CHALMERS GÖTEBORGS UNIVERSITET

Algorithms for Scheduling Teaching Assistants

A comparison focused on preference satisfaction, workload fairness, and computational performance

Bachelor's thesis in Software Engineering

Chloe Dusén

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG

Gothenburg, Sweden 2026
www.chalmers.se

Bachelor's thesis 2026

Algorithms for scheduling Teaching Assistants

A comparison focused on preference satisfaction, workload fairness, and computational performance

CHLOE DUSÉN

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Algorithms for scheduling Teaching Assistants
A comparison focused on preference satisfaction, workload fairness, and computational performance
CHLOE DUSÉN

Supervisor: Niklas Broberg, Computer Science and Engineering
Co-examiner: Staffan Björk, Computer Science and Engineering
Examiner: Patrik Jansson, Computer Science and Engineering

Bachelor's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2026

Algorithms for scheduling Teaching Assistants
A comparison focused on preference satisfaction, workload fairness, and computational performance
CHLOE DUSÉN
Chalmers University of Technology and University of Gothenburg

Abstract

Efficient scheduling of Teaching Assistants is a complex combinatorial optimization problem faced by many universities, involving constraints such as course requirements, availability, working hours, and fairness. A survey was conducted among Teaching Assistants at the Department of Computer Science and Engineering at Chalmers University of Technology and the University of Gothenburg to determine schedule preferences and system requirements. These preferences and requirements were used to create soft constraints and hard constraints for the scheduling algorithms.

This thesis investigates four algorithms: a simple greedy algorithm; a simple genetic algorithm; Tabu Search; and Simulated Annealing. Results reveal that although Greedy Algorithm was the quickest, it performed poorly on constraint satisfaction. Both Tabu Search and Simulated Annealing found high-quality solutions, but since Tabu Search was more computationally efficient, it was concluded to be the superior algorithm.

Keywords: scheduling, Tabu Search, Genetic Algorithm, Simulated Annealing, Greedy Algorithm, Teaching Assistant.

Algorithms for scheduling Teaching Assistants
A comparison focused on preference satisfaction, workload fairness, and computational performance
CHLOE DUSÉN
Chalmers University of Technology and University of Gothenburg

Sammandrag

Effektiv schemaläggning av amanuenser är ett komplext kombinatoriskt optimeringsproblem som många universitet står inför, med begränsningar som kurs behov, tillgänglighet, arbetstider, och rättvisa. En undersökning genomfördes bland amanuenser vid institutionen för Data- och Informationsteknik vid Chalmers tekniska högskola och Göteborgs Universitet för att fastställa systemkrav och schemaläggningspreferenser. Dessa preferenser och krav användes för att skapa flexibla riktlinjer och strikta vilkor för schemaläggningsalgoritmerna.

Denna avhandling undersöker fyra algoritmer: en enkel greedy algoritm; en enkel genetic algoritm; Tabu Search; och Simulated Annealing. Resultaten visar att även om greedy algoritmen var snabbast, presterade den dåligt på preferens och kravhantering. Både Tabu Search och Simulated Annealing fann lösningar av hög kvalitet, men eftersom Tabu Search var mer beräkningsmässigt effektiv drogs slutsatsen att det var den bättre algoritmen.

Keywords: Schemaläggning, Tabu Search, Genetic Algorithm, Simulated Annealing, Greedy Algorithm, Amanuens.

Acknowledgements

I want to extend my gratitude to my supervisor, Niklas Broberg. For his invaluable guidance, constructive feedback, and continued support throughout this project. I would also like to thank the Department of Computer Science and Engineering for their support and flexibility in making this project possible. Finally, I would like to thank the Teaching Assistants who participated in the survey as a part of this project.

Chloe Dusén, Gothenburg, May 2026

Contents

List of Figures

List of Tables

1 Introduction

Scheduling is a class of problems concerned with allocating limited resources over time under constraints, and many such problems are known to be computationally difficult. As a result, scheduling has been an active area of research within operations research, optimization, and computer science.

The Department of Computer Science and Engineering at Chalmers University of Technology and the University of Gothenburg (CSE) employs approximately 200 Teaching Assistants each year to staff its courses. According to Niklas Broberg, Assistant head of department for undergraduate and graduate education, the majority of Teaching Assistants are scheduled manually, which leads to large amounts of resources being spent on scheduling. This thesis aims to develop an automatic scheduling system that creates schedules using significantly less resources.

1.1 Purpose

The aim of this study is to compare and assess algorithms that could be used to improve the scheduling of Teaching Assistants at CSE. Currently all of the scheduling is done manually which is very time inefficient. The study aims to find an algorithm to efficiently create high-quality schedules according to Teaching Assistant preferences. The aim is that the results of this study combined with future work within interaction design would create a deployable product. This product would decrease the manual labour and hopefully allow for higher quality schedules across the department.

1.2 Societal and Ethical Aspects

There are certain ethical considerations when scheduling. The system must avoid introducing biases that could unfairly disadvantage individual Teaching Assistants. Factors such as availability and workload distribution must be handled equitably to ensure that every Teaching Assistant is treated fairly.

Additionally, since scheduling is an NP-hard problem one needs to be careful about energy consumption and efficiency. Large instances of the scheduling problem will inescapably require large amounts of resources, but it is still important to be mindful and minimize environmental impact. Therefore this thesis will not only compare solution quality but also algorithm efficiency.

1.3 Delimitations

To ensure the feasibility of the project within the given timeframe, multiple delimitations have been decided. To align with the research direction, the thesis focuses on a selection of algorithms that solve this version of the general scheduling problem. Therefore certain features of the system have been greatly simplified or removed. These include an input system, for Teaching Assistant availability and preferences, login, Canvas integration, server hosting, and a graphical representation of the finished schedules. The intended users of the system are employees at CSE. Therefore, the decision was made to focus solely on this department, to ensure that the system has a clear connection to stakeholder requirements and expectations.

2
Theory

Since scheduling is an NP-hard problem [1], its computational complexity grows rapidly with problem size, making exact solutions infeasible for large problems. As a result, modern scheduling research tends to focus on meta-heuristic and hyper-heuristic solutions [2], such as genetic algorithms (GA), Tabu Search (TS), and Simulated Annealing (SA) [3] [4]. Unlike mathematically exact solutions, meta-heuristic and hyper-heuristic solutions are non-deterministic and do not guarantee to locate the optimal solution.

Due to the non-exact nature of the above-mentioned algorithms, a comparison method is needed. When comparing the algorithms satisfaction of hard constraints, satisfaction of soft constraints, and computational efficiency need to be evaluated. This is typically done using an evaluation function that will assign a score to each schedule, based on hard and soft constraints. Hard constraints affect the feasibility of the solution and are therefore highly weighted in evaluation functions.

A solution where Teaching Assistant preferences are not respected may be feasible but it is not optimal; this is translated through soft constraints. Soft-constraint violations do not make a solution infeasible, just sub-optimal; therefore, they have a smaller impact on evaluation functions [5].

2.1 Algorithms

2.1.1 Local Search

The Local Search algorithm is a single-solution algorithm; it begins with an initial feasible solution and improves upon it, it cannot create a solution from scratch. It starts with the initial solution and checks if a small random change improves the solution. If it does, it commits the change and starts again. This will lead to a locally optimal solution, but not always a globally optimal solution. Local Search is incapable of escaping local optima since it would need to temporarily decrease solution quality. It is therefore not commonly used in scheduling.

2.1.2 Greedy Algorithms

Greedy Algorithms are naive single-solution algorithms. The name comes from the behaviour of the algorithms, they choose the at-the-time best solution without any thought about future choices [6]. Greedy Algorithms only do one pass through of a solution, and do not iterate or improve upon the solution thereafter.

Greedy Algorithms can be applied in many different ways, in this thesis one version will be created. The selected variant will hereafter be referred to as Greedy Algorithm in this thesis.

2.1.3 Genetic Algorithms

Genetic Algorithms have been widely applied in scheduling research [3] [4] including teacher scheduling [7], freight transport scheduling [8], and university course scheduling [3]. Genetic Algorithms are based on Darwinian evolution [9] [10], using selection, merging, and mutation to solve complex problems such as scheduling. Genetic Algorithms are population-based, which means that they explore multiple possible candidate solutions simultaneously. Mutation and merging operators are used to iteratively combine and create new generations of solutions.

Genetic Algorithms can be broken into three separate parts:

1. **Genetic representation**: Each solution needs to be translated to a genetic code. The genetic representation is a mapping between the genetic code and the solution that allows for genetic operators, such as mutation and merging.
2. **Specimen selection**: Each population of solutions is a result of the last. Like evolution the algorithm selects the top scoring solutions from the previous generation, to create the next.
3. **Genetic operators**: A new population is devised from the chosen candidates. Small random changes are applied (mutation) and multiple candidates may be combined to create new solutions (merging). Since changes are applied randomly Genetic Algorithms can escape local optima.

Genetic Algorithms can be applied in many different ways, in this thesis one version will be created. The selected variant will hereafter be referred to as Genetic Algorithm in this thesis.

2.1.4 Tabu Search

Tabu Search is a metaheuristic algorithm [11] that has been commonly used in scheduling research such as examination timetabling [12], electrical maintenance scheduling [13], and minimum spanning tree problems [14]. Fundamentally Tabu Search is based on local search, which means it is a single-solution algorithm, unlike the Genetic Algorithm [15]. A downside of local search algorithms is that they often get stuck in local optima since they only move to a neighbour if it gives a better result. Conversely Tabu Search can escape these local optima since it allows a temporary worsening of the solution, if no better moves are available [14] [16].

To further improve upon local search, Tabu Search has a Tabu list which contains recently visited solutions to improve efficiency and avoid revisiting suboptimal solutions [4]. The existence of the Tabu list means that Tabu Search will use more memory than a standard local search implementation.

The algorithm can be broken down into a series of repeating steps [17]:

1. **Start with an initial solution**: Tabu search does not have the ability to create new solutions, only improve existing ones. Therefore one needs an initial solution to improve.
2. **Make a neighbourhood**: Create new candidate solutions by introducing small random changes to the current solution.
3. **Evaluate the neighbourhood**: Check the fitness score of the candidate solutions.
4. **Make a move**: Choose a new solution to become the current solution. This is decided through the new solution's fitness score and cross-referencing the Tabu list.

2.1.5 Simulated Annealing

Like Genetic Algorithm, Simulated Annealing is a reality-based algorithm. It has been commonly used in scheduling research such as laboratory automation [18], electrical maintenance scheduling [13], and airline scheduling [19]. Whilst Genetic Algorithm is based on evolution Simulated Annealing is based on heating and slowly cooling metal. Simulated Annealing is also quite similar to Tabu Search, both are probabilistic single-solution local-search algorithms. The natural component is integrated through a temperature, which starts high and slowly cools (like metal). The algorithm avoids getting stuck in local optima by occasionally accepting worse solutions when the temperature is high. Due to the cooling temperature, the algorithm takes more risks early on and therefore escapes local optima that a standard local search implementation would get stuck in.

The algorithm can be broken down into a series of repeating steps [17]:

1. **Initial solution**: Like Tabu Search, Simulated Annealing cannot create a new solution, only improve on an existing one. Therefore one needs an initial solution to improve.
2. **Make a neighbourhood**: Create new candidate solutions by introducing small random changes to the current solution.
3. **Evaluate neighbourhood**: Check the fitness score of the candidate solutions.
4. **Make a move**: Choose a new solution to be the current solution. This is decided through the new solution's fitness score and the current temperature.
5. **Temperature adjustment**: Lower the temperature by reassigning the variable value.

3
Methods

A survey was created to collect information about Teaching Assistant opinions on the current scheduling systems and possible solutions. The survey had a larger scope and focused on Teaching Assistant availability input methods as well as fairness and constraints. The scope of the project has since changed and will focus solely on the algorithms and fairness, excluding input methods and graphical components due to time constraints.

The survey, located in its entirety in Appendix A, was sent to Teaching Assistants at CSE. The survey was sent to 231 individuals and received 40 answers corresponding to an answer rate of 17.3%. The survey focused on what Teaching Assistants desired and valued in scheduling systems.

The survey results (shown in Appendix B) were used to determine which aspects of scheduling Teaching Assistants valued. According to the survey results 92.5% of correspondents found the ability to assert temporal preferences important, which translates into the soft constraint SC:01, defined below. Additionally when asked about their definition of a "fair" schedule 50% of correspondents stated that time preferences needed to be included.

Fair scheduling was defined by 85% of correspondents as an equal amount of worked hours within the teaching team. Since the amount of assigned hours can differ greatly between Teaching Assistants within a teaching team a soft constraint was devised to compare the proportion of unused hours within the course, SC:02. SC:02 was implemented using the standard deviation of the percentage of used hours.

The soft constraints are:

SC:01 A Teaching Assistant shall not be scheduled during unwanted times.

SC:02 A Teaching Assistant shall not be scheduled more than others.

Hard constraints define the feasibility of the solutions and are heavily penalized in the fitness function. These were chosen by examining the domain and previous scheduling systems.

The hard constraints are:

HC:01 A Teaching Assistant shall not be scheduled more hours than allotted.

HC:02 A Teaching Assistant shall not be scheduled when unavailable.

HC:03 A session shall at least have the minimum required amount of Teaching Assistants.

A fitness function, sometimes called an evaluation function, was constructed to evaluate and compare the potential schedules. The formula for the fitness score is shown in equation 1. The hard constraint penalty was chosen to be 1000 for each violation of HC:02 and HC:03, and 10 per minute of overtime (HC:01). The soft constraint penalty was significantly lower at 10 per unwanted session and 100 times the variance in the percentage of used hours. The difference in penalty reflects the impact of the violations, but the specific penalty values were chosen ad-hoc. It is possible that other values could lead to more precise optimization. The optimization and evaluation of penalty values was beyond the scope of this thesis.

$$Score = sumHardViolations \times hardPenalty + sumSoftViolations \times softPenalty \tag{3.1}$$

The chosen algorithms Greedy Algorithm, Genetic Algorithm, Tabu Search, and Simulated Annealing were selected to cover a wide range of algorithmic approaches. The Genetic Algorithm as a population-based metaheuristic evolutionary algorithm. Tabu Search and Simulated Annealing as Local Search based single solution metaheuristic algorithms. Greedy Algorithm is included as a baseline constructive heuristic algorithm, that allows for evaluation of whether added algorithmic complexity leads to a significant increase in solution quality. All of the algorithms have a wide scientific background within scheduling research and are capable of handling constraint based optimization even on larger problem sets [3] [4].

Other algorithms, such as swarm-based algorithms [20] and Graph Coloring [21], were considered, but were excluded due to being less prevalent in university-based scheduling research. Integer or Linear Programming solutions were not considered due to the limited feasibility in large scheduling problems [5].

The programming language Java was used to implement the algorithms due to previous experience and good library support. The date and time handling as well as .csv reading was efficiently done using libraries.

The main parameters for the scheduling algorithms were determined through preliminary experimentation. A range of parameter values were tested for each algorithm, and the final values were chosen based on their ability to provide a balance between solution quality and computational efficiency. Further increases in parameter values generally resulted in marginal improvements in solution quality while increasing execution time. The full implementation is available in a public GitHub repository [22].

The Greedy Algorithm schedules the available Teaching Assistant with the greatest number of remaining hours. This selection rule was chosen to schedule Teaching Assistants evenly in accordance to SC:02. The approach serves as a straightforward baseline, allowing the performance of more complex algorithms to be evaluated against a simple scheduling strategy. As the algorithm does not rely on tuneable parameters, no calibration was required.

For the Genetic Algorithm, population size and generation count were varied during preliminary testing. A population size of 100 and 20 generations were selected, as larger values produced only minor improvements in solution quality with major increases in execution time. The selected values therefore provided a balance between exploration of the solution space and computational efficiency.

For Tabu Search, different Tabu list sizes were evaluated, and a size of 20 was selected. Smaller Tabu lists resulted in more frequent revisiting of previously explored solutions, whereas larger lists restricted the search excessively. A size of 20 provided sufficient diversification while maintaining search flexibility.

For Simulated Annealing, several combinations of initial temperatures and cooling schedules were evaluated. An initial temperature of 1000 with a cooling rate of 5% per iteration was chosen because it allowed extensive exploration during the early stages of the search while still converging within a reasonable number of iterations. Higher temperatures or slower cooling schedules did not produce substantial improvements in solution quality in the tested instances.

Course schedules were imported using the public facing TimeEdit schedule for the courses. This was chosen over private API key usage due to ease of access and use. The minimum amount of Teaching Assistants required was set to 3 as a default and could be manually adjusted for each session.

To test the algorithms, test cases needed to be constructed. Real courses at CSE were used to be as true to the potential deployment environment as possible. Additionally 5 Teaching Assistants personas were constructed using the results from the survey.

The survey shows that the largest availability issue for Teaching Assistants is their own course schedule and exams followed by personal commitments. Therefore the Teaching Assistant personas unavailable times were chosen by selecting two courses in the same study period as the course to be scheduled and blocking lectures etc. Additionally Teaching Assistants 1 and 2 have additional time slots blocked each week to simulate personal commitments.

Teaching Assistants 2 and 3 might have a long commute and therefore have a time preference against early morning sessions. Teaching Assistant 4 has a time preference against late afternoons to simulate needing to pick up children from preschool.

To minimise the impact of randomness when doing the comparison, Tabu Search and Simulated Annealing both receive the same initial solution, this ensures a fair comparison. The initial solution was created by selecting the top solution from a initial population. The initial population had a size of 50 and the solutions were all created by randomly assigning Teaching Assistants to sessions. The time to create the initial solution was added onto the runtimes for Tabu Search and Simulated Annealing to ensure fair comparison between all of the algorithms.

4

Results

The results of the Genetic Algorithm, Tabu Search, and Simulated Annealing will all vary between executions since the algorithms are non-deterministic. According to execution time Greedy Algorithm performs the best. According to the fitness score (a metric for solution quality) Tabu Search performed the best in all cases. Tabu Search performed the best overall, the improvement in solution quality far outweighs the additional execution time and algorithmic complexity compared to Greedy Algorithm.

A score of 0 means there are no hard-constraint violations or soft-constraint violations.

		5 Teaching Assistants		4 Teaching Assistants	
Course	Algorithm	Score	Time (ms)	Score	Time (ms)
Course 1	Genetic	80	707	130	733
	Greedy	25140	2	33180	1
	Tabu	0	107	70	61
	Simulated Annealing	0	258	70	320
Course 2	Genetic	0	462	60	568
	Greedy	43000	1	44000	1
	Tabu	0	39	0	57
	Simulated Annealing	0	152	0	250
Course 3	Genetic	60	586	120	533
	Greedy	5150	0	7180	1
	Tabu	0	38	80	50
	Simulated Annealing	0	163	80	168

Table 4.1: Comparison of scheduling algorithms across courses and Teaching Assistant availability

5

Discussion

This section will focus on analysing the results presented in the previous chapter (4) and discussing potential future studies and improvements.

5.1 Analysis of results

The results of the Genetic Algorithm, Tabu Search, and Simulated Annealing were all variable between executions since the algorithms are non-deterministic. The variations were very small, generally less than 50 points but in some cases extreme variations occurred. Since the majority of the variation was minimal it did not impact the conclusions that could be drawn from the results. The results are therefore presented as a single table for clarity and readability.

The Greedy Algorithm performed worst overall when evaluating solution quality. This is because the Greedy Algorithm simply selects the available Teaching Assistant that has the least amount of used hours. This causes it to schedule Teaching Assistants sub-optimally. From a time perspective Greedy Algorithm performs the best but due to the large sacrifice in solution quality, one can conclude that an increase in algorithmic complexity is worth the decrease in computational performance.

Tabu Search and Simulated Annealing generally have very similar results. This is probably due to the fact that they are similar algorithmically. Both are local-search-based algorithms with ways to escape local optima. For Simulated Annealing this is done through the temperature, which controls the probability to visit a worse solution and therefore escape a local optimum. In Tabu Search this is done through the Tabu list, which forces the algorithm to search new areas of the solution space, which allows it to escape local optima.

The downside with Tabu Search is the memory that is needed for the Tabu List. Simulated Annealing and Greedy Algorithm do not need this additional list, and may therefore use less memory. Therefore, if the focus is on minimizing memory usage, Simulated Annealing might be a better alternative despite being less time efficient.

In the specific use-case of scheduling Teaching Assistants at the Department of Computer Science and Engineering at Chalmers University of Technology and the University of Gothenburg this additional memory requirement was deemed negligible according to Niklas Broberg assistant head of department for undergraduate and graduate education. Tabu Search is therefore the best algorithm for the specific problem of scheduling Teaching Assistants at the department.

5.2 Reproducibility

The Chalmers University of Technology data office will be contacted to investigate the opportunity to publish the data from the Teaching Assistant Survey in the Svenska Nationella Datatjänst network [23]. This would allow the data to be used in future research.

5.3 Future work

Currently the solution schedules that are created, can be printed out in a human-readable format shown in Appendix C. Each Teaching Assistant has a unique identifier, which shows up under the sessions they are scheduled for. It is also possible to query all sessions to which a given Teaching Assistant is assigned.

A future improvement could be to create a user interface for Teaching Assistants, where it would be possible to easily add unavailable times and preferences. Additionally this system could include a graphical representation of the Teaching Assistant's schedule in a TimeEdit like format. This would greatly increase usability and allow for real world deployment.

A user study could be performed where the system is used to schedule real courses at the Department of Computer Science and Engineering at Chalmers University of Technology and the University of Gothenburg. This would allow for more precise feedback from Teaching Assistants, and could narrow down potential mismatches between the identified constraints and the user needs.

The system could also be modified to import schedules automatically through TimeEdit using API access. This was not available at the start of the project due to privacy concerns regarding personal schedules, and was therefore excluded from this thesis. But it could potentially be included if the system were to be deployed. This would further minimize the manual labor that is needed to schedule courses, since schedules could be imported automatically without an administrator working as a middle man by downloading .csv files.

Bibliography

[1] G. Ausiello and V. Paschos, "Reductions, completeness and the hardness of approximability," *European Journal of Operational Research*, vol. 172, no. 3, pp. 719–739, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221705005011

[2] E. K. Burke and S. Petrovic, "Recent research directions in automated timetabling," *European Journal of Operational Research*, vol. 140, no. 2, pp. 266–280, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221702000693

[3] Z. Zhu, Y. Zhou, and Q. Luo, "A hyper-heuristic algorithm based on genetic and greedy strategy for university course scheduling problem," *Applied Soft Computing*, vol. 186, p. 114150, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494625014632

[4] K. Kalita, N. Ganesh, and S. Balamurugan, *Metaheuristics for Machine Learning.* John Wiley Sons, Inc, 2024.

[5] K. Xiang, X. Hu, M. Yu, and X. Wang, "Exact and heuristic methods for a university course scheduling problem," *Expert Systems with Applications*, vol. 248, p. 123383, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424002483

[6] C.-Y. R. Huang, C.-Y. Lai, and K.-T. T. Cheng, "Chapter 4 - fundamentals of algorithms," in *Electronic Design Automation*, L.-T. Wang, Y.-W. Chang, and K.-T. T. Cheng, Eds. Boston: Morgan Kaufmann, 2009, pp. 173–234. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123743640500114

[7] Y.-Z. Wang, "An application of genetic algorithm methods for teacher assignment problems," *Expert Systems with Applications*, vol. 22, no. 4, pp. 295–302, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417402000179

[8] A. Sahli, W. Behiri, S. Belmokhtar-Berraf, and C. Chu, "An effective and robust genetic algorithm for urban freight transport scheduling using passenger rail network," *Computers Industrial Engineering*, vol. 173, p. 108645, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835222006337

[9] Nationalencyklopedin, "darwinism," Accessed 2026-05-30. [Online]. Available: https://www.ne.se/uppslagsverk/encyklopedi/lÃěng/darwinism

[10] Chun-Wei Tsai and Ming-Chao Chiang, *Handbook of Metaheuristic Algorithms.* Academic Press, 2023, ch. Chapter Seven: Genetic algorithm.

[11] C.-W. Tsai and M.-C. Chiang, “Chapter four - metaheuristic algorithms,” in *Handbook of Metaheuristic Algorithms*, ser. Uncertainty, Computational Techniques, and Decision Intelligence. Academic Press, 2023, pp. 71–78. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780443191084000174

[12] S. Abdullah, S. Ahmadi, E. K. Burke, M. Dror, and B. McCollum, “A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem,” *Journal of the Operational Research Society*, vol. 58, no. 11, pp. 1494–1502, 2007. [Online]. Available: https://doi.org/10.1057/palgrave.jors.2602258

[13] A. Froger, M. Gendreau, J. E. Mendoza, Éric Pinson, and L.-M. Rousseau, “Maintenance scheduling in the electricity industry: A literature review,” *European Journal of Operational Research*, vol. 251, no. 3, pp. 695–706, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221715008012

[14] P. C. Pop, “The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances,” *European Journal of Operational Research*, vol. 283, no. 1, pp. 1–15, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221719304217

[15] T. Öncan, J.-F. Cordeau, and G. Laporte, “A tabu search heuristic for the generalized minimum spanning tree problem,” *European Journal of Operational Research*, vol. 191, no. 2, pp. 306–319, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221707008934

[16] R. Dechter, “chapter 7 - stochastic greedy local search,” in *Constraint Processing*, ser. The Morgan Kaufmann Series in Artificial Intelligence, R. Dechter, Ed. San Francisco: Morgan Kaufmann, 2003, pp. 191–208. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9781558608900500086

[17] M. Abd-El-Barr, “Topological network design: A survey,” *Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 501–509, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S108480450800101X

[18] Y. Arai, K. Takahashi, T. Horinouchi, K. Takahashi, and H. Ozaki, “Sagas: Simulated annealing and greedy algorithm scheduler for laboratory automation,” *SLAS Technology*, vol. 28, no. 4, pp. 264–277, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2472630323000195

[19] M. E. Çiftçi and V. Özkır, “Optimising flight connection times in airline bank structure through simulated annealing and tabu search algorithms,” *Journal of Air Transport Management*, vol. 87, p. 101858, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969699720300545

[20] K.-C. Ying, P. Pourhejazy, and K.-L. Huang, “Tracing the evolution of particle swarm optimization in scheduling: A systematic review using main path analysis,” *Swarm and Evolutionary Computation*, vol. 100, p. 102239, 2026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650225003967

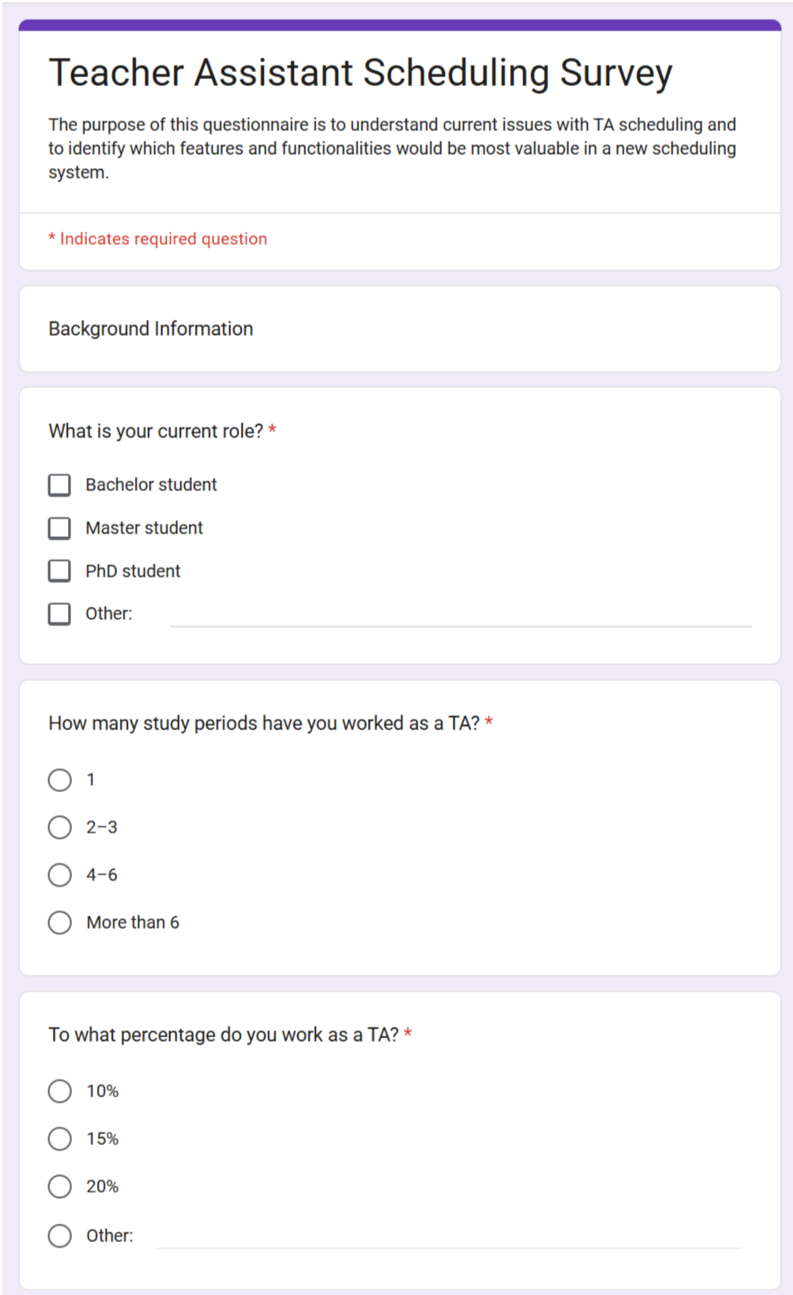
[21] S. Thevenin, N. Zufferey, and J.-Y. Potvin, "Graph multi-coloring for a job scheduling application," *Discrete Applied Mathematics*, vol. 234, pp. 218–235, 2018, special Issue on the Ninth International Colloquium on Graphs and Optimization (GO IX), 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166218X16302487

[22] C. Dusén, 2026. [Online]. Available: https://github.com/kilodussin/BachelorThesis

[23] Swedish National Data Service, "SND makes all types of digital research data accessible," Accessed 2026-05-30. [Online]. Available: https://snd.se/en/about-us

A

Appendix 1



Teacher Assistant Scheduling Survey

The purpose of this questionnaire is to understand current issues with TA scheduling and to identify which features and functionalities would be most valuable in a new scheduling system.

* Indicates required question

Background Information

What is your current role? *

- Bachelor student
- Master student
- PhD student
- Other:

How many study periods have you worked as a TA? *

- 1
- 2-3
- 4-6
- More than 6

To what percentage do you work as a TA? *

- 10%
- 15%
- 20%
- Other:

Figure A.1: Part 1 Teaching Assistant Survey

Availability Constraints

Which of the following affect your availability as a TA? *
(Select all that apply)

- [ ] My own course schedule
- [ ] Exams
- [ ] Work outside university
- [ ] Family or personal commitments
- [ ] Health-related reasons
- [ ] Other: ______

How do you currently communicate unavailable times to course responsibles? *
(Select all that apply)

- [ ] Spreadsheet (Google Sheets/Excel)
- [ ] Email
- [ ] Messaging platforms (Slack, Discord, etc.)
- [ ] Verbal agreement
- [ ] Other: ______

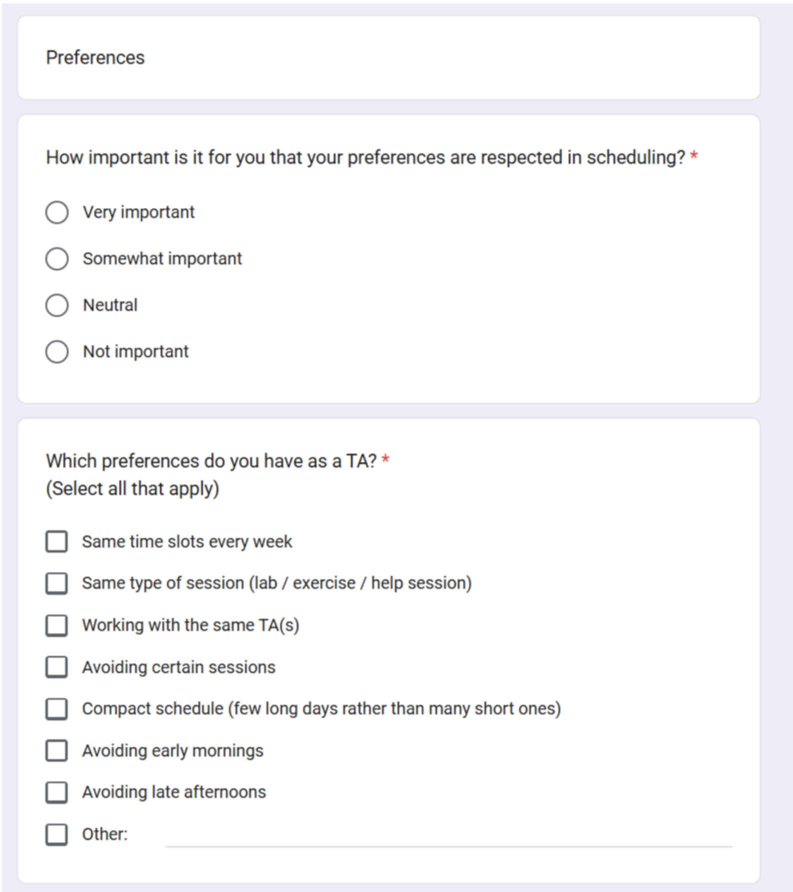
How precise is your availability? *

- [ ] I can specify exact time slots
- [ ] I can specify days but not exact times
- [ ] My availability changes week to week
- [ ] I often don't know far in advance
- [ ] Other: ______

Are there time slots you can never work? *
(Select all that apply)

- [ ] Early mornings
- [ ] Late afternoons/evenings
- [ ] Fridays
- [ ] Weekends
- [ ] No
- [ ] Other: ______

Figure A.2: Part 2 Teaching Assistant Survey



Preferences

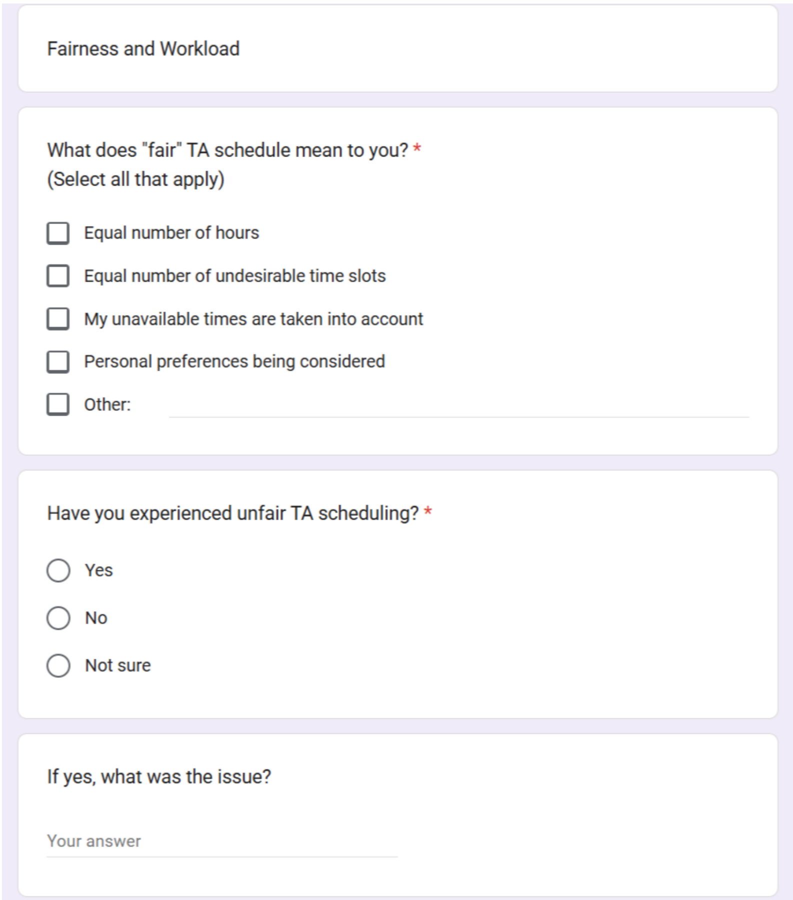
How important is it for you that your preferences are respected in scheduling? *

- Very important
- Somewhat important
- Neutral
- Not important

Which preferences do you have as a TA? *
(Select all that apply)

- Same time slots every week
- Same type of session (lab / exercise / help session)
- Working with the same TA(s)
- Avoiding certain sessions
- Compact schedule (few long days rather than many short ones)
- Avoiding early mornings
- Avoiding late afternoons
- Other:

Figure A.3: Part 3 Teaching Assistant Survey



Fairness and Workload

What does "fair" TA schedule mean to you? *
(Select all that apply)

- [ ] Equal number of hours
- [ ] Equal number of undesirable time slots
- [ ] My unavailable times are taken into account
- [ ] Personal preferences being considered
- [ ] Other:

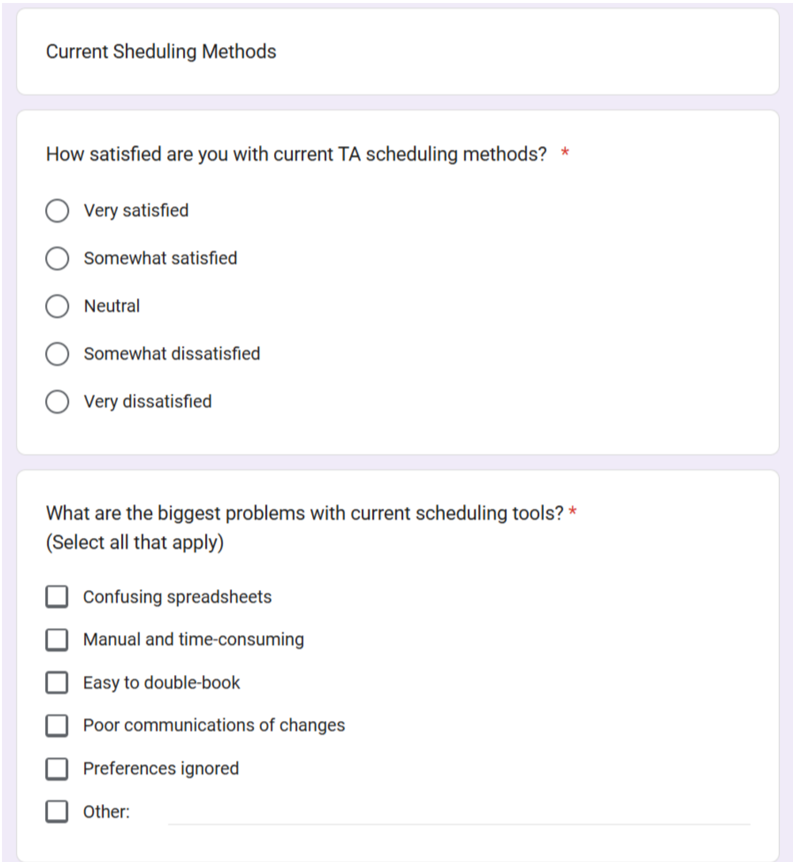
Have you experienced unfair TA scheduling? *

- ○ Yes
- ○ No
- ○ Not sure

If yes, what was the issue?

Your answer

Figure A.4: Part 4 Teaching Assistant Survey



Current Sheduling Methods

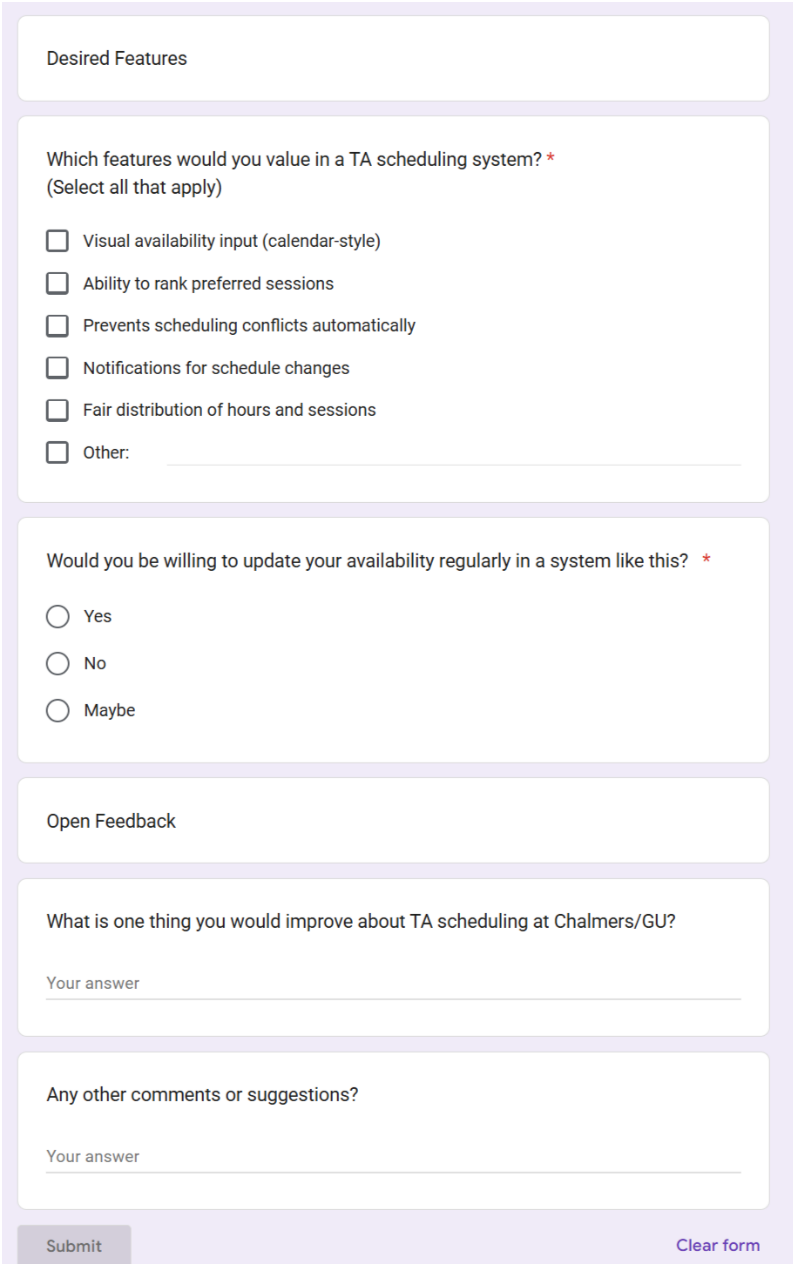
How satisfied are you with current TA scheduling methods? *

- Very satisfied
- Somewhat satisfied
- Neutral
- Somewhat dissatisfied
- Very dissatisfied

What are the biggest problems with current scheduling tools? *
(Select all that apply)

- Confusing spreadsheets
- Manual and time-consuming
- Easy to double-book
- Poor communications of changes
- Preferences ignored
- Other:

Figure A.5: Part 5 Teaching Assistant Survey



Desired Features

Which features would you value in a TA scheduling system? *
(Select all that apply)

- [ ] Visual availability input (calendar-style)
- [ ] Ability to rank preferred sessions
- [ ] Prevents scheduling conflicts automatically
- [ ] Notifications for schedule changes
- [ ] Fair distribution of hours and sessions
- [ ] Other:

Would you be willing to update your availability regularly in a system like this? *

- ◯ Yes
- ◯ No
- ◯ Maybe

Open Feedback

What is one thing you would improve about TA scheduling at Chalmers/GU?

Your answer

Any other comments or suggestions?

Your answer

Submit

Clear form

Figure A.6: Part 6 Teaching Assistant Survey

B

Appendix 2

How important is it for you that your preferences are respected in scheduling?	Which of the following affect your availability as a TA? (Select all that apply)
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams, Family or personal commitments
Very important	My own course schedule, Exams, Work outside university, Family or personal commitments
Very important	My own course schedule
Very important	My own course schedule, Exams, Family or personal commitments
Very important	My own course schedule, Exams
Very important	My own course schedule
Somewhat important	My own course schedule
Somewhat important	Family or personal commitments
Somewhat important	My own course schedule, Family or personal commitments
Very important	My own course schedule, Exams
Very important	My own course schedule, Family or personal commitments
Somewhat important	My own course schedule, Exams
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule
Very important	My own course schedule, Exams
Neutral	My own course schedule
Very important	My own course schedule, Family or personal commitments
Very important	My own course schedule, Exams
Very important	My own course schedule
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams, Family or personal commitments
Very important	My own course schedule, Family or personal commitments
Neutral	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams
Very important	My own course schedule, Exams, Family or personal commitments, Health-related reasons
Somewhat important	My own course schedule, Exams
Very important	My own course schedule
Somewhat important	My own course schedule
Somewhat important	My own course schedule, Family or personal commitments
Very important	My own course schedule, Exams, Health-related reasons, Student union activities
Neutral	My own course schedule
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Exams, Family or personal commitments
Somewhat important	My own course schedule, Family or personal commitments
Somewhat important	My own course schedule, Exams
Somewhat important	My own course schedule, Family or personal commitments
Very important	My own course schedule, Exams

Figure B.1: Part 1 Teaching Assistant Survey Results

Which preferences do you have as a TA? (Select all that apply)
Same type of session (lab / exercise / help session), Avoiding late afternoons
Avoiding early mornings
Compact schedule (few long days rather than many short ones)
Avoiding early mornings
Same time slots every week, Compact schedule (few long days rather than many short ones)
Same time slots every week
Avoiding early mornings, Avoiding late evenings (17-21)
Same time slots every week, Same type of session (lab / exercise / help session), Working with the same TA(s), Compact schedule (few long days rather than many short ones), Avoiding early mornings
Same time slots every week, Same type of session (lab / exercise / help session)
Avoiding early mornings
That the sessions do not clash with my own course schedule.
Same time slots every week, Avoiding late afternoons
Compact schedule (few long days rather than many short ones), Avoiding early mornings
Compact schedule (few long days rather than many short ones)
Same time slots every week, Same type of session (lab / exercise / help session), Avoiding early mornings
Same time slots every week, Avoiding early mornings
Same time slots every week, Same type of session (lab / exercise / help session)
Same time slots every week, Avoiding early mornings
Same type of session (lab / exercise / help session)
Same type of session (lab / exercise / help session)
Same time slots every week, Same type of session (lab / exercise / help session)
Same time slots every week, Same type of session (lab / exercise / help session), Working with the same TA(s), Avoiding early mornings
Same time slots every week, Avoiding early mornings
no
Same time slots every week, Avoiding certain sessions, Compact schedule (few long days rather than many short ones)
Same time slots every week, Same type of session (lab / exercise / help session), Avoiding late afternoons
Same type of session (lab / exercise / help session), Compact schedule (few long days rather than many short ones)
Same time slots every week, Same type of session (lab / exercise / help session)
Working with the same TA(s), Avoiding early mornings
does not matter to me
Same time slots every week
Same time slots every week, Same type of session (lab / exercise / help session), Working with the same TA(s)
My unavailable times are taken into account
Same time slots every week, Same type of session (lab / exercise / help session)
Same time slots every week, Same type of session (lab / exercise / help session), Working with the same TA(s)
Same time slots every week, Compact schedule (few long days rather than many short ones)
Avoiding early mornings
No preference
Same time slots every week, Avoiding early mornings
Working with the same TA(s), Compact schedule (few long days rather than many short ones), Avoiding late afternoons

Figure B.2: Part 2 Teaching Assistant Survey Results

What does "fair" TA schedule mean to you? (Select all that apply)
Equal number of hours, Equal number of undesirable time slots, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours
Equal number of hours, Personal preferences being considered
My unavailable times are taken into account
Equal number of hours, My unavailable times are taken into account
Equal number of hours, My unavailable times are taken into account
Personal preferences being considered
My unavailable times are taken into account
Equal number of hours, Equal number of undesirable time slots
Equal number of hours, My unavailable times are taken into account
Equal number of hours, Equal number of undesirable time slots
Equal number of hours, Personal preferences being considered
Personal preferences being considered
My unavailable times are taken into account
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours
Equal number of hours
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours
Equal number of hours
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours
Equal number of hours
Equal number of hours
Equal number of hours, Equal number of undesirable time slots, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours
Equal number of hours
My unavailable times are taken into account
Equal number of hours, Equal number of undesirable time slots
Equal number of hours
Equal number of hours, Equal number of undesirable time slots, My unavailable times are taken into account, Personal preferences being considered
Equal number of hours

Figure B.3: Part 3 Teaching Assistant Survey Results

Which features would you value in a TA scheduling system? (Select all that apply)
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions, Ability to mark unavaliability, ability to report hours worked outside of schedule
Visual availability input (calendar-style), Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Notifications for schedule changes
Visual availability input (calendar-style), Prevents scheduling conflicts automatically, Fair distribution of hours and sessions
Ability to rank preferred sessions, Fair distribution of hours and sessions
Prevents scheduling conflicts automatically, Notifications for schedule changes
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style)
Visual availability input (calendar-style), Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Fair distribution of hours and sessions
Notifications for schedule changes
Ability to rank preferred sessions, Notifications for schedule changes
Ability to rank preferred sessions, Prevents scheduling conflicts automatically
Visual availability input (calendar-style), Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Notifications for schedule changes
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Notifications for schedule changes
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Prevents scheduling conflicts automatically
Notifications for schedule changes
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Fair distribution of hours and sessions
Visual availability input (calendar-style), Fair distribution of hours and sessions
Visual availability input (calendar-style), Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Fair distribution of hours and sessions
Visual availability input (calendar-style), Ability to rank preferred sessions, Prevents scheduling conflicts automatically, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style), Prevents scheduling conflicts automatically, Notifications for schedule changes
Visual availability input (calendar-style), Prevents scheduling conflicts automatically
Notifications for schedule changes, Fair distribution of hours and sessions
Ability to rank preferred sessions, Notifications for schedule changes, Fair distribution of hours and sessions
Visual availability input (calendar-style)

Figure B.4: Part 4 Teaching Assistant Survey Results

C
Appendix 3

A sample of a schedule output:
Start: 2025-09-04T13:15 End: 2025-09-04T15:00
TA1
TA3
TA4

Start: 2025-09-04T15:15 End: 2025-09-04T17:00
TA2
TA5
TA1

Start: 2025-09-05T08:00 End: 2025-09-05T09:45
TA4
TA5
TA1

Start: 2025-09-08T15:15 End: 2025-09-08T17:00
TA3
TA5
TA1

Start: 2025-09-11T13:15 End: 2025-09-11T15:00
TA1
TA5
TA4

Start: 2025-09-11T15:15 End: 2025-09-11T17:00
TA3
TA2
TA1

Start: 2025-09-12T08:00 End: 2025-09-12T09:45
TA5
TA1
TA4

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG