

Studiengang Informatik
„Künstliche Intelligent“,
WS2021-22
Prof. Dr. Jörg Homberger

Projekt Dokumentation

- KI-Gruppe Vino -

vorgelegt von:

Fahretdin Okutan

Matrikelnummer: 1002336

Vilian Papazov

Matrikelnummer: 1000160

Vajos Amaranditis

Matrikelnummer: 1001217

Mouhamad Masri

Matrikelnummer: 380038

vorgelegt am: 21.01.2022

Inhaltsverzeichnis

1	Einleitung	2
2	Vorgehensweise	2
3	Algorithmus.....	2
4	Auswertung.....	4
5	Fazit	5

1 Einleitung

Das Ziel der Erstellung des Agentensystems ist es für Agenten A und B eine Minimierung der Rüst- beziehungsweise Fertigungskosten zu erreichen. Dabei müssen sich beide Agenten auf eine Reihenfolge, welche die Kosten bestimmt, einigen.

Für die Programmierung des Systems wählten wir Java und entwickelten das bereits zur Verfügung gestellte Programm weiter.

2 Vorgehensweise

Um die Kosten zu bestimmen, gab der derzeitige Mediator die Reihenfolge der Sequence $\{1,2,3,4,5\dots 200\}$ an. Daraufhin wurde genau eine Zahl mit einer anderen vertauscht. Beispielsweise wurde für die neue Reihenfolge $\{2,1,3,4,5,6\dots 200\}$ das Arrayelement mit dem Index 0 mit dem Element mit dem Element des Index 1 vertauscht.

Das Agentensystem soll mit der vorgegebenen Mengengröße 200 arbeiten. Bei einer Menge dieser Größenordnung, anders als bei kleineren Sequenzen wie beispielsweise $M = \{1,2,3,4\}$, welche in der Vorlesung mit beschriebenem Lösungsweg durch Tauschen der Elemente bereits zu 24 möglichen Anordnungen führte, ist es aus Gründen der Rechnerperformance und der Berechnungsdauer nicht möglich ist, eine optimierte Lösung auf diese Weise zu berechnen oder alle 200 Permutationsreihenfolgen zu betrachten.

Der Versuch ein System zu entwickeln, bei welchem der Mediator genau eine zufällige Permutation der Reihenfolge zurückgibt, versprach uns kein ausreichendes Ergebnis.

Daraufhin entwickelten wir einen heuristischen Algorithmus, um ein möglichst besseres Ergebnis mit geringer Rechenzeit zu erhalten. Dabei war uns bewusst, dass hierbei nicht die optimale Lösung für das Agentensystem entsteht.

3 Algorithmus

Schritt 1-) Suche des kleinsten, nicht besuchten Wertes

Beginn: `costMatrix[0]`

Der Algorithmus beginnt bei Index 0 und iteriert über das Array. Dabei wird der kleinste, noch nicht besuchte Wert gesucht.

Als Beispiel zur weiteren Erklärung gehen wir davon aus, dass an der Indexstelle 47 der kleinste, noch nicht besuchte Wert gefunden wurde.

Schritt 2:

Gefundenen Wert als besucht setzen

```
costMatrix[0][47]
```

Da der gesuchte Wert an der Stelle Index = 47 gefunden wurde, wird dieser nun als besucht gesetzt.

```
besucht[47] = true
```

Schritt 3:

Abspeichern des Indexwertes

```
sequence.add(47)
```

Daraufhin wird der Wert des Index als nächster Wert für die Sequence mittels `sequence.add()` abgespeichert.

Schritt 4:

Suche des nächsten, kleinsten, noch nicht besuchten Wertes und Wiederholung

```
costMatrix[47]
```

Beginnend bei Index = 47 wird in unserem Beispiel erneut über das Array iteriert und der nächste kleinste, noch nicht besuchte Wert gesucht. Daraufhin folgend werden die Schritte 2 - 4 wiederholt, bis alle Werte des Arrays besucht wurden.

```
1  ArrayList<ArrayList<Integer>> all_sequences = new ArrayList<>();
2  for(b = 0 ; b< getContractSize();b++){
3      boolean[] visited = new boolean[getContractSize()];
4      ArrayList<Integer> result = new ArrayList<Integer>();
5      int i = b;
6      int j = 0;
7      result.add(i);
8      visited[i]=true;
9      while(!EverythingVisited(visited)){
10
11          j = getIndexOfSmallestValueNotVisited(costMatrix[i],visited);
12          i = j;
13          result.add(i);
14          visited[i] = true;
15      }
16      all_sequences.add(result);
17  }
```

Abbildung 1 [Algorithmus Ausschnitt]

Schritt 5:

Zuletzt wird aus allen gefundenen Sequences die kleinste Sequenz genommen.

4 Auswertung

Jeder der beiden Agenten besitzt nun eine einzige Sequence, welche er als die Kleinste ansieht und von den jeweiligen Agenten für die Verhandlungen verwendet werden darf.

Dabei entspricht ProposalA der von Agent A gewünschten Sequence, während ProposalB der von Agent B gewünschten Sequence entspricht.

	ProposalA	ProposalB
Agent A	461	10564
Agent B	9520	6991

Tabelle 1 Erreichte Werte der jeweiligen Agenten in Bezug auf ProposalA und ProposalB

Um bei den Verhandlungen der Agenten eine Einigung zu erzielen, muss einer der Agenten eine Verschlechterung akzeptieren, während der jeweils andere Agent von einem möglichen Rabatt profitiert. Dennoch muss es für beide Agenten rentabel sein.

Daher bieten die Agenten gegenseitig Rabatte an, um ihre eigene Sequence zu behalten. Dabei entstehen folgende Anzahlen an Zugeständnissen der beiden Agenten.

Agent A	Agent B
10103	2529

Tabelle 2 Maximale Anzahl Zugeständnisse

Agent A: $10564 - 461 = 10103$

Agent B: $9520 - 6991 = 2529$

Beide Agenten bieten einander abwechselnd ihre Proposals mit den zusätzlichen Rabatten an. Dabei erreicht Agent A einen Wert von 2990 (461 mit Proposal A, zuzüglich 2530 Zugeständnisse), während Agent B einen Wert von 6989 erzielt (9520 mit Proposal A, abzüglich 2530 Rabatt).

Agent A: Contract: $461 + 2530 \text{ Zugeständnis} = 2991$

Agent B: Contract: $9520 - 2530 \text{ Rabatt} = 6990$

5 Fazit

Insgesamt erreichen die Agenten einen Wert von 9981. Dabei wird das ProposalA von Agent A mit Zugeständnissen angenommen.

Während den Verhandlungen ist es für Agent A unersichtlich, ab welchem Wert Agent B dem Angebot zustimmen wird. Agent B könnte weiterhin Angebote ablehnen und die Verhandlungen dadurch verlängern, um dem bestmöglichen Proposal zuzustimmen. Jedoch besteht dabei das Risiko, dass Agent B häufiger als 2529-mal zustimmen muss und einen eigenen Verlust riskiert. Um dieses Risiko zu umgehen, stimmt er dem ProposalA von Agent A zu.