

Grundlagen der Künstlichen Intelligenz (CS 205)

Prof. Dr. M. Helmert
G. Röger
Frühjahrssemester 2012

Universität Basel
Fachbereich Informatik

Übungsblatt 4

Abgabe: 12. April 2013

Aufgabe 4.1 (2+1+3 Punkte)

Zeigen Sie:

- (a) A^* ohne Reopening kann mit einer zulässigen, inkonsistenten Heuristik zu suboptimalen Lösungen führen.
- (b) A^* kann mit einer konsistenten, nicht zielerkennenden Heuristik zu suboptimalen Lösungen führen.
- (c) A^* mit sicherer Heuristik h ist auch ohne Duplikatelimination semi-vollständig (d.h., es ist garantiert, dass A^* mit h eine Lösung findet, wenn eine Lösung existiert), wenn $cost(a) > 0$ für alle Aktionen a gilt. Die Zustandsräume sollen wie in der Vorlesung endlich sein.

Hinweis: Teil (a) und (b) kann man zeigen, indem man jeweils einen geeigneten Beispielszustandsraum und die zugehörigen Heuristikwerte angibt.

Aufgabe 4.2 (2+1+1)

Betrachten Sie das n -Damen-Problem, bei dem, analog zum 8-Damen-Problem aus der Vorlesung, n Damen so auf einem $n \times n$ -Spielfeld platziert werden sollen, dass sich keine Damen gegenseitig bedrohen (d.h. in der gleichen Zeile, Spalte oder Diagonale stehen).

- (a) Implementieren Sie eine Hillclimbing-Suche mit der auf Folie 8/18 (Printversion Kapitel 8) beschriebenen Heuristik und Nachbarschaft. Falls es in der Nachbarschaft mehrere beste Kandidaten gibt, dann wählen Sie zufällig einen davon (uniform, also alle mit gleicher Wahrscheinlichkeit). Die Anzahl der Damen n soll dabei ein Eingabeparameter sein, der initiale Zustand soll zufällig erstellt werden.

Die normale Hillclimbing-Suche terminiert, sobald der Heuristikwert in einem Schritt nicht verbessert werden kann. Lassen Sie die Suche für 5, 8 und 20 Damen jeweils 10 000 mal laufen und geben Sie für jede Problemgröße die folgenden Werte an:

- Anzahl gelöster Probleminstanzen
 - Durchschnittliche Anzahl von Schritten bei gelöster Probleminstanz
 - Durchschnittliche Anzahl von Schritten bei ungelöster Probleminstanz
- (b) Man kann die Erfolgsquote verbessern, indem man auch Schritte zulässt, bei denen der Heuristikwert gleich bleibt. Hier besteht aber die Gefahr, dass die Suche in eine Endlosschleife gerät (Warum?). Ändern Sie Ihre Suche aus Aufgabenteil (a) so, dass auch solche Schritte erlaubt sind, verlangen Sie aber, dass jeweils nach spätestens 100 Schritten eine Heuristikverbesserung eintritt.
- Wiederholen Sie Ihr Experiment aus Aufgabenteil (a) mit der neuen Suche und geben Sie die entsprechenden Zahlen an.
- (c) Eine andere Möglichkeit, die Erfolgsquote zu verbessern, ist es, die Suche im Falle des Scheiterns neu zu starten (also wieder einen zufälligen Zustand zu generieren). Verwenden Sie

weiterhin Ihre Version aus Aufgabenteil (b), erlauben Sie sie aber bei jedem der 10 000 Versuche bis zu 5 Neustarts, um eine Lösung zu finden. Die Anzahl der benötigten Schritte pro Lösung ist dann die Summe der Schritte in diesen Suchen.

Wiederholen Sie ihr Experiment mit dieser Modifikation und geben Sie die entsprechenden Zahlen an. Wie viele Neustarts werden jeweils durchschnittlich bei einer gefundenen Lösung benötigt?

Geben Sie bitte neben den Experimentergebnissen auch Ihren Code ab.

Die Übungsblätter dürfen in Gruppen von zwei Studierenden bearbeitet werden. Bitte schreiben Sie beide Namen auf Ihre Lösung.