

# Group 24 Assignment Report

**An Evolutionary Algorithm for an elevator to deliver as many people as possible to their desired floor**

Oskar Sundberg      Linus Savinainen  
Samuel Wallander Leyonberg      Gustav Pråmell  
Joel Scarinius Stävmo

September 23, 2024

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Optimize elevator routing</b>	<b>3</b>
2.1	Mathematical formulation . . . . .	4
2.2	Similar published academic work . . . . .	4
2.1	Why this evolutionary approach . . . . .	5
<b>3</b>	<b>Algorithm</b>	<b>5</b>
<b>4</b>	<b>Experimental part</b>	<b>6</b>
<b>5</b>	<b>Results and Analysis</b>	<b>7</b>
<b>6</b>	<b>Conclusions</b>	<b>8</b>

# 1 Introduction

The recommended software for typesetting assignment reports is L<sup>A</sup>T<sub>E</sub>X. It will allow you to prepare high-quality documents, especially in the area of Computer Science. This document can serve as a template for reports. Each section begins with brief instructions in red text. All the instructions in red, as well as the dummy text, should be removed in the final version to submit. The L<sup>A</sup>T<sub>E</sub>X source of this file includes examples of using the most needed commands and environments. You can find plenty of other examples with explanations in many web forums and discussion groups on the Internet. The easiest way to edit your report is to use <https://www.overleaf.com/>. Overleaf does not require any setup on your computer, and it is free to create an account.

The book *Writing for Computer Science* [1] is a useful assistance on how to write properly and present your work when it comes to Computer Science topics. It is a strong recommendation to follow its guidelines and limit the usage of AI tools to generate text. Keep in mind that the examiner is an expert in Evolutionary Computation and therefore, any false information generated by an AI tool is easily notable. Such case may lead to failing the assignment.

The introduction should briefly introduce the assignment and its purpose.

This report addresses the challenge of optimizing an elevators route to improve the efficiency of picking up and delivering passengers to their desired floors. In buildings with many floors, efficient elevators are crucial for convenient travel between floors. Poorly designed elevator systems will lead to long waiting time, unnecessary stops and unsatisfied users. Elevator technology have made progress during the years, but many elevators still struggle with finding an efficient way to serve passengers.

The hypotheses that evolutionary algorithms will be able to find near-optimal routes for elevators aiming to maximize the number of passengers served while minimizing travel distance and/or travel time is the core focus in the report. Different strategies and hyperparameters are experimented with to enable demonstration of how evolutionary algorithms can in the long run satisfy users, lower cost for reparation and save energy, due to more efficient use of elevators.

## 2 Optimize elevator routing

The second section should present the problem you tackle using your evolutionary approach. Overall, this section should include:

- The mathematical formulation considered in your study. Some problems have a clear mathematical model (e.g., Travelling Salesman Problem), while others do not (e.g.,  $n$ -Queens). Based on the problem you chose, search the literature and find a proper way to present the problem.
- One paragraph that briefly presents at least 3 published academic works where any evolutionary approach is used to solve the problem. It would be wise to cite here works that influenced your algorithm. This practice saves you time from looking for additional academic resources. You can find more information about reading and searching in the literature in [2].
- The motivation behind the evolutionary approach you decided to develop. A good practice would be to align the motivation with some literature gap found in the academic works you presented above. However, this is not mandatory. You can motivate your selection on the characteristics of the algorithm making it proper for the problem.

**Note:** Change the section's title to match the name of the problem you chose for your assignment.

## 2.1 Mathematical formulation

## 2.2 Similar published academic work

[3], "Each vertex of a graph initially may contain an object of a known type. A final state, specifying the type of object desired at each vertex, is also given. A single vehicle of unit capacity is available for shipping objects among the vertices. The swapping problem is to compute a shortest route such that a vehicle can accomplish the rearrangement of the objects while following this route."

## Mapping the elevator problem to the swapping problem.

- Vertices are represented as floors in the elevator.
- The object is defined as a person.
- Initial state consists of  $N$  persons waiting at the floors.
- The final state is reached when all persons are at their destination floor.

- The elevator is a vehicle with a max capacity that moves the persons between floors, i.e. a single vehicle shipping objects among the vertices.
- The elevator problem is aiming to reduce the waiting time of the persons in the elevator and thereby reduce the distance traveled.

According to [3], "Even the simple swapping problem is NP-hard."

## 2.1 Why this evolutionary approach

## 3 Algorithm

The third section should present the evolutionary approach you developed. You can divide this section into subsection. In any case, you should mention the following details:

**Evolutionary approach.** Clearly describe the algorithm you developed. You should clearly explain the evolutionary operators you used and what modifications you did to match the problem. It is extremely important to present also a pseudocode of your algorithm. An example is given in 1, below. For more insight into presentation of algorithms, you can advise [4].

To typeset pseudocode in L<sup>A</sup>T<sub>E</sub>X you can use one of the following options:

- Choose ONE of the (`algpseudocode` OR `algcompatible` OR `algorithmic`) packages to typeset algorithm bodies, and the `algorithm` package for captioning the algorithm.
- The `algorithm2e` package.

You can find more information here: <https://www.overleaf.com/learn/latex/Algorithms>

**Solution representation.** Clearly describe the solution representation you used. You can use figures to improve the comprehensibility of this part.

**Fitness function.** It is also very important to mention the fitness function you used. In many cases, the objective function of the problem is not the same as the fitness function used in an evolutionary algorithm. An example, following the principles of [5], is given below.

$$F = \sum_{i=1}^d x_i^2 \quad (1)$$

---

**Algorithm 1** Example of an algorithm's pseudocode

---

**Require:**  $n \geq 0$ **Ensure:**  $y = x^n$  $y \leftarrow 1$  $X \leftarrow x$  $N \leftarrow n$ **while**  $N \neq 0$  **do**    **if**  $N$  is even **then**         $X \leftarrow X \times X$          $N \leftarrow \frac{N}{2}$ 

▷ This is a comment

**else if**  $N$  is odd **then**         $y \leftarrow y \times X$          $N \leftarrow N - 1$     **end if****end while**

---

where  $x_i$  is the  $i$ -th gene (i.e., decision variable) in the solution and  $d$  corresponds to the number of decision variables in the problem.

**Note:** Change the section's title to match the name of the algorithm you developed for your assignment.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

## 4 Experimental part

This section describes the setup of experiments [6]:

- Provide the details of the hardware and software that you used.
- Describe the steps you carried out during your experiments.
- Detail the data you used for the evaluation of your algorithm.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

## 5 Results and Analysis

This section should present the obtained results and provide an insightful analysis of them. You can present the results using graphs, tables, or any other visualization method suits your purpose. Do not forget to include proper captions [7] in any of these illustration methods you use. You do not need to provide any execution details as they are already presented in Sec. 4.

A good practise would be to compare your algorithm with a simpler approach, such as (a) a naive method, (b) a Hill Climbing approach, or (c) a simple evolutionary algorithm. In the third case, you can use the simpler version of the algorithm you developed, i.e., the original algorithm without your modifications. In that case, you should briefly describe the comparing method(s) in Sec. 4. Alternatively, you can use some reference results derived from the repositories you found some benchmark instances.

To display tables, the `booktabs` package might be useful. For example, Table 1 shows how you should increase the size of  $n$ , when running your code. You can advice [7] to see a few examples of proper tables.

Table 1: Example of comparison the developed algorithm's results with the best ones from a repository.

Instance	Optimum (Repository xyz)	EA	time (s)
st70	678.597	677.109	0.67
ei176	545.387	544.369	1.16
kroA100	21285.443	21285.443	1.69
rd100	7910.396	7910.396	2.14
Pr136	96772	96770.924	7.11
Pr144	58537	58535.221	7.97
a280	2856.769	2856.769	33.47

You can use different illustration methods to present different aspects of your analysis. Figure 1 gives an example using the `pgfplots` package.

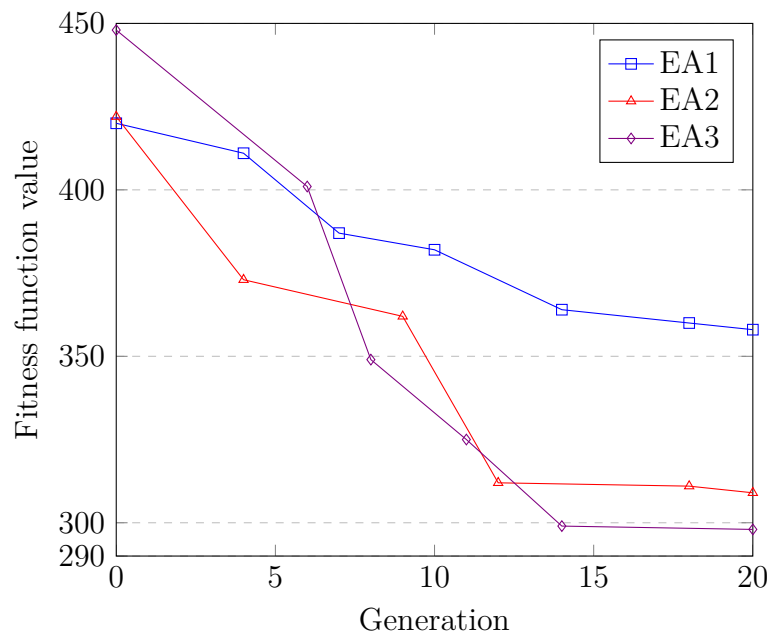


Figure 1: Example of convergence analysis.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

## 6 Conclusions

In this section you should provide a concise summary of what has been done, the obtained results and some recommendations on how this study could be extended.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.



## References

- [1] J. Zobel, *Writing for computer science*. Springer, 2014.
- [2] J. Zobel, “Reading and reviewing,” in *Writing for Computer Science*, pp. 19–33, Springer, 2014. [https://doi.org/10.1007/978-1-4471-6639-9\\_3](https://doi.org/10.1007/978-1-4471-6639-9_3).
- [3] S. Anily and R. Hassin, “The swapping problem,” *Networks*, vol. 22, no. 4, pp. 419–433, 1992.
- [4] J. Zobel, “Algorithms,” in *Writing for Computer Science*, pp. 115–128, Springer, 2014. [https://doi.org/10.1007/978-1-4471-6639-9\\_10](https://doi.org/10.1007/978-1-4471-6639-9_10).
- [5] J. Zobel, “Mathematics,” in *Writing for Computer Science*, pp. 131–143, Springer, 2014. [https://doi.org/10.1007/978-1-4471-6639-9\\_9](https://doi.org/10.1007/978-1-4471-6639-9_9).
- [6] J. Zobel, “Experimentation,” in *Writing for Computer Science*, pp. 197—215, Springer, 2014. [https://doi.org/10.1007/978-1-4471-6639-9\\_14](https://doi.org/10.1007/978-1-4471-6639-9_14).
- [7] J. Zobel, “Graphs, figures, and tables,” in *Writing for Computer Science*, pp. 83–113, Springer, 2014. [https://doi.org/10.1007/978-1-4471-6639-9\\_11](https://doi.org/10.1007/978-1-4471-6639-9_11).