

Customizable Roundtrips with Tour4Me

Meta-heuristic Approaches for Personalized Running and Cycling Routes

Lisa Salewsky

TU Dortmund, Fakultät für Informatik



Reviewer:
Prof. Dr. Kevin Buchin
Mart Hagedoorn, M. Sc.

24. November 2023

Einleitung

Warum eine App zum Erstellen von Rundrips?

Einleitung

Warum eine App zum Erstellen von Rundrips?



Einleitung

Warum eine App zum Erstellen von Rundrips?



Einleitung

Warum eine App zum Erstellen von Rundrips?



Einleitung

Warum eine App zum Erstellen von Rundrips?



Einleitung

Warum eine App zum Erstellen von Rundrips?



Gemeinsamkeiten

- Kann von A nach B gehen

Einleitung

Warum eine App zum Erstellen von Rundrips?



Gemeinsamkeiten

- Kann von A nach B gehen
- In der Freizeit aber oft Rundtrips

Einleitung

Warum eine App zum Erstellen von Rundrips?



Gemeinsamkeiten

- Kann von A nach B gehen
- In der Freizeit aber oft Rundtrips
- Möglichst "häbsche" Routen mit zusätzlichen individuellen Wünschen gewünscht

Einleitung

Warum eine App zum Erstellen von Rundrips?

Einleitung

Warum eine App zum Erstellen von Rundrips?

- viele Lösungen für kürzeste Wege von A nach B

Einleitung

Warum eine App zum Erstellen von Rundrips?

- viele Lösungen für kürzeste Wege von A nach B
- Optimieren alltägliche Routen

Einleitung

Warum eine App zum Erstellen von Rundrips?

- viele Lösungen für kürzeste Wege von A nach B
- Optimieren alltägliche Routen
- Joggen und Radfahren in der Freizeit:

Einleitung

Warum eine App zum Erstellen von Rundrips?

- viele Lösungen für kürzeste Wege von A nach B
- Optimieren alltägliche Routen
- Joggen und Radfahren in der Freizeit:
- weitere Bedingungen für attraktive Strecken nötig

Einleitung

Warum eine App zum Erstellen von Rundrips?

- viele Lösungen für kürzeste Wege von A nach B
- Optimieren alltägliche Routen
- Joggen und Radfahren in der Freizeit:
- weitere Bedingungen für attraktive Strecken nötig
- aber kaum Ansätze für Rundtrips mit Präferenzen

Einleitung

Was macht eine Strecke attraktiv?

Einleitung

Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person

Einleitung

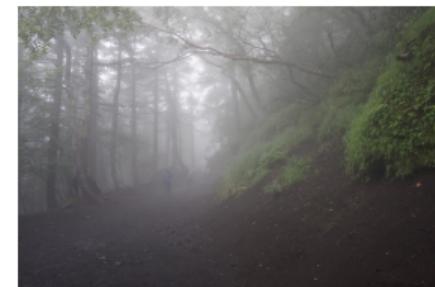
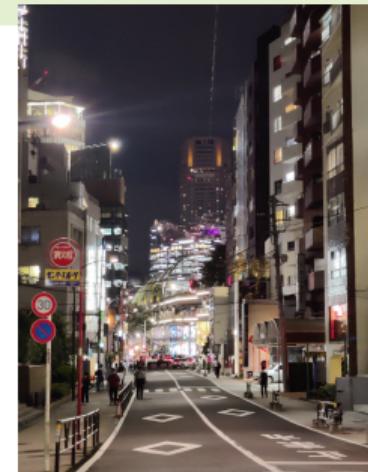
Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren

Einleitung

Was macht eine Strecke attraktiv?

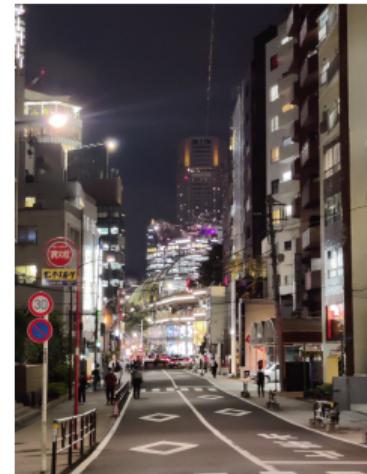
- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)



Einleitung

Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)
 - Untergrund (Teer, Kies, Sand, Erde, etc.)



Einleitung

Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)
 - Untergrund (Teer, Kies, Sand, Erde, etc.)
 - Steigung



Einleitung

Was macht eine Strecke attraktiv?

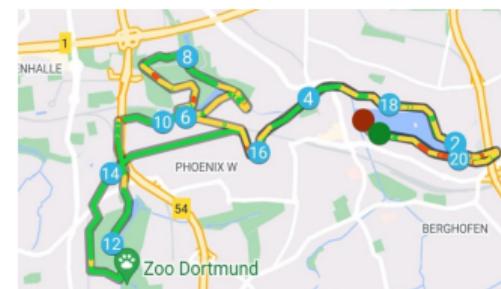
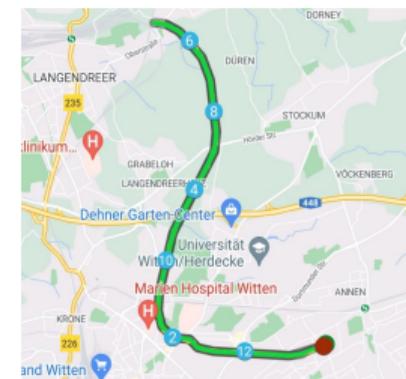
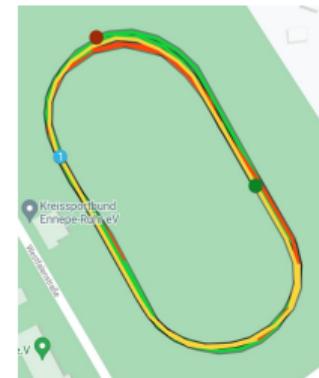
- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)
 - Untergrund (Teer, Kies, Sand, Erde, etc.)
 - Steigung
 - Umgebung (Wald, Park, Wohngegend, etc.)



Einleitung

Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)
 - Untergrund (Teer, Kies, Sand, Erde, etc.)
 - Steigung
 - Umgebung (Wald, Park, Wohngegend, etc.)
 - Form (Rund, gerade mit U-Turn, viele Abzweigungen, etc.)



Einleitung

Was macht eine Strecke attraktiv?

- abhängig von individuellen Bedürfnissen und vom Level der Person
- viele verschiedene Faktoren
 - Wegtyp (Straße, Waldpfad, Radweg etc.)
 - Untergrund (Teer, Kies, Sand, Erde, etc.)
 - Steigung
 - Umgebung (Wald, Park, Wohngegend, etc.)
 - Form (Rund, gerade mit U-Turn, viele Abzweigungen, etc.)
- Einfluss dieser auf Strecke individuell auswählen

Was ist das Ziel?

Was ist das Ziel?

- Grundlage: Tour4Me

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr
Parameter, Änderung des Interfaces

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr
Parameter, Änderung des Interfaces

- Einbeziehen der genannten Faktoren

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen: Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr Parameter, Änderung des Interfaces

- Einbeziehen der genannten Faktoren
- algorithmisch und in der GUI

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen: Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr Parameter, Änderung des Interfaces

- Einbeziehen der genannten Faktoren
- algorithmisch und in der GUI
- Anpassen der GUI auf neue Optionen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen: Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr Parameter, Änderung des Interfaces

- Einbeziehen der genannten Faktoren
- algorithmisch und in der GUI
- Anpassen der GUI auf neue Optionen

Was ist das Ziel?

- Grundlage: Tour4Me
- App erweitern und nutzbarer machen

Mögliche algorithmische Erweiterungen:
Metaheuristiken

- AntColony
- Simulated Annealing
- Genetisch
- Kombinationen

Mögliche UI Erweiterungen: Mehr
Parameter, Änderung des Interfaces

- Einbeziehen der genannten Faktoren
- algorithmisch und in der GUI
- Anpassen der GUI auf neue Optionen

⇒ allgemein: Tour4Me so anpassen, dass für (fast) jeden Startpunkt und jede Routenlänge ein (möglichst gutes) Ergebnis ausgegeben wird

Tour4Me Demo

Tour4Me

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]

⇒ Rundtrips mit Nebenbedingungen noch komplexer [8]

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
 - Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
 - algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]
- ⇒ Rundtrips mit Nebenbedingungen noch komplexer [8]
- vorhandene Tools meist eingeschränkt

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]

⇒ Rundtrips mit Nebenbedingungen noch komplexer [8]

- vorhandene Tools meist eingeschränkt
- Auswahl an Nebenbedingungen beschränkt

Viele Algorithmen, Ansätze und viel Forschung zu Kürzeste-Wege-Problemen

- Reviews, Vergleiche und Analysen [12, 15, 16]
- Zusätzlich weitere heuristische Ansätze
 - local search Varianten [2, 10, 14]
 - neighborhood basierte Ideen [2, 10, 14]
- algorithmische Routing-Probleme bereits NP-schwer [13]
 - Traveling Salesman (TSP)[9]
 - vehicle routing [2, 10]

⇒ Rundtrips mit Nebenbedingungen noch komplexer [8]

- vorhandene Tools meist eingeschränkt
- Auswahl an Nebenbedingungen beschränkt
- Editieren schwierig/ ignoriert Nebenbedingungen

Beispiele

RouteYou

Beispiele

RouteLoops

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser
 - Kombination mit bereits implementierten Ansätzen

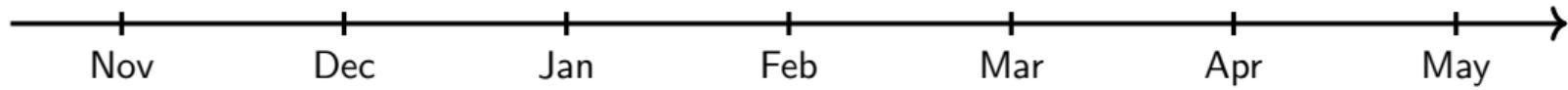
- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser
 - Kombination mit bereits implementierten Ansätzen
- Auswertung und Vergleich der Ansätze

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser
 - Kombination mit bereits implementierten Ansätzen
- Auswertung und Vergleich der Ansätze
- Anpassen des User Interfaces

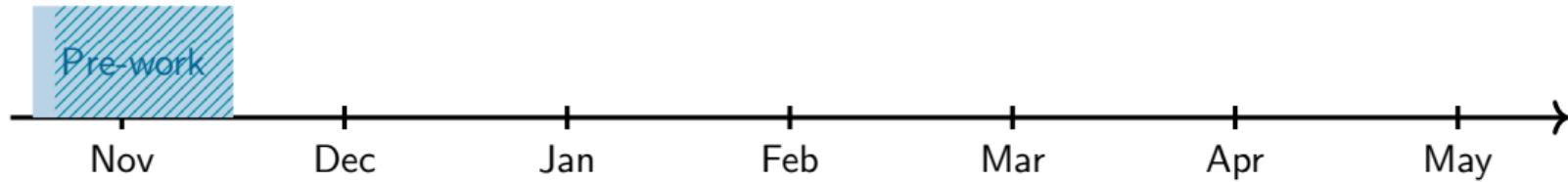
- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser
 - Kombination mit bereits implementierten Ansätzen
- Auswertung und Vergleich der Ansätze
- Anpassen des User Interfaces
- Auswahl des/der best passenden Umsetzungen

- Interface zum Testen neuer Ansätze und Metaheuristiken (C#)
- Einbauen neuer Auswahloptionen für User (z.B. Höhe/Anstieg)
- Implementieren verschiedener Ansätze
 - Metaheuristiken
 - Kombinationen dieser
 - Kombination mit bereits implementierten Ansätzen
- Auswertung und Vergleich der Ansätze
- Anpassen des User Interfaces
- Auswahl des/der best passenden Umsetzungen
- Übertragen in eine finale Tour4Me Anwendung (vollständig c++ oder #)

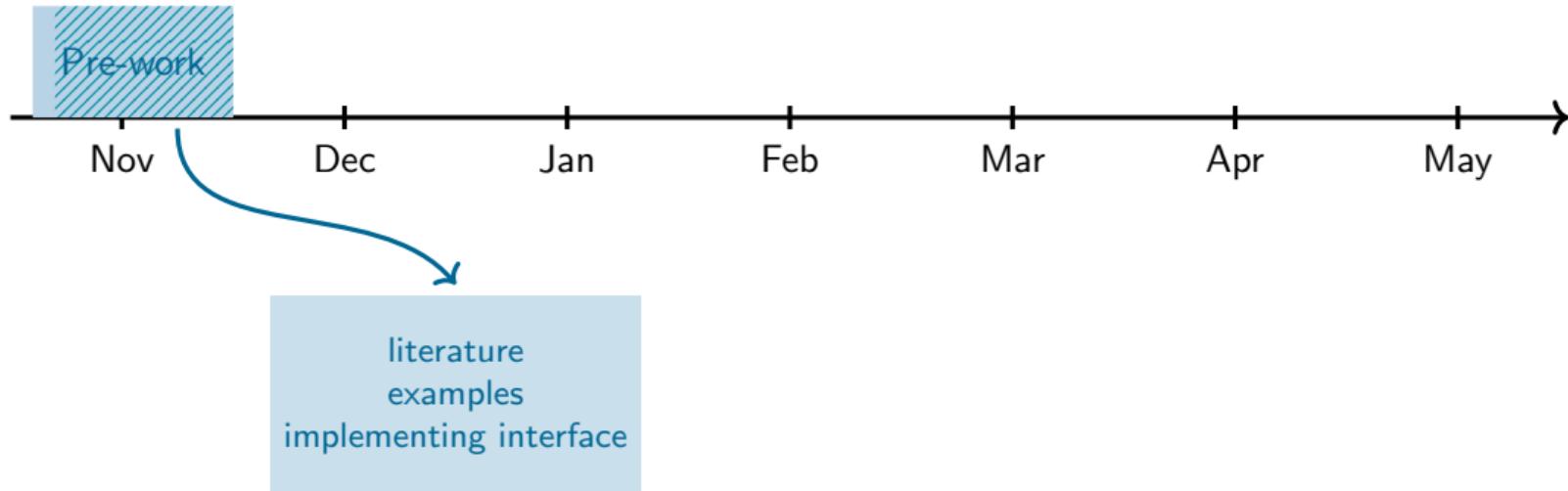
Timeline



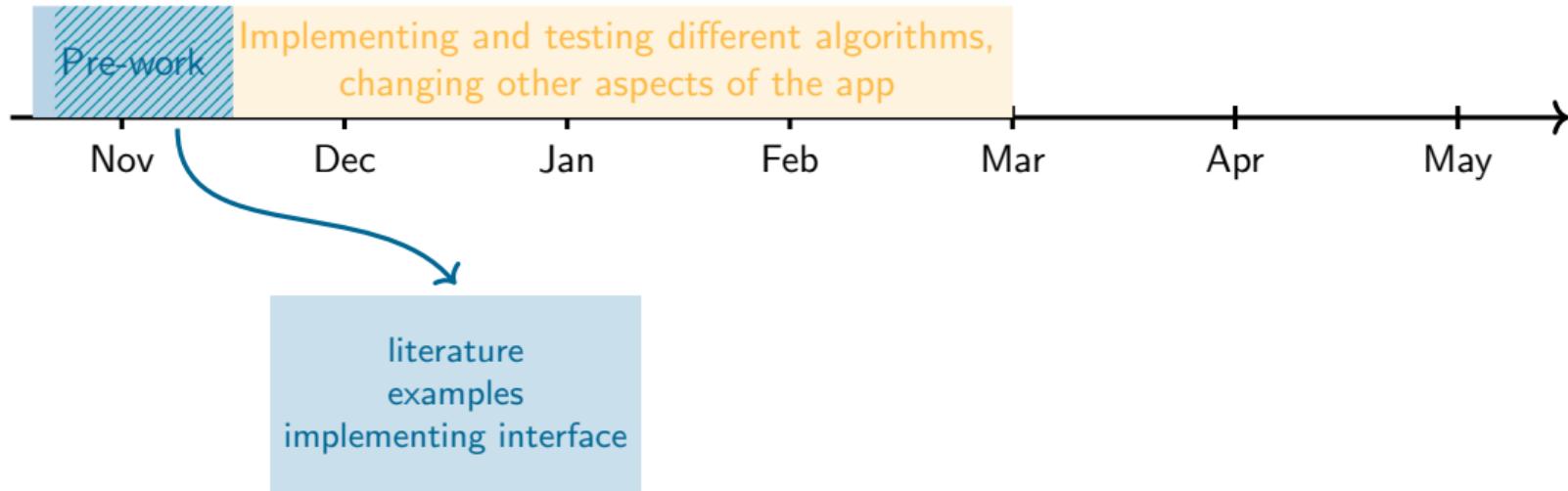
Timeline



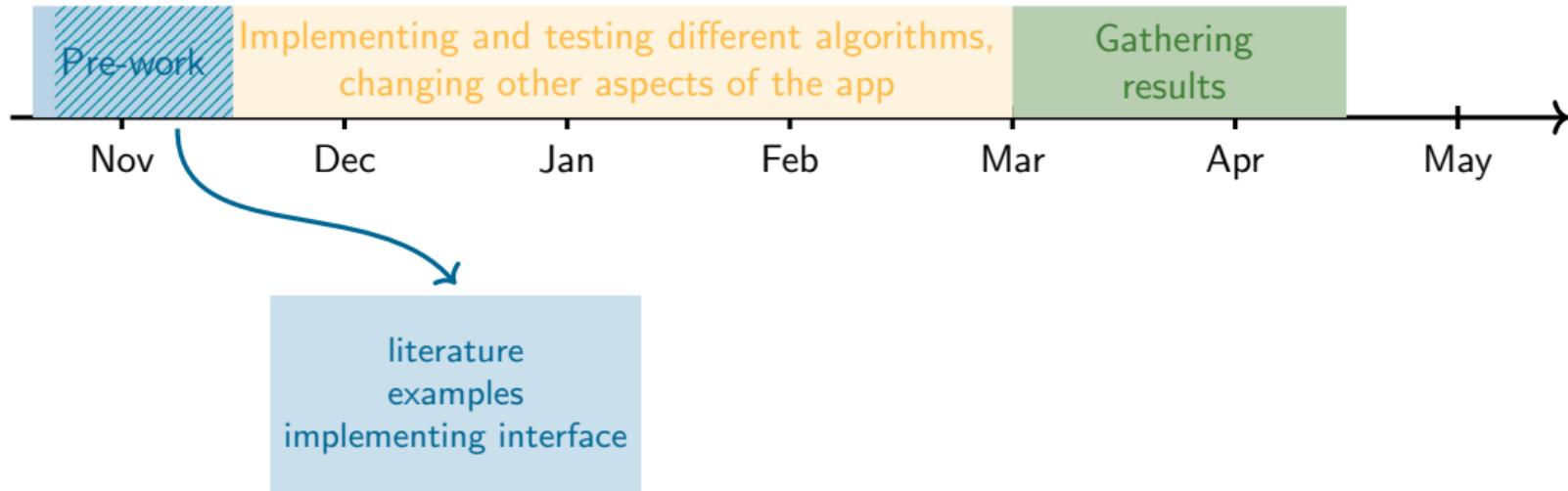
Timeline



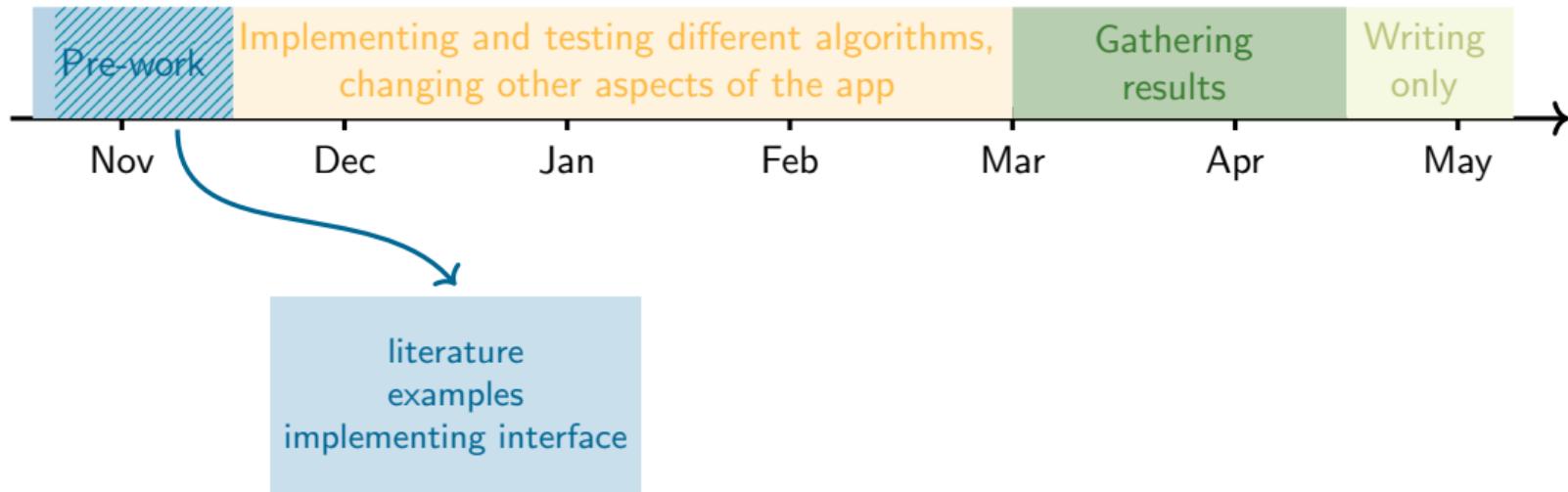
Timeline



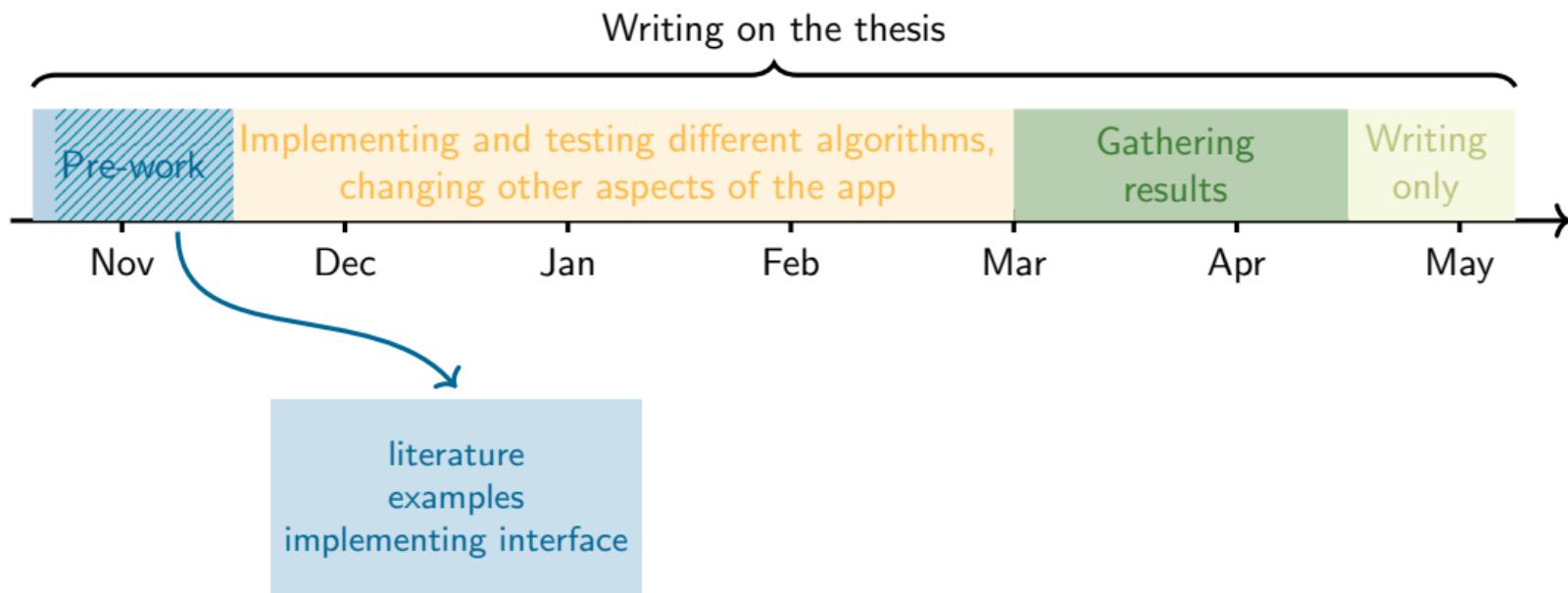
Timeline



Timeline



Timeline



Literatur |

- [1] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. 2016. Route Planning in Transportation Networks. In *Algorithm Engineering: Selected Results and Surveys*, Lasse Kliemann and Peter Sanders (Eds.). Springer International Publishing, Cham, 19–80. https://doi.org/10.1007/978-3-319-49487-6_2
- [2] Olli Bräsy and Michel Gendreau. 2005. Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science* 39, 1 (Feb. 2005), 104–118. <https://doi.org/10.1287/trsc.1030.0056>
- [3] Boris V. Cherkassky, Andrew V. Goldberg, and Tomasz Radzik. 1996. Shortest Paths Algorithms: Theory and Experimental Evaluation. *Mathematical Programming* 73, 2 (May 1996), 129–174. <https://doi.org/10.1007/BF02592101>
- [4] Daniel Delling. 2011. Time-Dependent SHARC-Routing. *Algorithmica* 60, 1 (May 2011), 60–94. <https://doi.org/10.1007/s00453-009-9341-0>
- [5] Daniel Delling, Andrew V. Goldberg, Thomas Pajor, and Renato F. Werneck. 2017. Customizable Route Planning in Road Networks. *Transportation Science* 51, 2 (May 2017), 566–591. <https://doi.org/10.1287/trsc.2014.0579>
- [6] Daniel Delling, Thomas Pajor, and Renato F. Werneck. 2015. Round-Based Public Transit Routing. *Transportation Science* 49, 3 (Aug. 2015), 591–604. <https://doi.org/10.1287/trsc.2014.0534>
- [7] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. 2009. Engineering Route Planning Algorithms. In *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*, Jürgen Lerner, Dorothea Wagner, and Katharina A. Zweig (Eds.). Springer, Berlin, Heidelberg, 117–139. https://doi.org/10.1007/978-3-642-02094-0_7
- [8] Andreas Gemsa, Thomas Pajor, Dorothea Wagner, and Tobias Zündorf. 2013. Efficient Computation of Jogging Routes. In *Experimental Algorithms (Lecture Notes in Computer Science)*, Vincenzo Bonifaci, Camil Demetrescu, and Alberto Marchetti-Spaccamela (Eds.). Springer, Berlin, Heidelberg, 272–283. https://doi.org/10.1007/978-3-642-38527-8_25
- [9] Michel Gendreau and Jean-Yves Potvin (Eds.). 2010. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, Vol. 146. Springer US, Boston, MA. <https://doi.org/10.1007/978-1-4419-1665-5>
- [10] Stefan Irnich, Birger Funke, and Tore Grünert. 2006. Sequential Search and Its Application to Vehicle-Routing Problems. *Computers & Operations Research* 33, 8 (Aug. 2006), 2405–2429. <https://doi.org/10.1016/j.cor.2005.02.020>

Literatur II

- [11] E. L. Lawler and D. E. Wood. 1966. Branch-and-Bound Methods: A Survey. *Operations Research* 14, 4 (Aug. 1966), 699–719. <https://doi.org/10.1287/opre.14.4.699>
- [12] Amgad Madkour, Waleed G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, and Saleh Basalamah. 2017. A Survey of Shortest-Path Algorithms. arXiv:1705.02044 [cs]
- [13] Gerhard Reinelt. 2003. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer, Berlin, Heidelberg.
- [14] Stefan Ropke. 2005. *Heuristic and Exact Algorithms for Vehicle Routing Problems*. Ph. D. Dissertation. Technical University of Denmark.
- [15] Christian Sommer. 2014. Shortest-Path Queries in Static Networks. *Comput. Surveys* 46, 4 (April 2014), 1–31. <https://doi.org/10.1145/2530531>
- [16] Muhammad Rhifky Wayahdi, Subhan Hafiz Nanda Ginting, and Dinur Syahputra. 2021. Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path. *International Journal of Advances in Data and Information Systems* 2, 1 (Feb. 2021), 45–52. <https://doi.org/10.25008/ijadis.v2i1.1206>

Beispiel Box

International Collegiate Programming Contest (ICPC)

- Association for Computing Machinery (ACM)
 - seit 1970
 - an Universitäten weltweit
-
- Teams von 3 Studierenden
 - 10 Probleme mit verschiedenem Schwierigkeitsgrad
 - 1 Computer pro Gruppe
 - Hilfsmittel: „Cheat Sheet“
 - Lösungen werden zu einem Judge Server hochgeladen
 - Gewinner: die Gruppe, welche die meisten Probleme gelöst hat

Beispiel column plus Box

- **11.10.** Einführungstreffen
- **18.10.** Systemeinführung
- **25.10.** Tipps und Tricks
- **08.11.** Datenstrukturen und Algorithmenentwurfsmethoden
- **15.11.** Such- und Sortieralgorithmen
- **22.11.** Dynamisch Programmieren
- **29.11.** Strings
- **06.12.** Übungswettbewerb 1
- **13.12.** Graphtraversierung
- **20.12.** Flussalgorithmen und Matchings
- **10.01.** Algorithmische Geometrie
- **17.01.** Übungswettbewerb 2
- **24.01.** Wintercontest oder Interner Wettbewerb

Wöchentliche Treffen
12:15 – ca. 13:30
Besprechung, Vortrag
13:30 – 15:45
Probleme lösen,
Hilfestellung

Beispiel Spalten mit Boxen

Kommunikation

- 1** Sprache
- 2** Stimme & Körpersprache
- 3** Einfachheit & Prägranz, Zeit

Beispiel Spalten mit Boxen

Kommunikation

- 1 Sprache
- 2 Stimme & Körpersprache
- 3 Einfachheit & Prägranz, Zeit

Methodik

- 1 Struktur
- 2 Stimulanz
- 3 Medien- & Materialeinsatz
- 4 Interaktion

Beispiel Spalten mit Boxen

Kommunikation

- 1 Sprache
- 2 Stimme & Körpersprache
- 3 Einfachheit & Prägranz, Zeit

Methodik

- 1 Struktur
- 2 Stimulanz
- 3 Medien- & Materialeinsatz
- 4 Interaktion

Vortragsqualität

- 1 Korrektheit & Technische Tiefe
- 2 Beantwortung der Fragen
- 3 Präsentationsziel

Motivation beim Vortrag!

Thema 1: Datenstrukturen und Algorithmenentwurfsmethoden

- Listen
- Arrays
- Stacks
- Heaps
- Hashing
- Greedy
- Divide & Conquer
- Brute-Force
- Backtracking