

4. Labor zur

## Künstliche Intelligenz

GRUPPENÜBUNGEN:

### (G 11) Lokale Suchalgorithmen

Generieren Sie eine große Anzahl von  $8 \times 8$  Puzzle und 8-Damen Problem Instanzen und lösen Sie diese Probleme (falls möglich) mit Hilfe verschiedener lokalen Suchalgorithmen (Bergsteiger Algorithmus, Bergsteiger mit random restart und simuliertes Abkühlen). Messen Sie die Kosten jeder Suche und Prozentsatz der gelösten Probleme und erstellen Sie eine graphische Darstellung dieser Werte gegen die optimalen Kosten. Kommentieren Sie Ihre Ergebnisse.

### (G 12) Suchprobleme lösen

Gegeben sei das Problem des kürzesten Weges zwischen zwei Punkte in der Ebene aus Lab 3. Wir verändern dieses Problem wie folgt:

- Gegeben ist jetzt ein Agent dessen Umgebungswahrnehmung durch und eine Positionsliste der sichtbaren Kanten der Polygone, relativ zu seiner Position besteht. Diese Umgebungswahrnehmung beinhaltet aber nicht die Position des Agenten in der Ebene!!! Der Roboter muss seine Position von der Karte erlernen oder berechnen können. Für den Anfang können wir annehmen, dass jede Lage des Roboters eine verschiedene Sicht hat.
  - Jede Aktion ist ein Vektor, der den geradlinigen Weg des Roboters beschreibt. Falls der Weg keine Hindernisse hat (d.h. keine Polygone überschneidet), ist die Aktion erfolgreich; falls nicht, stoppt der Roboter an dem Punkt wo sein Weg auf ein Hinderniss trifft. Falls der Agent sich nicht mehr bewegt und den Zielpunkt erreicht hat, teleportiert die Umgebung den Agenten auf eine zufällige Position außerhalb der Polygone.
  - Das Leistungsmaß vergibt dem Agenten 1 Punkt für jede Distanzeinheit, die er durchläuft und 1000 Punkte beim Erreichen des Zielpunktes.
- a) Implementieren Sie diese Umgebung und ein Agent für das Lösen des Problems nach den obigen Spezifikationen. Nach jeder Teleportierung muss der Agent ein neues Problem formulieren, inklusive die Entdeckung seiner Position in der Ebene.
  - b) Messen Sie die Leistung ihres Agenten und berichten Sie diese Leistung über 100 Episoden.
  - c) Verändern Sie die Umgebung, so dass in 30 Prozent der Fälle ihr Agent ein ungewünschtes Ziel erreicht. Dieses Ziel wird zufällig aus der Menge der sichtbaren Kanten gewählt. Falls keine sichtbaren Kanten existieren, dann bewegt sich der Agent überhaupt nicht. Dies ist ein Modell der Bewegungsfehlern eines realen Roboters. Ändern Sie ihren Agenten, so dass bei der Entdeckung eines Fehlers, er sofort seine Position berechnet und ein Plan erstellt, um zu seiner früheren Position zurückzukehren und von dort weiter, in der richtigen Richtung, zu machen. Vorsicht, manchmal kann auch diese Aktion fehlerhaft sein! Geben Sie ein Beispiel eines Agenten, der zweimal hintereinander einen Bewegungsfehler macht und trotzdem sein Ziel erreicht.

d) Gegeben die folgenden zwei Fehlerbehebungsmethoden:

1. Suche den am nächsten positionierten Knoten auf der Ursprungsrouten;
2. Bilde eine neue Route zum Ziel, von der jetzigen Position ausgehend.

Vergleichen Sie die Leistung der drei obengenannten Fehlerbehebungsmethoden. Wie wird diese Leistung von den Suchkosten beeinflusst?

e) Angenommen, es gibt Positionen mit identischer Wahrnehmung (z.B. die Welt ist ein Gitter mit quadratischen Hindernissen). Wie sehen jetzt die Lösungen aus?