1 Problembeschreibung und -analyse

1.1 Problembeschreibung

Die Hauptschwierigkeit des Problems liegt daran, dass es sich eigentlich um eine Kombination aus (mindestens) zwei Problemen handelen. Jedes Problem für sich ist gut beschrieben und gelöst worden, doch für alles gleichzeitig nicht.

1. Graph coloring - Visit stuff
2. Evacuate

1.2 Annahmen über das Problem

- Roboter sind punktförmig
- Können in der Bewegung saugen
- Keine Beschränkung, wie viele Roboter sich in einem Punkt befinden
- Kommunikation ist instantan und ohne Berechnungszeit (Senden wie Empfangen)

1.3 Disclaimer

Einschränkungen nicht simuliert, aber deren Auswirkungen

2 Problembetrachtung

Das Problem kann unter vielen verschiedenen betrachtet werden. Je nachdem, in welcher Domäne der Informatik es eingeordnet wird, gibt es unterschiedliche Lösungsansätze.

In diesem Abschnitt wird kurz beschrieben,

2.1 Graphenproblem

Suche

2.2 Optimierungsproblem

2.3 Künstliche Intelligenz

3 Umgebung

Die Modellierung des Meeresbodens zum Ziel, die folgenden Fragen zu lösen oder einen guten Kompromiss zu finden:

1. Wie lassen sich Kreise auf einer Ebene anorden, sodass die nicht von Kreisen bedeckte Fläche minimal ist (vergleichbar mit dem Ausstechen von Kreisen aus Keksteig)

2. Wie lässt sich der Meeresboden unterteilen, sodass eine Simulation möglichst einfach wird?

Geplant ist, dass die Missionsdauer in Zeitschritt von 1s aufgeteilt wird. In jedem Zeitschritt bewegen sich die Roboter einen geometrishen Schritt auf dem Meeresboden weiter und säubern ihn dabei von Manganknollen. Der Vorteil darin besteht, das die Optimierung der Ausbeute darin besteht, auszuwählen, welche Zelle als nächstes besucht wird, sodass möglichst viele verschiedene Zellen gesaugt werden.

Dazu werden unendlichen Weiten des pazifischen Meeresgrundes in Abschnitte in Form von Polygonen eingeteilt (parkettiert), damit der Wertebereich der nächsten möglichen Schritte diskretisiert wird.

Die Wahl, mit welchem Polygon gearbeitet wird, hat direkte Auswirkungen auf Genauigkeit und Leistungsfähigkeit der Simulation, wie im Folgenden gezeigt wird.

Es gibt nur drei verschiedene Polygone, mit denen gleichmäßig ohne Zuhilfenahme von Füllstücken parkettiert werden kann 1 [1].

Der Meeresboden kann dann als Graph gesehen werden, bei denen Knoten als Zellen gesehen werden und geometrisch einem Polygon entsprechen. Angeordnet werden diese so, dass zwischen den Schwerpunkten zweier benachbarter Zellen exakt 1m Abstand ist. Dies hat den Grund, dass ein Roboter in jedem Zeitschritt in die Mitte der nächste Zelle wechseln kann.

Der Zusammenhang zwischen realer Umgebung und Modellierung kann in Fig. 2

Roboter werden so simuliert, dass die Bewegungen nur über die Kanten einer Zelle möglich sind. Daher ist Anzahl der Ecken identisch mit den möglichen Bewegungsrichtungen. Somit gilt: je mehr Ecken, desto mehr Freiheitsgrade. Die Kanten geben an, von welchem Knoten zu welchen Nachbarn gewechselt werden kann. Somit hat jeder Knoten auch so viele Kanten wie das gewählte Polygon Ecken hat.

Da eine Simulation mit unterschiedlichen Polygonen unnötig komplex wird, entscheidet es sich zwischen gleichschenkligem Dreieck, Quadrat und regelmäßigem Sechseck.

Der Arbeitsbereich eines Roboters ist kreisförmig, die Zellen, in denen er sich befindet, jedoch ein Polygon. Daher gibt es in den Ecken der Zelle Bereiche, die nicht (unmittelbar) gesaugt werden. Berechnen lässt sich der Verlust über das

[1] Source

Abbildung 1: Diskretisierung des Spielfeldes mittels verschiedener Polygone

Abbildung 2: Zusammenhang zwischen Meeresboden und Modellierung

Abbildung 3: Innenkreis und Polygon

Verhältnis von der Fläche des Innenkreises des Polygons zum Flächeninhalt des Polygons selbst.

Die folgende Tabelle zeigt, wie viel Prozent einer Zelle nicht gesaugt werden, je nachdem, welches Polygon gewählt wurde.

Die Auswahl viel danach leicht, da die Simulation mit einem Quadrat als Polygon folgende Vorteile bietet:

- Einfache Datenstruktur
- Einfaches Berechnen der Nachbarn
- Visualisierung entspricht pixeln, keine Umrechnung nötig
- Weniger Verlust als Dreieck
- Weniger Freiheitsgrade als Sechseck (weniger Auswahl bedeutet weniger Rechenaufwand)
- Einfache Berechnung von Entfernungen zwischen zwei Zellen 4.1

Abbildung 4: Fehlerrechnung

4 Theory Crafting

4.1 Manhattan-Metrik

4.1.1 Entfernung

4.1.2 Kreise

4.2 Platzierung

Keine Häufigkeitsverteilung gegeben

4.3 Missionsdauer

4.4 Finden des Sammelpunktes

4.5 Zeitbeschränkung

5 Implementierung

6 Diskussion

6.1 Fehlerrechnung

6.1.1 Abweichung vom Optimum

6.1.2 Fehler durch Modellierung