GDZ Workforce Assignment and Routing Systems Design Project

2023 – 2024

December 4th, 2023

Project Team: Ahmet Yücel Tanrıverdi
Harun Yüksel
İslam Valehli
Sami Bardakçı
Youssef Nsouli

Project Advisors: Meral Azizoğlu
Sakine Batun

Sponsoring Organizations: Presify
GDZ Elektrik Dağıtım A. Ş.

Company Advisors: Ertuğrul Özer (Presify)
Serhat Taysi (GDZ)

INDUSTRIAL ENGINEERING DEPARTMENT
MIDDLE EAST TECHNICAL UNIVERSITY
06800 Ankara, Turkey

Table of Contents

Introduction

Presify

Specializing in crafting sophisticated big data analytics software catered explicitly to energy markets and systems, Presify harnesses the prowess of high-level statistics, advanced data analysis, and profound engineering expertise. The bedrock of Presify's unparalleled ability to engineer advanced big data analytics software lies within the collective expertise and deep-seated knowledge of its astute staff of Industrial Engineers.

GDZ and Service Distribution

GDZ, one of the 21 electricity distribution companies in Türkiye and Presify's customer, ensures reliable distribution of electricity in Izmir and Manisa covering 13,123 km^2 (about half the area of Belgium) of area across 47 districts and 2,383 neighborhoods and serving 3.6 million customers. The firm has distributed the service area into multiple regions (Table 1):

Regions	Number of Service Centers
İzmir Metropol	5
North İzmir	6
South İzmir	12
Manisa	16

Table 1: Number of service centers in each region

On-site Teams and Jobs

Each service center houses multiple on-site teams that are tasked with going around the respective region and completing the tasks allotted to them. There are a total of 106 teams working one, two, or three shifts and composed of two or more workers. It is important to note that if a team works more than one shift, then for that team, the individuals working in that team change from one shift to the next. These tasks include:

- **Preventive and General Maintenance**: periodic inspection and maintenance of electrical infrastructure, such as transmission lines, transformers, substations, and distribution equipment, to prevent breakdowns and ensure smooth operation.
- **Fault Repair and Troubleshooting**: responding to power outages and quickly addressing power outages and electrical faults; emergency responses done by mobilizing teams to immediately resolve critical errors and security risks.

- **Meter Reading and Billing**: collecting data from your customers' electricity meters to accurately track electricity consumption; calculating customer bills.
- **Customer Service**: handling customer inquiries, complaints, and service requests; enabling new power connections for residential and business customers.
- **Regulatory Compliance**: compliance with regulations and standards established by regulatory authorities to ensure safety, environmental responsibility, and fair pricing.
- **Environmental and Safety Commitment**: promotion of employee and public safety practices; implementation of environmentally friendly initiatives; services given by field service technicians work; testing and calibration of electrical equipment.

Vehicles

In addition, GDZ has five types of vehicles, which are (from most abundant to least): boom trucks (trucks equipped with cranes), 4x2 cars, vans, automobiles, and 4x4 cars. There are a total of 155 vehicles.

Some jobs can only be done by certain types of vehicles. Moreover, the vehicles are not assigned to the on-site teams on a permanent basis, but are allocated daily, according to what the team is expected to do throughout the day and according to what vehicle is available at the service center at that time. However, the vehicles are assigned to each service center, meaning it is difficult to trade vehicles between service centers on a daily or weekly basis.

Other Details

- When it comes to a company’s customer base, there are five types of subscriptions: domestic, industrial, agricultural, commercial, and general street illumination. The urgency of jobs is affected by the nature of the subscription. For example, faults in areas with many commercial subscriptions means that there is a high chance that a hospital or fire station may have suffered a blackout, which would need immediate attention.
- The flow chart of a job can be expressed in figure 2:

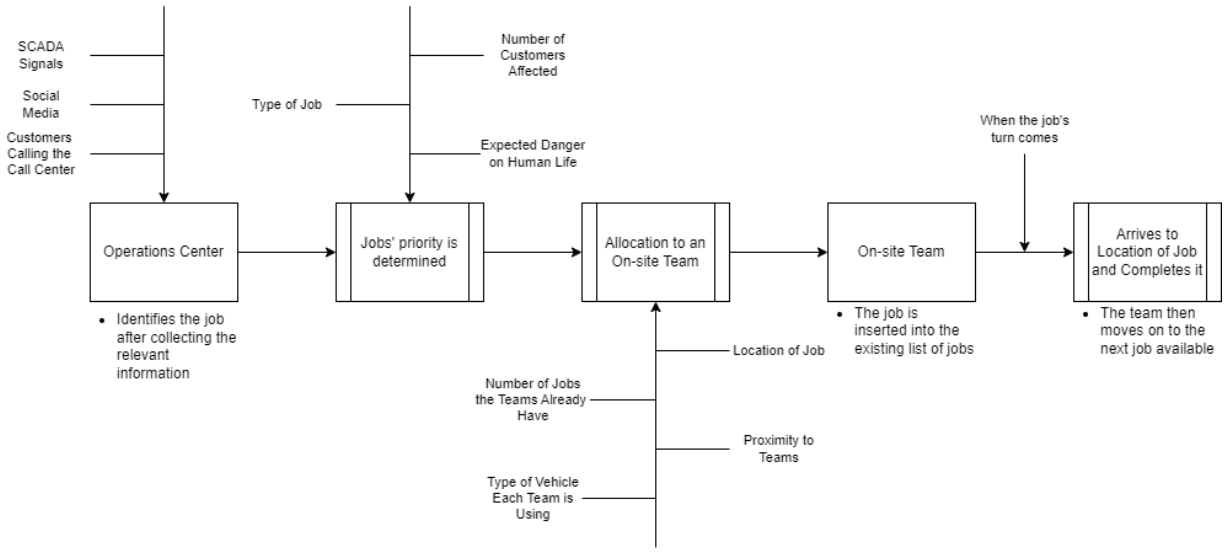

Figure 2: Flow chart of a task.

- Finally, it is imperative to touch on the crux of the system: its dynamicity. If the system's jobs were always available the time the teams start their shifts, then the problem would be as simple as a $1||\sum T_j$ or a $1|r_j|\sum T_j$ scheduling problem. However, some jobs do not even exist at the start of the shifts, and randomly appear throughout the day, forcing GDZ and its Operations Center to make multiple reassignments and reroutings in one single day.

Problem Identification

GDZ has been mainly complaining about the inefficiency of the completion of the jobs assigned to the on-site teams, despite the overall utilization rate of the teams being around 80%. Many jobs are not being completed as fast as they should be, and jobs are being accumulated from one shift to the next. To inspect the core of the problem, it is important to distinguish the general stages of completing a task (Figure 3):

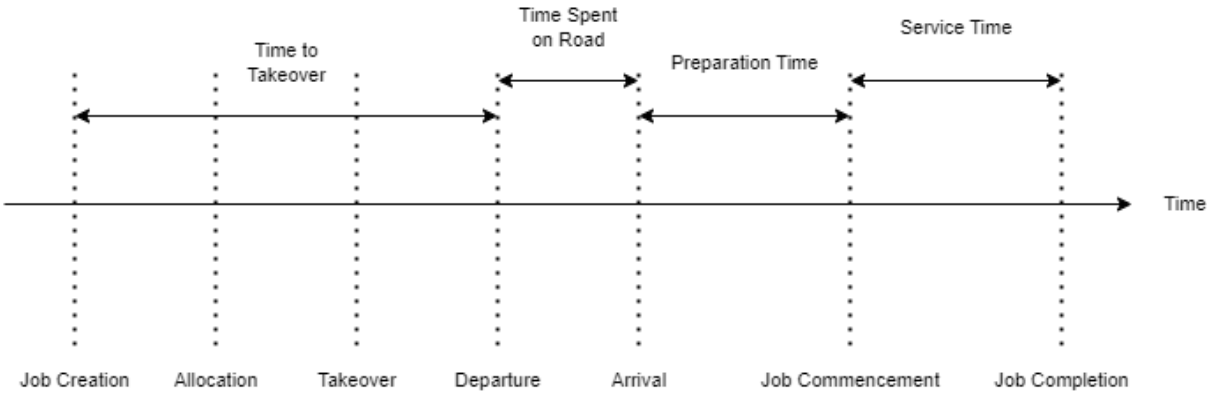

Figure 3: Stages of task completion.

The averages of the times demonstrated in the above figure are show in figure 4:

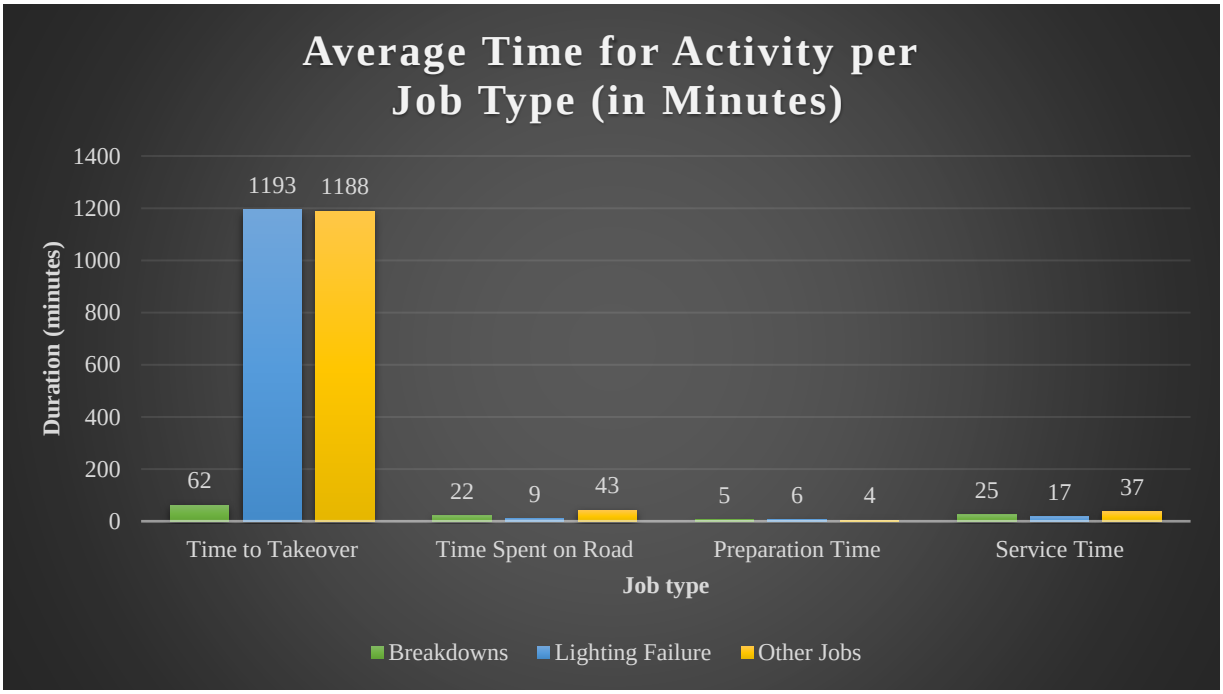

Figure 4: Average time needed for activities per job type, in minutes.

As indicated, the time to takeover is incredibly high, yet the other activities do not take as much time. Upon further inspection, the reason is determined as that there are jobs being pushed from one shift to the next, because the teams are not completing their allotted jobs in their designated shift, then in the next shift, the teams get more and more jobs, causing the time to takeover to eventually explode. An illustration of the number jobs remaining from the previous shift and jobs that get appear mid-shift is shown (figure 5):

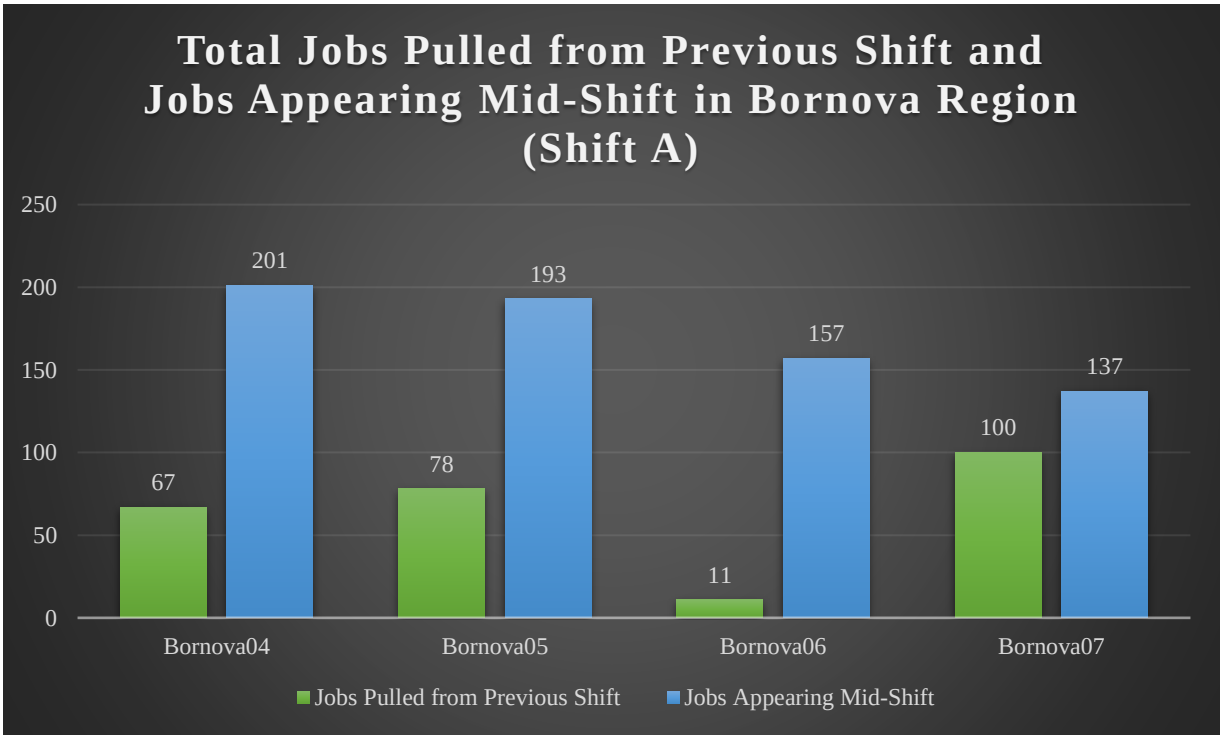

Figure 5: Total jobs pulled from the previous shift and total jobs appearing mid-shift for the teams in the Bornova Region throughout shift A (for the month of October).

In addition to this, GDZ has complained about how they are facing difficulty prioritizing tasks; some jobs are being completed before a more important and closer job.

Problem Symptoms

Upon further inspection, the problem seems to be of three interlinked subproblems:

1- **Allocation**: teams are not being allocated effectively, leading to teams being overworked while others are underworked.
2- **Routing**: the sequence followed by the jobs is far from optimal.
3- **Re-assignments and Reroutings**: when jobs that pop up mid-shift fail to be inserted into the task schedule in an effective manner.

Allocation

Although the teams are designed to one service center and one region, there are even workload imbalances between the teams in the same service center. Continuing with Bornova's shift A teams, their average daily workload is shown in figure 6:

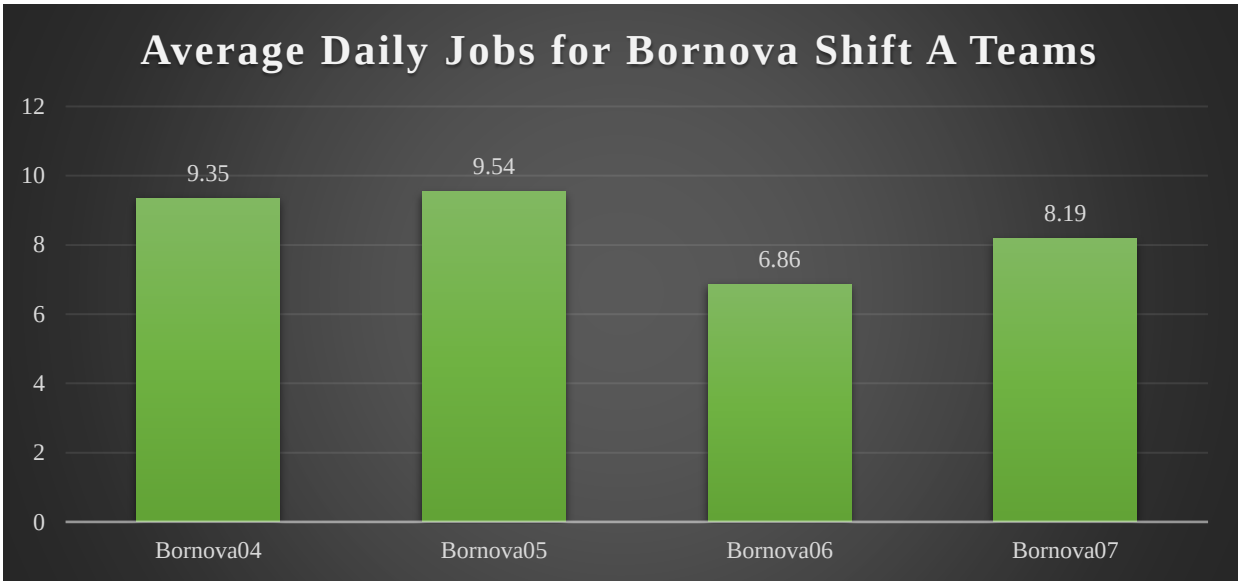

Figure 6: Average daily workload of Bornova's shift A teams over one month.

These variations lead to some teams being overworked while others being underworked. As such, some teams will have leftover jobs and have them pushed over into the next shift, while other teams stay waiting for jobs, which is wasteful.

Another problem encountered with allocation is one related to vehicle models. Sometimes, a team has to postpone a job because they do not have an appropriate vehicle for the job. The data for this is not presented currently, however it can be used to see how many jobs the teams have to postpone daily due to vehicle mismatches.

Routing

Proving suboptimality in routing has been a challenge, due to the dynamic nature of the problem. It is easier, however, to prove that it is problematic. A peak into one of the days reveals that, sometimes, some jobs are allocated before the shift starts, yet some of these jobs are completed after the shift ends. Some of said jobs are high priority. An example from the Burnova region is given in the illustration below (figure 7):

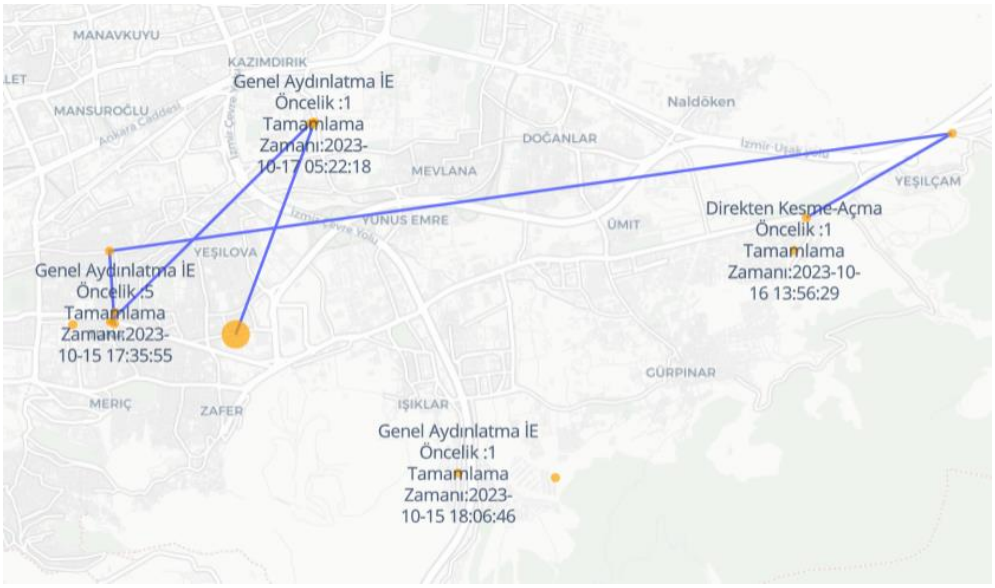

Figure 7: A roadway of the jobs completed by Bornova04 on the 15th of October, between 8:00am and 5:30pm.

The lines in figure 7 show the direction of the team moving from one job to the next, and the larger circle indicates the last job of the day before the shift ends. The unconnected circles are jobs who were allocated to the team before 8:00am but were completed after 5:30pm. As can be seen, some of these jobs have a priority of one, yet were left unresolved for more than a shift's worth of time, which proves that the routing of the team is problematic.

Note: more examples are available in Appendix A.

Another way to prove that the routing is suboptimal is to simulate the system with a heuristic. A (heavily) modified version of the nearest neighbors method is used for historical data. The costs of the arcs were calculated as $t_{ij}\sqrt{(priority)_j}$, where t_{ij} is the time needed to go from node i to node j, and is multiplied by the square root of the priority of the job at node j. The results yield a similar or better result cost-wise in more than half of the simulated days and are still faster in more than half of the days. Details are in Appendix B.

Re-assignments and Reroutings

The main problem with re-assignment and rerouting is that new jobs are not being inserted into the teams' schedules appropriately. Sometimes jobs are being completed despite their being a more important and closer job nearby. GDZ considers that if two jobs are present

to a team and one has a higher ranking than the other (the ranking considers both job type, priority, and distance), and the team goes to the job that has a lower ranking, than that is a ranking violation. Upon sifting the data, at least 12% of the routings consisted of ranking violations. Moreover, most of these ranking violations included jobs that had appeared mid-shift.

Conclusion

Possible Approaches to Solution

First, to approach the allocation issue, it is possible to consider the current workload of the teams in each region before allocating a job to them. It is also possible to divide the region into areas of equal expected "demand," based on historical data.

Second, for the routing problem, a mathematical model is certainly in play, and the encouragement of using google maps and scheduling principles can be a great help to reduce the time spent on the road. Weighing jobs by their priorities and their potential endangerment to human life and/or monetary assets could help improve overall performance.

Finally, for the last subproblem, it could be important to discuss an effective way to add newly created jobs into the mathematical model and have it calculate a new route quickly and still produce a near-optimal solution.

Future Plans

As it lies now, when it comes to monetary cost, little has been done. We will be requesting data regarding the financials of the system and analyzing the tradeoffs of customer satisfaction and costs.

Moreover, we recognize the importance of understanding the day-to-day life of the workers in the on-site teams and to understand what these teams expect throughout their day and understand sources of delay. So, it is important to have a discussion with some of the workers.

Finally, initial mathematical models will be drawn, starting with indices and parameters, and continuing with constraints.

Appendix A

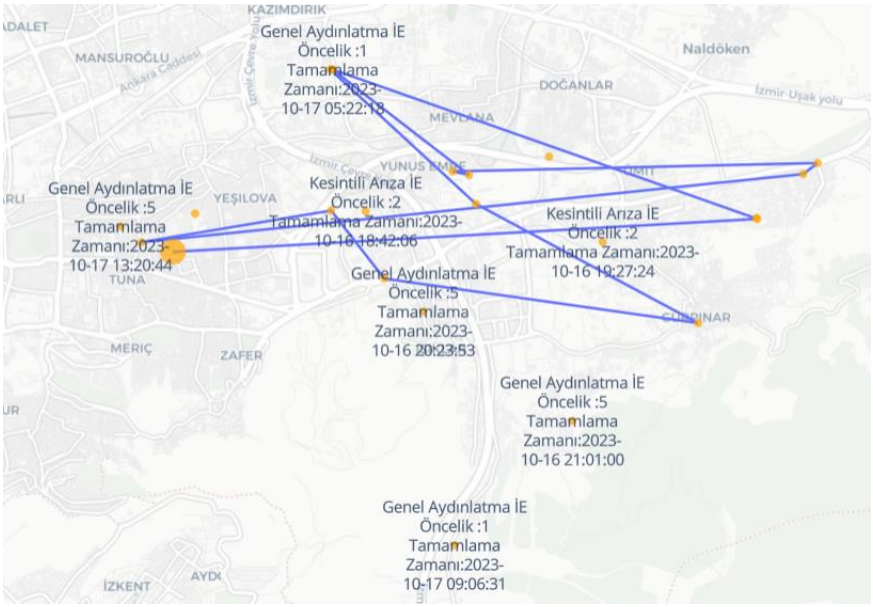

Figure A1: Roadway of Bornova04 on the 16th of October.

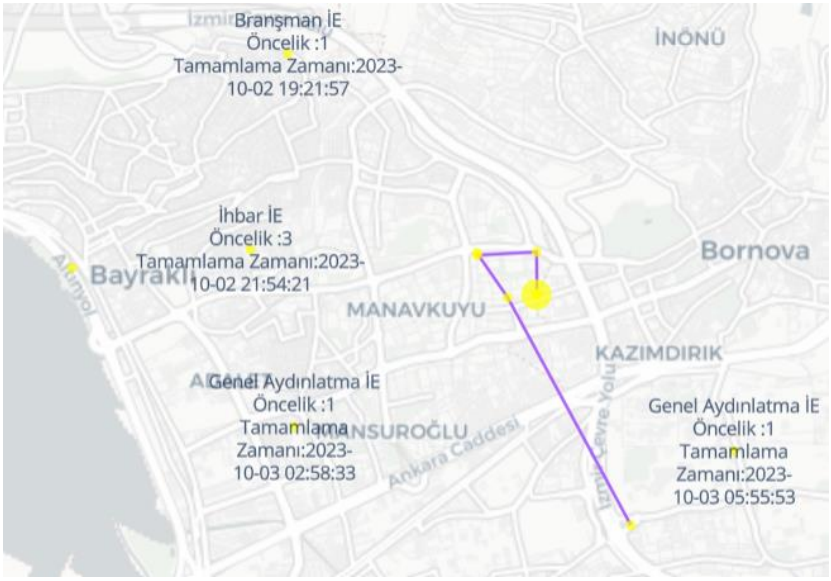

Figure A2: Roadway of Bornova07 on the 2nd of October.

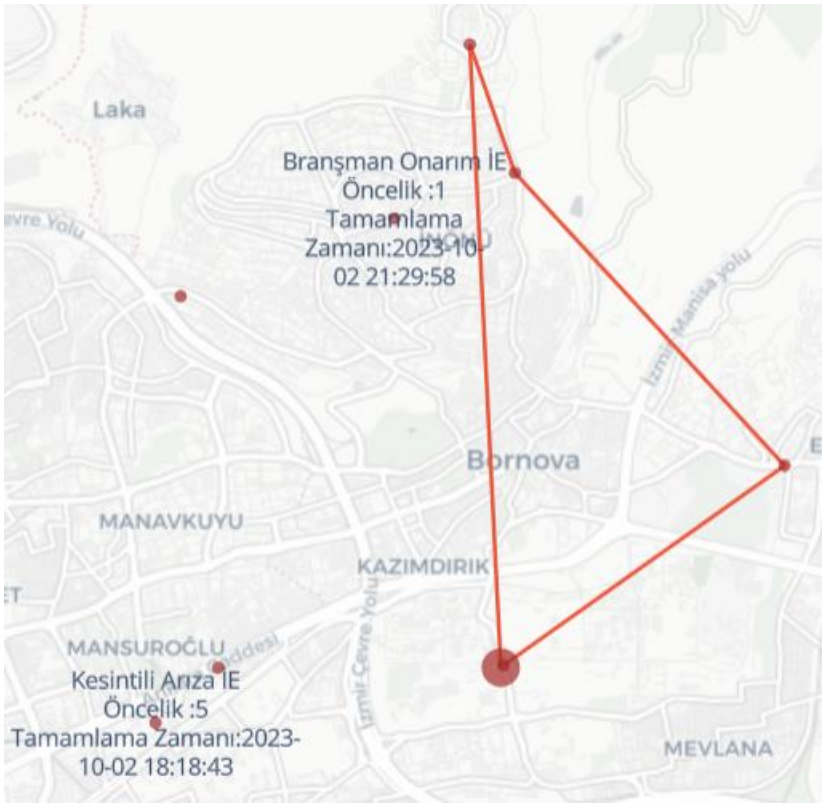

Figure A3: Roadway of Bornova05 on the 2nd of October.

Appendix B

Description

The nearest neighbor algorithm typically is used for travelling salesman problem (TSP) where the salesman seeks out the nearest node from wherever they are at any point in time. Its benefits are that it is a very short-term heuristic, which is helpful, like in this case, to be able to quantify suboptimality faster, without needing to fully simulate the system, bit by bit.

The heuristic used here is a modified version of the normal one, due to the dynamicity of the problem at hand; only a handful of jobs are "discovered" when a shift starts, and new jobs pop up mid-shift, and it is imperative to give these new jobs no less priority only due to their earliness. In addition, some jobs have a priority of 1, which is considered maximum priority. These jobs must be completed as fast as possible, regardless of incurred costs.

Algorithm

Let C represent the completed jobs, D represent the discovered but uncompleted jobs (waiting jobs), and U represent the jobs that have NOT yet been discovered. Suppose the total number of jobs is n.

0) Find out what jobs have been available or "discovered" before the shift starts. Those jobs belong to D. All the other jobs belong to U, which are the jobs that are not yet discovered and do not exist yet. Set the starting job location as the current node.
1) Complete the job at hand. Update the clock according to the service time of the job at hand. Now, $job \in C$, and D is updated as $D - \{job\}$
2) Check for any newly discovered jobs after the clock has been updated. If any new jobs pop up, add them to D and remove them from U.
3) If $D \neq \emptyset$, go to step 4-a, else if $D = \emptyset$ and $U \neq \emptyset$, go to step 4-b, else if $D = U = \emptyset$, go to step 5.

4-a) If any job in D has a priority of 1, then $next\_node$ is that node. Else: a cost is calculated for every job in D, and the $next\_node$ is determined in such a way that $next\_node = \min_{j \in D}\{t_{kj}\sqrt{(priority)_j}\}$, where t_{jk} is the time (as in travelling time) to go from the current job k's location to job j's location. The service times are not taken into account since that is

unknown information to the travelling salesman. Add t_{kj} minutes to the clock, then go to step 1.

4-b) Wait for the first job to pop up. When a job pops up, update the clock to that job's creation time and set its location to $next\_node$. Then, go to step 1.

5) Record clock after all the jobs are exhausted (i.e.: $|C| = n$). Compare with current system.

Notes:

- While calculating the "costs" for the simulated system, a PFD allowance of 0.8 was given, which is the average utilization rate of GDZ's teams.
- The times are calculated by extracting the distances using Google Maps API and dividing the distances by the average speed of the team on that specific day. While calculating the costs of the current system, due to the absence of data regarding which routes the team took to reach a certain destination, the Google Maps API distances were used.
- The priority given by GDZ is given as 1 to 10, were 1 is most important and 10 is least important. To adjust these values for the minimization problem at hand, the new priorities were adjusted as $11 - priority$, so that the job with a previous priority of 1 is now 10, and vice-versa.
- The costs were calculated in terms of the square root of priority as opposed to priority itself. The reason is because, in the current system, the priority system is a ranking system to rank the importance of the jobs and not a weighting system. Meaning, that a job or priority 1 (highest priority) is not necessarily twice as important as a job of priority 2, and not 10 times more important than a job of priority 10. So, the weight is determined to be $w_j = \sqrt{(priority)_j}$. In the future, a better weight may be used.

Results

Upon running the algorithm on a sample of 173 shifts, 109 of the simulated runs were, at worst, 1% more costly or less costly than the actual runs. In addition, 117 of the simulated runs were faster than the actual runs. On average, the improvements reduced the run-time by 33 minutes.