

Masterseminar für das WT 2025

an der Professur für Allgemeine Betriebswirtschaftslehre,
insbesondere Business Analytics & Management Science
(Prof. Dr. Claudius Steinhardt)

Bei Interesse an einer Seminararbeit schreiben Sie bitte **bis 01.12.2024** eine E-Mail mit Ihrem vollständigen Namen, **zwei Themenpräferenzen** sowie Ihrer aktuellen Notenübersicht an matthias.soppert@unibw.de.

Die Reihenfolge Ihrer Anmeldungen spielt bei der Themenvergabe keine Rolle. Sie werden am 02.12.2024 über die Themenzuteilung informiert. Ihre verbindliche Zusage muss anschließend bis spätestens zum 05.12.2024 erfolgen. Ein **Kick-Off Termin** findet am 13.12.2024 im Zeitraum von 11:00 Uhr bis 12:00 Uhr **online** statt. Die Bearbeitungszeit (8 Wochen für ca. 12-seitige Seminararbeit und 2 Wochen für Erstellung einer 30-minütigen Präsentation) endet am 06.03.2025.

Gerne können Sie sich auch mit eigenen Themenvorschlägen an uns wenden.

Thema 1: Interpretierbare Optimierung

Themengebiet: Lineare und ganzzahlige Optimierung, Interpretierbarkeit

Dank der stetigen und rasanten Verbesserung von Software zur Lösung von Optimierungsmodellen können zunehmend große Optimierungsprobleme gelöst werden. Zwar erlaubt diese Entwicklung den verstärkten Einsatz für Problemstellungen in realen Anwendungen, jedoch wird diese zunehmend komplexere Optimierungssoftware für den Anwender oftmals als undurchsichtige Black Box wahrgenommen, was die Akzeptanz gegenüber der Lösungen seitens der Anwender mindert. Um dem entgegen zu wirken, wurde kürzlich ein Framework entwickelt (Goerigk und Hartisch, 2023), welches die Interpretierbarkeit von Optimierungsmodellen ermöglicht. Das Ziel der Masterseminararbeit ist es, dieses Framework zu erläutern und anhand eines selbst erdachten Beispiels zu demonstrieren. Die Umsetzung soll in Python erfolgen.

Zu verwendende Methodik und Software: Lineare und ganzzahlige Optimierung, Implementierung in Python (Solver für Optimierung: Gurobi)

Thema 2: Interpretierbares Dynamic Pricing

Themengebiet: Pricing & Revenue Management, Dynamic Pricing, Interpretable Machine Learning

Im Zuge sich stark verbreitender automatisierter Entscheidungsfindung in der Praxis und dem stetig wachsenden Einsatz von Machine Learning Modellen besteht die Notwendigkeit, die oftmals als undurchsichtige („Black Box“) wahrgenommenen Algorithmen für den Anwendenden besser interpretierbarer zu machen. Die Arbeit soll im Kontext eines Dynamic Pricing Problems Möglichkeiten für eine einfach interpretierbare Darstellung einer gegebenen (optimalen) Preisstrategie erörtern. Denkbar sind die Verwendung eines Entscheidungsbaums als approximatives Ersatzmodell der Preisstrategie und die Ermittlung sogenannter Shapley Values. Der entwickelte Ansatz soll in Python implementiert werden und anhand einer Rechenstudie exemplarisch demonstriert werden.

Zu verwendende Methodik und Software: Supervised Machine Learning, Implementierung in Python (Package für ML: sklearn)

Thema 3: Tourenplanung unter Berücksichtigung von Lieferoptionen

Themengebiet: Vehicle Routing Problem, lineare und ganzzahlige Optimierung, Large Neighborhood Search

Bei der Paketauslieferung wird zur Berechnung eines optimalen Tourenplans das Vehicle Routing Problem verwendet. Eine Ergänzung hierzu betrachtet das Problem unter der Annahme, dass Kunden zwei oder mehr mögliche Orte angeben können, wohin sie das Paket geliefert haben möchten (Lieferoptionen). Zum Beispiel kann ein Kunde seinen Wohnort, seine Arbeitsadresse und eine Packstation in der Nähe als mögliche Lieferoptionen angeben. Die Arbeit soll basierend auf einem wissenschaftlichen Paper zu diesem Thema ein mathematisches Optimierungsmodell in Python implementieren und anhand eines selbst erdachten Beispiels analysieren. Zudem soll die im betrachteten Paper vorgestellte Heuristik (Large Neighborhood Search) anhand eines Beispiels erklärt werden.

Zu verwendende Methodik und Software: Lineare und ganzzahlige Optimierung, Large Neighborhood Search, Implementierung in Python (Solver für Optimierung: Gurobi)

Thema 4: Pricing im Bereich Attended Home Delivery

Themengebiet: Pricing & Revenue Management, Attended Home Delivery, lineare und ganzzahlige Optimierung

Viele Paketauslieferungen setzen voraus, dass der Kunde zum Zeitpunkt der Lieferung zu Hause ist. Das führt dazu, dass Lieferanten dem Kunden schon bei der Bestellung verschiedene Zeitslots anbieten, zu denen der Kunde das Paket erhalten kann. Durch Preissteuerung möchte der Lieferant dem Kunden dabei einen Anreiz bieten, ein für den Lieferanten günstiges Zeitfenster auszuwählen. Die Arbeit soll ein mathematisches Optimierungsmodell zu diesem Thema aus der Literatur in Python implementieren und anhand eines selbst erdachten Beispiels analysieren.

Zu verwendende Methodik und Software: Lineare und ganzzahlige Optimierung, Modellierung, Implementierung in Python (Solver für Optimierung: Gurobi)

Thema 5: Zielkonflikte in Ridepooling-Systemen

Themengebiet: Dial-a-Ride Problem, lineare und ganzzahlige Optimierung

Praxisbeispiele wie MOIA in Hamburg zeigen, dass Ridepooling-Systeme helfen können, das Mobilitätsangebot nachhaltig zu erweitern. Dabei stehen oftmals monetäre Absichten des Anbieters sozialen bzw. ökologische Ansichten gegenüber. So werden bspw. im öffentlichen Nahverkehr Versorgungsziele definiert, um der breiten Öffentlichkeit einen Zugang zu Mobilität zu gewährleisten. Ziel dieser Arbeit ist es, ein aus der Literatur bekanntes mathematisches Optimierungsmodell zu erweitern, um verschiedene Zielkonflikte der Nachhaltigkeit zu analysieren. Hierfür soll ein exemplarisches Beispiel in Python implementiert und ausgewertet werden.

Zu verwendende Methodik und Software: Lineare und ganzzahlige Optimierung, Modellierung, Implementierung in Python (Solver für Optimierung: Gurobi)

Thema 6: Generierung von Benchmark-Instanzen für ein Ridepooling-System

Themengebiet: Instance Generator, On-Demand Transportation, Dial-a-Ride Problem

Häufig werden in der Literatur Benchmark-Instanzen und reale Daten genutzt, um Lösungsansätze zu evaluieren. Dabei ermöglichen Benchmarks reproduzierbare Ergebnisse, wohingegen sie oftmals die Komplexität spezifischer Problemstellungen nicht widerspiegeln können. Echtweltdaten, die bspw. von Praxispartnern bereitgestellt werden, bilden zwar die Realität gut ab, können jedoch oftmals nicht veröffentlicht werden. Eine Möglichkeit, die Vorteile beider Datentypen zu verknüpfen, ist ein Generator, der auf Grundlage von realen Daten reproduzierbare Instanzen erstellt, die öffentlich zugänglich sind. Ziel dieser Arbeit ist es, mithilfe dieses Generators, eigene Instanzen für ein Ridepooling-System zu erstellen und auf Grundlage eines Lösungsverfahrens diese mit bisherigen Benchmark-Instanzen zu vergleichen.

Zu verwendende Methodik und Software: Implementierung in Python (mit JSON), Lineare Optimierung (Solver für Optimierung: Gurobi)