

Cucina intelligente di Massimo

Intelligent Agent Systems

Ausarbeitung

für das **Center for Advanced Studies der**
Dualen Hochschule Baden-Württemberg

von

Jonathan Diebel, Jannis Kaniaros und Dario Nieddu

31. März 2025

Matrikelnummern 9883850, 5934448,

Kurs T3M40501

Dozenten Prof. Dr. Nathan Sudermann-Merx

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
1 Einleitung	1
2 Design	2
2.1 Agenten	2
2.1.1 Customer Agents	2
3 Mathematische Formulierung des Optimierungsproblems	4
4 Offene Fragen	8
5 Resultate	9

Abkürzungsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Restaurants sind Orte, an denen Menschen nicht nur gutes Essen genießen, sondern auch eine Auszeit vom Alltag nehmen können. Ob für ein romantisches Dinner, ein schnelles Mittagessen oder ein entspanntes Treffen mit Freunden – die Gastronomie bietet für jeden Anlass das passende Ambiente. Doch in einer Welt, die immer schneller wird und in der Zeit oft zur kostbarsten Ressource wird, stehen auch Restaurants vor der Herausforderung, sich an die Bedürfnisse ihrer Gäste anzupassen.

Ein neues Konzept revolutioniert nun das Restaurantlebnis: Kunden können bereits beim Betreten des Restaurants oder bei der Reservierung angeben, wie viel Zeit sie für ihren Besuch einplanen. Ob 30 Minuten für eine schnelle Mahlzeit oder zwei Stunden für ein ausgedehntes Dinner – dieses innovative Modell ermöglicht es, den Service individuell auf die Wünsche der Gäste abzustimmen. So wird nicht nur die Effizienz gesteigert, sondern auch das Restaurantlebnis persönlicher und flexibler gestaltet.

In dieser Arbeit wird das Konzept der zeitbasierten Reservierung genauer untersucht und ein optimiertes intelligentes Mutliagentensystem vorgestellt, das die Umsetzung in einem Restaurant, der *Cucina intelligente di Massimo*, unterstützt.



Abbildung 1.1: Cover

2 Design

2.1 Agenten

Im Modell werden drei verschiedene Agenten eingesetzt: *Customer Agents*, *Service Agents* sowie ein *Manager Agent*. Die Agenten sind in einem Multi-Agenten-System organisiert, in dem sie miteinander kommunizieren und sich gegenseitig beeinflussen.

2.1.1 Customer Agents

Customer Agents beschreiben eine Gruppe von Kunden, die ein Restaurant besuchen und eine Bestellung aufgeben wollen. Die Attribute der Agenten werden größtenteils randomisiert generiert. Dazu zählen die genaue Anzahl der Personen, die maximale Wartezeit sowie das ausgewählte Gericht.

Die Agenten werden über einen internen Status gesteuert, der den aktuellen Zustand des Agenten beschreibt. Die möglichen Zustände sind *Waiting for Service Agent*, *Waiting for food*, *Eating*, *Finished Eating*, *Rejected* und *Done*. Der Status wird durch die verschiedenen Aktionen des Agenten verändert.

Wichtig im Zuge dieses Konzepts ist, dass die Agenten das Restaurant nicht verlassen, sobald ihre Zeit abgelaufen ist. Stattdessen wird das Essen beendet, dafür sinkt jedoch die Bewertung.

Wenn die Customer Agents ihr Essen beendet haben, geben sie eine Bewertung ab. Diese Bewertung ist eine Funktion mit mehreren Zufallsfaktoren:

$$r = \text{round}(\max(r_{\min}, \min(r_{\max}, r_{\max} - \alpha \cdot \text{exceedance} - \beta \cdot \text{error} + \gamma)), 2)$$

wobei:

- r die finale Bewertung ist,
- r_{\min} die minimal mögliche Bewertung ist,
- r_{\max} die maximal mögliche Bewertung ist,
- α die Gewichtung der Wartezeitstrafe ist,

- *exceedance* der Quotient aus der tatsächlichen Wartezeit und der angegebenen maximalen Zeit ist (nur bei Überschreitung, sonst 0),
- β die Gewichtung für fehlerhafte Bestellungen ist,
- *error* eine zufällige Fehlerquote ist,
- γ eine zufällige Bewertungsvariabilität abhängig von der Gruppengröße ist.

Im *PEAS-Framework* lassen sich die Customer Agents wie folgt beschreiben:

Performance Measure Einhaltung der maximalen Zeit

Environment Das Restaurant, die Service Agents

Actuators Bestellung aufgeben, Essen bewerten

Sensors Bewertung des Essens, Wartezeit

Bei den Customer Agents handelt es sich gemäß des *AIMA-Frameworks* um *Simple Reflex Agents*, da sie ihre Aktionen basierend auf dem aktuellen Zustand und den wahrgenommenen Informationen ausführen, ohne eine interne Zustandsrepräsentation der Welt zu verwenden. Sie reagieren direkt auf die wahrgenommenen Reize, wie z.B. die erhaltene Bestellung, um ihre nächsten Aktionen zu bestimmen.

3 Mathematische Formulierung des Optimierungsproblems

Wir haben ein engagiertes Team von Mitarbeitern und eine Reihe von Einschränkungen, die sich vor allem aus dem Arbeitsschutzgesetz ergeben. Einige Mitarbeiter arbeiten weniger Stunden und Schichten, während andere mehr arbeiten. Das muss nicht ausgewogen oder fair sein, sondern muss bestmöglich zur Zielfunktion beitragen. Die Herausforderung besteht darin, die Mitarbeiter für den nächsten Tag so einzusetzen, dass der Gesamtgewinn maximiert wird. Die Mitarbeiter werden im Laufe des Tages weder versetzt noch ausgewechselt. Darüber hinaus hat jeder Mitarbeiter einen spezifischen `customer_capacity`-Faktor, der seine Effizienz oder sein Qualifikationsniveau darstellt – je höher dieser Faktor, desto höher das Gehalt des Mitarbeiters.

Nachfolgend wird das Optimierungsproblems der Schichtplanung mathematisch mit allen erforderlichen Entscheidungsvariablen, Parametern, Zielfunktionen und Nebenbedingungen definiert.

Entscheidungsvariablen

Sei:

- $x_{a,t} \in \{0, 1\}$

Eine binäre Variable, die angibt, ob der Service-Agent a im Zeitfenster t arbeitet (1 = arbeitet, 0 = arbeitet nicht).

- $y_{a,s} \in \{0, 1\}$

Eine binäre Variable, die angibt, ob der Service-Agent a der Schicht s zugewiesen wurde (1 = zugewiesen, 0 = nicht zugewiesen).

- $z_{a,s} \in T$

Eine kontinuierliche Variable, die die Länge der Schicht s für den Service-Agenten a in Zeitschritten $t \in T$ darstellt.

Parameter

Sei:

- c_a
Gehalt pro Zeitschritt für den Service-Agenten a .
- p_a
Kapazität des Service-Agenten a . Gibt an, wie viele Kunden er gleichzeitig in einem Zeitschritt bedienen kann.
- v_t
Prognostizierte Anzahl an Besuchern im Zeitfenster t .
- $L_a = 36$
Maximale Länge einer einzelnen Schicht für den Service-Agenten a . (36 Schritte = 6 Stunden).
- $P_a = 6$
Minimale Pausenzeit zwischen zwei Schichten eines Service-Agenten (6 Schritte = 1 Stunde).
- $M_a = 48$
Maximale Anzahl an Zeitschritten (10-Minuten-Blöcke), die ein Service-Agent a pro Tag arbeiten darf (48 Schritte = 8 Stunden).
- $S_a = 3$
Maximale Anzahl an Schichten, die ein Service-Agent a pro Tag arbeiten darf.
- $T = \{1, 2, \dots, 144\}$
Menge aller Zeitfenster eines Tages, wobei jedes Zeitfenster eine Dauer von 10 Minuten hat (144 Zeitfenster für einen Zeitraum von 24 Stunden).

Zielfunktion

Das Ziel ist es, den Gewinn des Restaurants zu maximieren, indem die Kundennachfrage erfüllt und gleichzeitig die Personalkosten minimiert werden. Die Zielfunktion lautet:

$$\text{Maximiere } f(x) = \sum_{t \in T} \min \left(\sum_{a \in A} p_a x_{a,t}, v_t \right) - \sum_{t \in T} \sum_{a \in A} c_a x_{a,t}$$

Dabei:

- Der erste Term repräsentiert den Gesamterlös basierend auf der Anzahl der bedienten Kunden. Dieser wird durch die prognostizierte Besuchernachfrage (v_t) und die Gesamtkapazität der arbeitenden Agenten ($p_a x_{a,t}$) begrenzt.
- Der zweite Term repräsentiert die gesamten Gehaltskosten für alle arbeitenden Agenten.

Nebenbedingungen

1. Erfüllung der Nachfrage

Die Gesamtkapazität der zugewiesenen Agenten muss mindestens so groß sein wie die Besuchernachfrage in jedem Zeitfenster:

$$\sum_{a \in A} p_a x_{a,t} \geq v_t, \quad \forall t \in T$$

2. Maximale Arbeitszeit pro Agent

Jeder Agent darf maximal $M_a = 48$ Zeitschritte pro Tag arbeiten:

$$\sum_{t \in T} x_{a,t} \leq M_a, \quad \forall a \in A$$

3. Schichtlänge

Jede Schicht muss zwischen mindestens 6 und maximal 36 Zeitschritten liegen. Es werden Hilfsvariablen ($y_{a,s}$) verwendet, um die Schichten zu identifizieren, und deren Längen ($z_{a,s}$) zu berechnen:

- Wenn ein Agent kontinuierlich über aufeinanderfolgende Zeitschritte arbeitet, bildet dies eine Schicht.
- Nebenbedingung zur Schichtlänge:

$$6y_{a,s} \leq z_{a,s} = |\{t : x_{a,t} = 1, t \text{ gehört zur Schicht } s\}| \leq 36y_{a,s}, \quad \forall a \in A, \forall s$$

4. Maximale Anzahl an Schichten

Jeder Agent darf maximal $S_a = 3$ Schichten pro Tag übernehmen:

$$\sum_{s=1}^{S_a} y_{a,s} \leq S_a, \quad \forall a \in A$$

5. Mindestpausenzeit zwischen zwei Schichten

Zwischen zwei aufeinanderfolgenden Schichten eines Agents muss eine Pause von mindestens $P_a = 6$ Zeitschritten liegen:

$$t_{\text{Ende},s} + P_a \leq t_{\text{Start},s+1}, \quad \forall s, \forall a$$

6. Schichtbeginn und -ende nur zu vollen Stunden

Jede Schicht muss zu einem vollen Stundenbeginn starten und enden (alle sechs Schritte):

$$t_{\text{Start},s}, t_{\text{Ende},s} \mod 6 = 0, \quad \forall s, \forall a$$

7. Binäre Entscheidungsvariablen

Die Variablen müssen binär sein:

$$x_{a,t}, y_{a,s} \in \{0, 1\}, \quad \forall a, t, s$$

Diese mathematische Formulierung deckt die Anforderungen des Problems ab und dient als präzise Grundlage für die Implementierung eines Optimierungsmodells zur Schichtplanung in Python.

4 Offene Fragen

5 Resultate