\* ALGAV Sprint C Report

**Deliveries Planning using Electrical Trucks.**

3DE

Minju Lee(1220450), Christina Austestad Hardeland(1220385)

**Index**

[1. Introduction 3](#_Toc124085855)

[2. Creation of the initial population of the genetic algorithm (GA) 3](#_Toc124085856)

[3. Random Crossover 4](#_Toc124085857)

[4. Selection of the new generation of the population 5](#_Toc124085858)

[5. Efficacy Analysis 6](#_Toc124085859)

[6. Parametrization of the ending condition of the AG 6](#_Toc124085860)

[7. Several trucks 7](#_Toc124085861)

[8. Study of methods 7](#_Toc124085862)

[1. Introduction 7](#_Toc124085863)

[2. Predictive modelling for demand forecasting 7](#_Toc124085864)

[3. Optimization algorithms for routing and scheduling 8](#_Toc124085865)

[4. Reinforcement learning for inventory management 9](#_Toc124085866)

[5. Conclusion 9](#_Toc124085867)

[9. Conclusions 10](#_Toc124085868)

## Introduction

The goals for Sprint were to understand, implement our problem and improve the given genetic algorithm. To solve this, we divided the tasks and worked on the given tasks.

## Creation of the initial population of the genetic algorithm (GA)

Creation of the initial population of the genetic algorithm (GA) considering two individuals (sequence of the deliveries) generated by different heuristics able to create good solutions according to the previous sprint. If the generated individuals are the same, you can transform one of them with a mutation (e.g. exchange two consecutive genes). Other elements from the population can be randomly generated. Notice that repeated individuals in the population must be avoided.

텍스트이(가) 표시된 사진

자동 생성된 설명Adapt the Genetic Algorithm to one of the heuristics problems, the next warehouse to be visited is the one not visited more closer of for which we arrive more rapidly. Permutation of lower time process is part of initial population.

Figure 1: generate\_population(Pop) and generate\_population/4.

The upper picture is showing that we’re using heuristics process in code and generate\_population/4 generates all the generation as much as user wants. If the user wants to generate 6 generations, then it starts from 0 to 5.

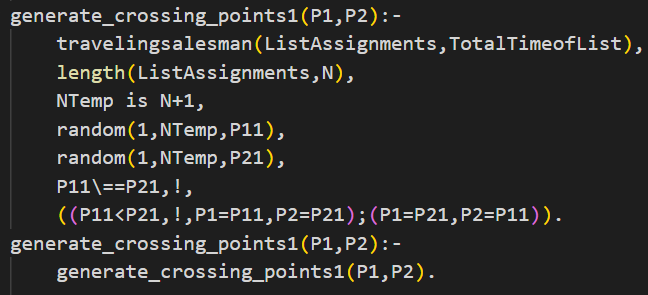


Figure 2: generate\_crossing\_points1(P1, P2).

Crossover is attempted on successive individuals 2 to 2 of the population, which may be a limitation. To find out if the crossing takes place, a random number between 0 and 1 is generated and compared with the parameterized crossing probability, if it is lower, the crossing is made.

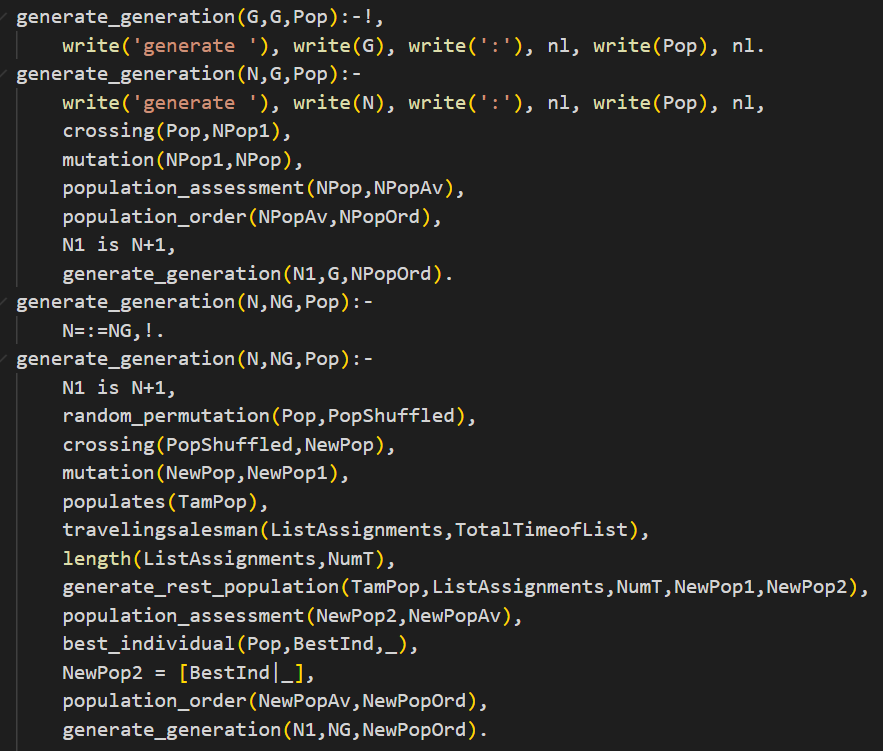


Figure 3: generate\_generation/3.

Finally, including lower time heuristics solution in the initial populations, we repeated mutation and crossover as passing generations. As results, we can get the better delivery plan than previous generation, selecting the best two solution in previous generation. The next generation’s delivery plans are made randomly by two best solutions in the previous one.

## Random Crossover

A problem with the given genetic algorithm is that there is no random crossover between individuals in the population avoiding the sequence of crossovers is always between the first and second, third and fourth, and so on.

To solve this problem, the predicate random\_permutation/2 from SWI Prolog is used. It performs a random shuffle of the list of chromosomes before performing a crossover. The code is added in the generate\_generation predicate before the crossing takes place, as shown in Figure 1.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Figure 4: random\_premutation/2 added to the code.

## Selection of the new generation of the population

In the given genetic algorithm, the current population is being replaced, pair by pair, and there is therefore no guarantee that the best individual will pass to the next generation.

One way to change the code such that the two better individuals of all elements go to the next generation, is to join the top performing individuals from the current generation with their descendants in the next generation. This can be done by modifying the generate\_generation/3 predicate to consider the elements of the previous generation and their descendants after crossover and mutation when selecting the top-performing individuals for the next generation.

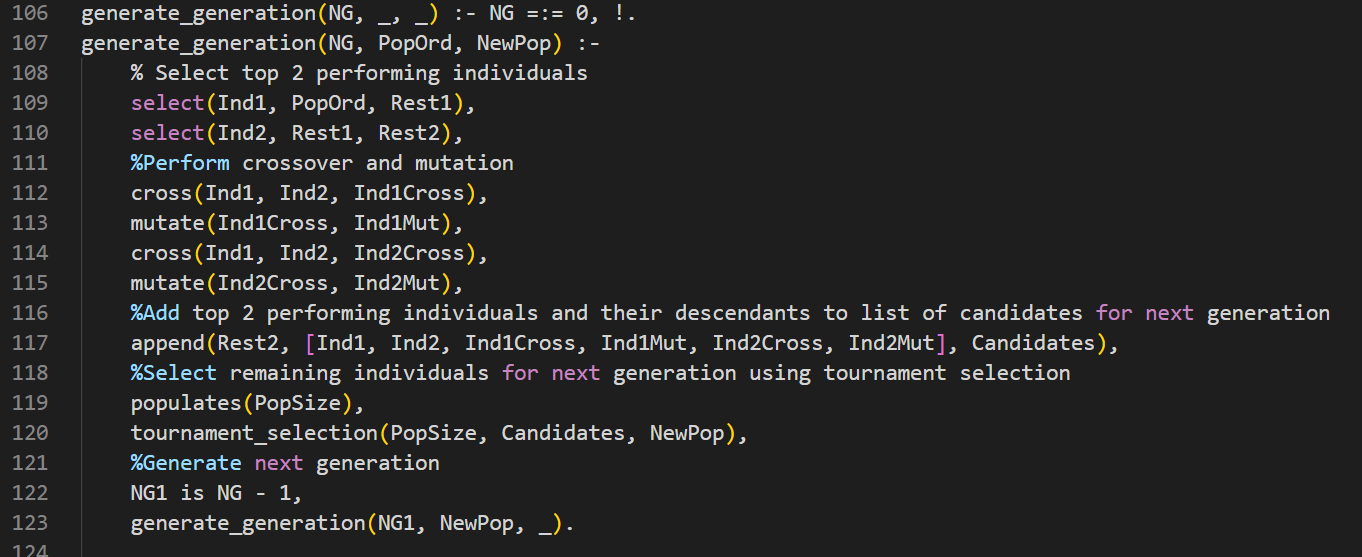
The generate\_generation/3 predicate, as shown in Figure 2, generates a new generation of individuals by selecting the top two performing individuals from the current generation, then performing crossover and mutation on them to produce their descendants, and finally selecting the remaining individuals for the next generation using tournament selection. It does this by calling the tournament\_selection/3 predicate with the desired population size and a list of candidates which includes the top two performing individuals and their descendants.

Figure 5: edited generate\_generation/3 for a better selection process.

Et bilde som inneholder tekst

Automatisk generert beskrivelseThe tournament\_selection/3 predicate selects the remaining individuals for the next generation using tournament selection. It does this by recursively calling itself with a reduced number of individuals to select and a reduced list of candidates until the desired number of individuals has been selected. Each recursive call selects two candidates at random using the random\_member/2 predicate and then selects the best candidate based on evaluation using the tournament/2 predicate. The tournament/2 predicate compares the evaluations of the two candidates and returns the one with the better evaluation.

Figure 6: tournament\_selection/3 and tournament/2.

This will ensure that the top-performing individuals from the current generation are always included in the next generation while still allowing for some diversity by including the descendants of other individuals as well.

## Efficacy Analysis

All of the results in the table are containing initial population from one of the heuristics problems. We adapted only lower time process to Genetic Algorithm. As we can see in the table below, time for deliveries for the better solution obtained by the modified GA is better than the optimal solution from Sprint B before over 9 deliveries. As getting increased the number of deliveries, we can see that run count of generating generation is getting important because the time gap between previous and last generation is getting larger. In conclusion, as generations pass, the process produces better result than previous one.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of deliveries | Time for deliveries for the optimal solution from sprint B | Time for deliveries for the better solution obtained by the modified GA | Time for deliveries for the better solution obtained by the original GA | Medium value of the time for deliveries for the last generation of the modified GA | The medium value of the time for deliveries for the previous generation of the original GA |
| 6 | 539 | 532 | 542 | 547 | 548 |
| 7 | 556 | 517 | 553 | 582 | 713 |
| 8 | 581 | 525 | 693 | 729 | 751 |
| 9 | 592 | 717 | 738 | 777 | 832 |
| 10 | 622 | 667 | 829 | 915 | 956 |

## Parametrization of the ending condition of the AG

According to two conditions: number of generations and obtaining NA individual with evaluation lower or equal to a certain value or stabilization of the population, we adapt parametrization of the ending condition of the Genetic Algorithm.

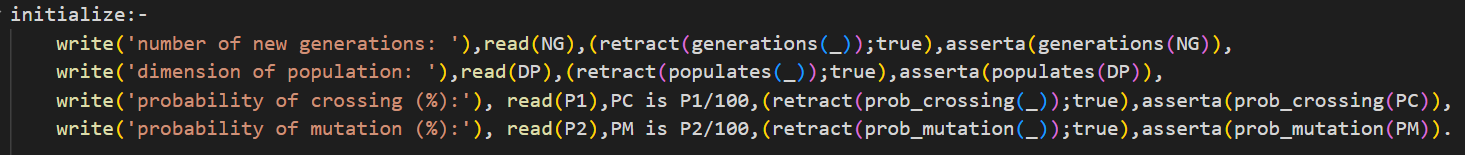


Figure 7: initialize and read the value from the user.

To create multiple generations, we get the value from user as much as user wants. The number of generations start from 0, 1, 2, and can be any number as user wants. Dimension of population is the number of children. The initial population is heuristics solution, and it is intended that the best, or some of the best individuals go to the next generation then just joint in the same list the elements of the previous population with their descendants. Each value, probability of crossing is being 50% and probability of mutation is being 25% in our problem. Therefore, we generate generations, selecting best two individuals from previous generations.

## Several trucks

Use of the GA to handle several trucks, representing in the same chromosome the deliveries of the several trucks. At least two elements of the initial population must be generated by applying a balancing method (number of deliveries and loads, the number of deliveries, and geographical aspects). Avoid unfeasible solutions (with more load than the truck capacity).

As we were only two students working on this project, we unfortunately did not have the time to finish this part of the project. We instead focused on the other parts of this sprint.

## Study of methods

A study of methods of Machine Learning applied to the problem of goods distribution.

### 1. Introduction

Goods distribution refers to the movement of goods from the point of production to the point of consumption, and it is a critical aspect of supply chain management. Proper planning is essential for ensuring that goods are delivered efficiently and effectively, and machine learning can be a powerful tool for optimizing various aspects of goods distribution planning.

Machine learning is a subset of artificial intelligence that involves the use of algorithms and statistical models to allow computer systems to "learn" from data without being explicitly programmed. It has the potential to greatly improve the efficiency and effectiveness of goods distribution by automating tasks and making more accurate predictions.

In this study, we will explore the various ways in which machine learning can be applied to the problem of goods distribution planning. We will examine the use of predictive modelling for demand forecasting, optimization algorithms for routing and scheduling, and reinforcement learning for inventory management. Through case studies and examples, we will demonstrate the potential of machine learning to improve the planning of goods distribution and discuss the future outlook for the use of these technologies.

### 2. Predictive modelling for demand forecasting

Predictive modelling is a key area of machine learning that involves the use of algorithms and statistical models to make predictions about future outcomes based on past data. In the context of goods distribution, demand forecasting refers to the process of predicting future demand for certain products in different locations. This is important because it helps logistics companies plan their routes and optimize their inventory levels.

There are various machine learning algorithms that can be used for demand forecasting in goods distribution. One common approach is to use decision trees, which involve building a tree-like model of decisions and their possible consequences (Quinlan, 1986). This can help a company predict demand for a particular product in a specific region based on past sales data, weather patterns, and other relevant factors.

Random forests are another popular method for demand forecasting in goods distribution. They involve creating an ensemble of decision trees and aggregating the predictions made by each tree. This can help improve the accuracy of the predictions by reducing the variance and bias that may be present in a single decision tree.

Neural networks are a more advanced type of machine learning algorithm that can also be used for demand forecasting in goods distribution. They involve creating a network of interconnected nodes that can process and analyze data in a way that mimics the structure and function of the human brain. Neural networks are particularly useful for handling large, complex datasets and can be trained to make accurate predictions even in the presence of noise or missing data(Jung & Kim, 2016).

There have been many successful applications of machine learning for demand forecasting in goods distribution. For example, a company might use a machine learning model to predict demand for a particular product in a specific region based on past sales data, weather patterns, and other relevant factors. This can help the company better plan its routes and optimize its inventory levels, leading to cost savings and improved customer satisfaction.

### 3. Optimization algorithms for routing and scheduling

Optimization algorithms are a key tool in machine learning that can be used to solve complex optimization problems. In the context of goods distribution, routing and scheduling refer to the process of determining the most efficient routes and schedules for delivery trucks.

There are various machine learning algorithms that can be used for routing and scheduling in goods distribution. One common approach is to use genetic algorithms, which involve simulating the process of natural evolution in order to find the optimal solution to a problem. This can be particularly useful for optimizing routing and scheduling in goods distribution because it allows for the consideration of multiple variables and constraints.

Simulated annealing is a machine-learning algorithm that can be used for routing and scheduling in goods distribution. It involves finding the optimal solution to a problem by iteratively making small changes to the current solution and evaluating the results. This process is inspired by the physical process of annealing, in which a material is slowly cooled in order to reduce defects and increase its structural purity. Simulated annealing can be useful for finding good, but not necessarily optimal, solutions in cases where the optimization problem is complex or the search space is large. This is because it allows the algorithm to "explore" the search space by making random changes to the current solution and accepting those that result in an improvement (Delahaye, Chaimatanan & Mongeau, 2019). This can help the algorithm find good solutions that might have been missed by a more deterministic optimization algorithm. However, because simulated annealing relies on randomness, it is not guaranteed to find the absolute optimal solution.

Ant colony optimization (ACO) is a metaheuristic algorithm that can be used to find the shortest delivery routes between two points in the field of goods distribution. ACO algorithms work by simulating the behaviour of a colony of ants as they search for food, with each delivery truck treated as an "ant" that leaves a "pheromone trail" behind it as it follows a particular path. The pheromone trails are updated based on the rewards received for following a particular path, with the intensity of the trails decreasing over time. ACO algorithms have several key components, including the pheromone trail update rule, the transition rule, and the objective function, as well as several adjustable parameters, such as the evaporation rate and the exploration rate(Dorigo & Stützle, (2003).

There have been many successful applications of machine learning for routing and scheduling in goods distribution. For example, a company might use a machine learning algorithm to determine the most efficient routes for its delivery trucks, taking into account factors such as traffic patterns, distance, and delivery time windows. Similarly, a company might use machine learning to schedule deliveries and pickups in a way that maximizes efficiency and minimizes costs.

### 4. Reinforcement learning for inventory management

Reinforcement learning is a type of machine learning that involves an agent learning to make decisions in an environment to maximize a reward. In the context of goods distribution, inventory management refers to the process of ensuring that a company has the right amount of inventory in the right place at the right time. This is important because it can help a company avoid running out of stock or having too much excess inventory, which can lead to unnecessary costs.

There are various machine learning algorithms that can be used for inventory management in goods distribution. One common approach is to use Markov decision processes (MDPs), which involve modelling the interactions between an agent and its environment as a sequence of states, actions, and rewards (van Otterlo & Wiering, 2012). This can help a company predict how much inventory it needs to have on hand in different locations at different times based on past sales data and other relevant factors.

Dynamic programming is another method that can be used for inventory management in goods distribution. It involves breaking down a complex optimization problem into smaller, simpler subproblems and solving them in a recursive manner. This can be useful for finding the optimal inventory levels for a company, taking into account factors such as lead times, holding costs, and ordering costs.

Monte Carlo simulation is a machine learning algorithm that can also be used for inventory management in goods distribution. It involves using random sampling and statistical analysis to estimate the probability of different outcomes (Barto & Duff, 1993). This can be useful for predicting the likelihood of different inventory scenarios and helping a company make informed decisions about its inventory levels.

There have been many successful applications of machine learning for inventory management in goods distribution. For example, a company might use a machine learning model to predict how much inventory it needs to have on hand in different locations at different times based on past sales data and other relevant factors. This can help the company avoid running out of stock or having too much excess inventory, leading to cost savings and improved customer satisfaction.

### 5. Conclusion

In this study, we explored the various ways in which machine learning can be applied to the problem of goods distribution planning. We examined the use of predictive modelling for demand forecasting, optimization algorithms for routing and scheduling, and reinforcement learning for inventory management. Through case studies and examples, we demonstrated the potential of machine learning to improve the planning of goods distribution and discussed the future outlook for the use of these technologies.

Overall, machine learning has the potential to greatly improve the efficiency and effectiveness of goods distribution by automating tasks and making more accurate predictions. By predicting demand, optimizing routes and schedules, and managing inventory, machine learning can help logistics companies save time, money, and resources.

Looking ahead, it is likely that machine learning will continue to play an increasingly important role in the field of goods distribution. As data becomes more widely available and machine learning algorithms continue to advance, it is likely that we will see even more innovative applications of these technologies in the field.

## Conclusions

**References:**

Tarallo, E., Akabane, G. K., Shimabukuro, C. I., Mello, J., & Amancio, D. (2019). Machine Learning in Predicting Demand for Fast-Moving Consumer Goods: An Exploratory Research. *IFAC-PapersOnLine*, *52*(13), 737-742. <https://doi.org/10.1016/j.ifacol.2019.11.203>

Gayialis, S. P., & Tatsiopoulos, I. P. (2004). Design of an IT-driven decision support system for vehicle routing and scheduling. *European Journal of Operational Research*, *152*(2), 382-398. <https://doi.org/10.1016/S0377-2217(03)00031-6>

Fujimaki R., Muraoka Y., Ito S., Yabe A. (2016). From prediction to decision making - Predictive optimization technology. NEC Technical Journal.

Quinlan, J.R. (1986). Induction of decision trees. Mach Learn 1, 81–106. <https://doi.org/10.1007/BF00116251>

Breiman, L. (2001). Random Forests. Machine Learning 45, 5–32. <https://doi.org/10.1023/A:1010933404324>

Jung, S., & Kim, T. (2016). New approach for the diagnosis of extractions with neural network machine learning. American Journal of Orthodontics and Dentofacial Orthopedics, 149(1), 127-133. <https://doi.org/10.1016/j.ajodo.2015.07.030>

Koksal, E., Hegde, A. R., Pandiarajan, H. P., & Veeravalli, B. (2021). Performance characterization of reinforcement learning-enabled evolutionary algorithms for integrated school bus routing and scheduling problem. International Journal of Cognitive Computing in Engineering, 2, 47-56. <https://doi.org/10.1016/j.ijcce.2021.02.001>

Masegosa, A. D., Oliva, D., & Onieva, E. (2022). A new Hyper-heuristic based on Adaptive Simulated Annealing and Reinforcement Learning for the Capacitated Electric Vehicle Routing Problem. *arXiv*. <https://doi.org/10.48550/arXiv.2206.03185>

Delahaye, D., Chaimatanan, S., Mongeau, M. (2019). Simulated Annealing: From Basics to Applications. In: Gendreau, M., Potvin, JY. (eds) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol 272. Springer, Cham. <https://doi.org/10.1007/978-3-319-91086-4_1>

Dorigo, M., Stützle, T. (2003). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In: Glover, F., Kochenberger, G.A. (eds) Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol 57. Springer, Boston, MA. <https://doi.org/10.1007/0-306-48056-5_9>

van Otterlo, M., Wiering, M. (2012). Reinforcement Learning and Markov Decision Processes. In: Wiering, M., van Otterlo, M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization, vol 12. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-27645-3_1>

Barto, A., Duff, M. (1993). Advances in Neural Information Processing Systems: Monte Carlo Matrix Inversion and Reinforcement Learning. Morgan-Kaufmann. <https://proceedings.neurips.cc/paper/1993/file/3b3dbaf68507998acd6a5a5254ab2d76-Paper.pdf>