This project is about using the domain knowledge to improve the quality of the results obtained by nature-inspired algorithms working on complex optimization problems such as the transport optimization problems. Let's explain the three main ideas behind this project.

Transportation services market is very competitive. To survive on this market transportation companies have to solve many complex optimization problems, for example they want to ensure that, when delivering goods, they can follow a possibly short route. But, how many possible routes there are to choose from? Unfortunately, there is an overwhelming number of possibilities. If we would like to protect just 20 locations one by one in a sequence there are 2 432 902 008 176 640 000 ways of doing it. It seems that the situation is hopeless even if we have a powerful computer at our disposal. But wait, do we have to check all the 2 432 902 008 176 640 000 possibilities just to find a reasonably good tour to visit 20 locations? Fortunately not!

If we agree that our solution does not have to be the best, but just reasonably close to the optimum, we can find a satisfactory solution relatively quickly. We will employ a range of techniques known under a common name of population-based optimization. This idea is probably the easiest to explain using evolutionary algorithms as an example. Imagine, that we have to find the highest point in a mountain range that we do not have a map for. Achieving this task on our own could take a lifetime if the area is vast. Therefore, let's organize a search team! Each member of the team will search a part of the mountain range and each will have a GPS device to measure at what altitude he or she is. So far, our approach does not seem particularly effective. We have more people at our disposal, but many members of the team will probably explore lower areas of the mountain range without any chance of finding the peak. That's true because what we have at this point is more or less a random search. So, let's change the game a little. Assume, that each member of the team will have a random chance, from time to time, to call and relocate one of the colleagues to a location nearby. To make the entire scheme work we need only one more thing: those who have climbed higher have to be given the opportunity to relocate someone placed at a lower location more often than vice versa. See how it works? Because the searchers located higher up call their colleagues more often the entire team slowly moves upwards. "Downward calls" sometimes happen, but they are not very likely. On modern machines evolutionary algorithms can find solutions to huge optimization problems, such as finding a possibly short route among 200 cities - the problem in which more than 10 to the power of 370 (one and 370 zeros!) different routes are possible. Of course, the obtained solutions are not perfect, but getting the solution quickly usually outweighs the fact that the proposed route is a few percent longer than the optimal.

While working on an optimization tool for a bus company that provided transport services for disabled children travelling between homes and schools we have identified at least three optimization criteria (bus operation cost, the distance travelled and the number of buses used at the same time) and a number of constraints (the capacity of buses, time windows when a child can leave home and when he/she must be at school, maximum travel time, bus driver working time constraints and some others). Also, there are less formal expectations, for example, carrying a passenger through or "near" the same location more than once should be avoided. Violating this requirement is perceived as "driving the passenger back-and-forth without need" and often causes concerns and protests. While working with specialists from the bus company, we learnt that they have many good practices that allow them to construct good bus routes. Can a computer find better solutions? Well, it can, but with a large effort and satisfying less formal expectations (for example "find not too complicated routes") is difficult.

This is where the idea of this project comes from. Can we combine expert knowledge with the computing power of modern computers? We think we can and we have some ideas how to do it. As a result we will have transportation services that are not only cheaper and faster, but also organized in a way that is more appealing to the end-users. Actually, the ideas we have, are more general, so they are applicable not just to transport optimization problems, but also to many other areas.