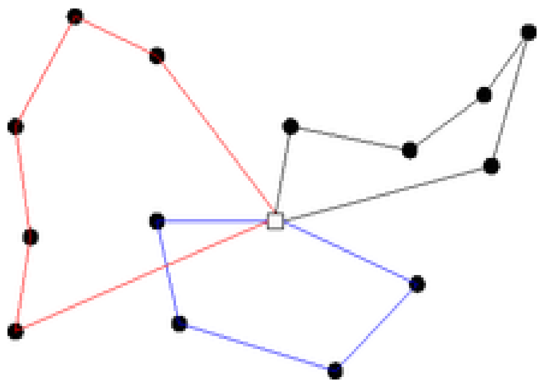

Figure 4: illustrating the vehicle routing problem (Source: Wikipedia)

In our case, the problem is solved heuristically by route-first cluster-second heuristics and modern genetic algorithms for VRPs. the solution is described as follow: a set of customers presented by nodes dispatched in the map. These nodes need to be visited by a set of riders (The depot has no meaning in our model). The goal of the algorithm is to define the order of all deliveries (called giant route) and the rider's assignments (routes) provided by Split Algorithm by taking into consideration as constraint a maximum number of customers served by one route. A set of deliveries are assigned to one single rider if these deliveries share the same restaurant as same supplier. Then rider makes the deliveries one by one in the defined order by giant route.