AI-Based Optimization of Urban Transport Routes in Azerbaijan

*Student: Murad Babayev, Group 2451i, NAA*

*mb.official.444@gmail.com*

*Supervisor: Tabriz Osmanli, NAA*

*tosmanli@naa.edu.az*

**Introduction and Problem Description.** In Baku, and other cities across Azerbaijan, moving around has become more of a challenge than it used to be. Population is rising, roads don’t always follow modern designs, and traffic changes from day to day. Many transport systems still rely on fixed routes — no matter what’s happening on the roads in real time.

So, I started to think what can help to solve target problems and end up with AI: could artificial intelligence help? Maybe there's a way to let systems respond to traffic conditions instead of always sticking to one plan. In this small study, I tested a few well-known algorithms — not just as theory, but in the context of how traffic behaves on a real route in Baku.

**Materials and Methods.** First, I asked myself — what really makes a route “bad”? Is it the distance? Or is it delays, jams, and unexpected problems? I picked a familiar route: from Qara Qarayev (a crowded district) to the National Aviation Academy near the airport. It’s about 16–17 km, and I know the typical traffic issues there from experience.

I broke the route into sections with different traffic weights and applied four algorithms to compare results. I kept things simple — just basic outcome tracking. The methods were:

* Dijkstra’s Algorithm, to find the shortest path.
* A\*, which adds a prediction of how far is left.
* Genetic Algorithm, that keeps testing new paths.
* Reinforcement Learning, which improves over time by learning.

Each algorithm was tested on the same route, using traffic levels I estimated based on how things usually are — not perfect, but pretty close to real conditions.

**Results and Discussion.** I tested four different algorithms to see how they’d behave on the same city route. First was Dijkstra. It gave me the shortest distance every time — 16.3 km — but the trip still took around 28 minutes, no matter what. That made sense, though — it doesn’t really care about traffic, just distance.

The next test was A\*. A\* adds a heuristic element that enables it to predict and steer clear of specific busy locations. It wasn’t a big change, but it showed that even a simple prediction can help in traffic.

Then I tried out the Genetic Algorithm, which was more experimental. This method considered variations of paths for some time until it found a path with approximately the same length of 16 km, but the time this time was significantly less, namely 22 minutes. This demonstrated that experimenting and making minor adjustments can lead to better outcomes.

The last methods that I used was Reinforcement Learning. It took longer to show results. At first, it was no better than the others. But after some runs, it started skipping known traffic zones and landed on a 15.5 km path that took just 18 minutes. That result stood out — not just for speed, but because it seemed like the model learned from the earlier runs.

RL is strong, it is not initially quick. To get better, more data and time are required. It gets better with time, not immediately.

Considering all results, we can get that, all the algorithms had their own strengths. RL gave the best results, yeah, but it also needed more time and setup. Honestly, mixing them might be the smarter move — like using Dijkstra or A\* to get quick answers, and letting RL do its learning in the background. In a place like Baku, where traffic is kind of all over the place, that mix could actually make things work better.