Nachwuchsforschergruppe
The Smart Virtual Worker

Automatisierte Handlungsselektion durch Reinforcement Learning

Helge Ülo Dinkelbach

Aktionsparameterisierung

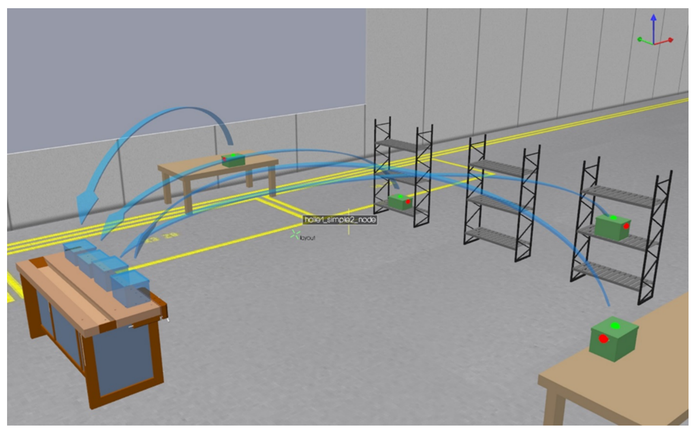
Aktuelle Ergonomietools benötigen einen vergleichsweise hohen Eingabeaufwand um einen Arbeitsprozess entsprechend abzubilden. Im Rahmen des Projektes wurde aus diesem Grund an Methoden gearbeitet, die diesen Arbeitsaufwand reduzieren. Dazu wurden eine Reihe von Basisbewegungen für den digitalen Arbeiter definiert, z. B. Laufen zu Objekten oder das Fügen von Objekten. Diese sind allgemein definiert und werden auf Basis der Aufgabendefinition in das mögliche Aktionsset des Agenten für das jeweilige Szenario überführt. Dieses Aktionsset dient als Basis für die Optimierung mittels Reinforcement Learning Methoden.

Aktionsvalidierung

Bei industriellen Arbeitsvorgängen dürfen gesetzliche Rahmenbedingungen nicht verletzt werden. Daher ist es notwendig, diese auch in einem digitalen Menschmodell entsprechend abzubilden. In der von uns vorgestellten Software geschieht dies durch eine Aktionsvalidierung. Bevor Aktionen ausgeführt werden, wird geprüft ob z. B. die maximale Traglast eingehalten wird. Ausserdem kann der Anwender im Zuge der Aufgabendefinition noch zusätzliche Bedingungen an Umweltfaktoren stellen, hier sei zum Beispiel die notwendige Mindesthelligkeit für eine Tätigkeit genannt. Werden diese Bedingungen nicht eingehalten, werden die entsprechenden Aktionen nicht vom Werker durchgeführt.

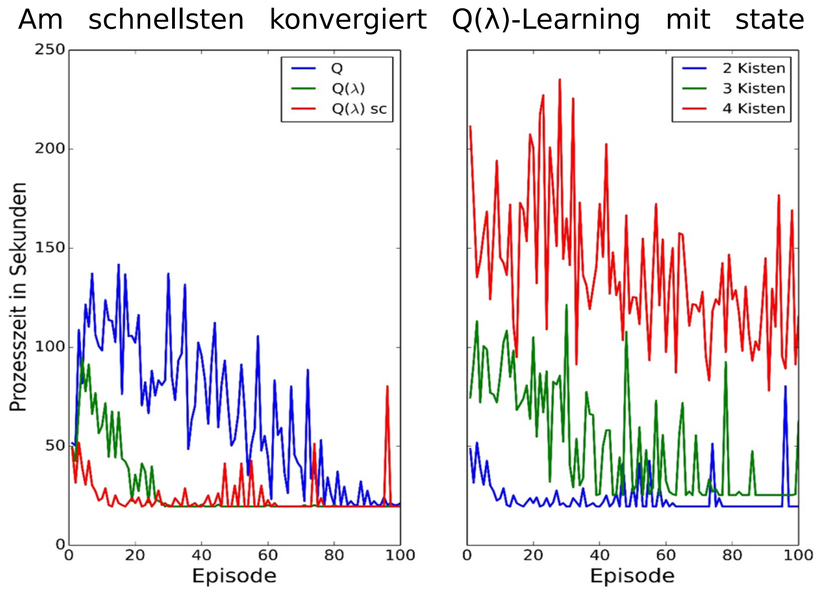
Vergleich der Methoden

Als Testszenario soll eine einfache Transportaufgabe betrachtet werden. Der Agent erhält die Aufgabe, eine Reihe von Gegenständen, die in der Werkhalle verstreut sind, an einem Ort zusammenzutragen. Konkret bedeutet das, dass in dieser Aufgabe 4 Objekte, die mit den oben beschriebenen Fähigkeiten kombiniert zu 16 Aktionen pro Zustand führen, auf eine gemeinsame Ablage geschafft werden sollen.



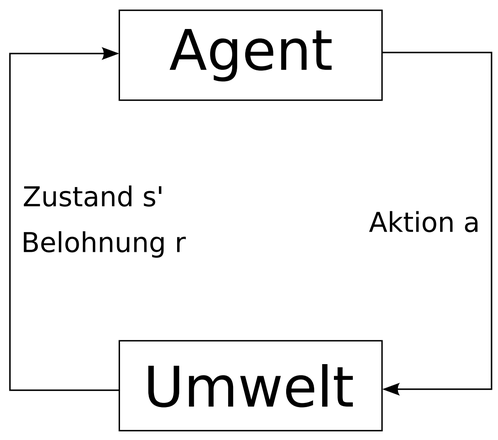
Ein Beispiel einer Umwelt des „Smart Virtual Worker“. Die blauen Pfeile markieren die Aufgabe des Arbeiters, hier die jeweiligen Objekte an die entsprechende Zielposition zu transportieren. Die autonome Handlungs-selektion muss eine Aktionsfolge generieren, so dass der Agent die Aufgabe z. B. mit minimaler Prozesszeit löst.

In unserem Vergleich haben wir mit jedem Algorithmus 100 Episoden simuliert und die benötigten Schritte bis zur Lösung der Aufgabe für jede Aufgabe bestimmt. Bei dieser Untersuchung wurde die niedrigste Prozesszeit als Optimierungskriterium verwendet. Links ist die Prozesszeit in Sekunden pro Episode für eine Transportaufgabe von 2 Kisten für die drei vorgestellten Verfahren abgebildet. Man kann erkennen, dass Q(λ)-Learning schneller als Q-Learning konvergiert. Am schnellsten konvergiert Q(λ)-Learning mit state conditions (sc). Auf der rechten Seite wurde das Q(λ)-Learning mit state conditions für Transportaufgaben von 2, 3 und 4 Kisten untersucht. Während für 2 und 3 Kisten 100 Episoden ausreichen, sind für 4 Kisten mehr als 100 Episoden notwendig.



Reinforcement Learning

Reinforcement Learning Methoden sind iterative Lernverfahren, mit dem Ziel, dass ein Agent eine möglichst optimale Strategie zur Lösung eines gestellten Problems findet, indem er zunächst das Handlungsspektrum und aus Aktionen folgende Konsequenzen durch Exploration erlernt. In jedem Simulationsschritt wird vom Agenten basierend auf seinem Zustand s eine Aktion a gewählt, die zu einem Folgezustand s' und einer Belohnung r führt. Für das einfache Optimierungskriterium der niedrigsten Prozesszeit kann die Belohnungsfunktion z.B. wie folgt definert werden:



$$r = \begin{cases} 0.0 & \text{falls Zielzustand erreicht} \\ -10.0 & \text{falls gewählte Aktion nicht durchführbar} \\ -t & \text{sonst} \end{cases}$$

Q-Learning und Erweiterungen

Es gibt verschiedene Verfahren um die Auswirkungen der Belohnung auf die Strategie des Agenten zu beschreiben. Beim Q-Learning wird für jedes Zustandsaktionspaar eine Bewertung Q(s,a) ermittelt, die in jedem Zeitschritt abhängig von der Belohnung angepasst wird. Die Änderung ΔQ(s,a) ergibt sich wie folgt:

$$\Delta Q(s,a) = \alpha(\underbrace{r + \gamma \max_a Q(s',a) - Q(s,a)}_{\delta})$$

Daraus ergibt sich für die Bewertung:

$$Q(s,a) = Q(s,a) + \Delta Q(s,a)$$

Das Q-Learning benötigt viele Episoden um ein stabiles Lernergebnis zu erhalten, da typischerweise nicht jede Aktion direkt bewertet wird, sondern die Bewertung erst später als Konsequenz einer Sequenz von Aktionen erfolgt. Weiter zurückliegende Aktionen werden erst über die Zeit durch Rückwärtspropagation der Bewertung aktualisiert. Die Idee des Q(λ)-Learning ist es nun, die erhaltene Belohnung nicht nur auf den aktuellen Zustand sondern auch auf frühere Zustände zu übertragen. Dies soll zu einer schnelleren Konvergenz des Lernens führen. Wird ein Zustanddurchlaufen, wird eine zusätzliche Variable (meist mit für „eligibility traces“ bezeichnet) inkrementiert. Diese e-Werte werden nun auch Bestandteil der Aktualisierung aller Q(s,a)-Werte:

$$Q(s,a) = Q(s,a) + \alpha e(s,a)\delta$$

Im Anschluss wird geprüft, ob die nächste gewählte Aktion der aktuellen greedy-Aktion entspricht. Wenn ja, wird e(s,a) exponentiell für alle Zustandsaktionspaare dekrementiert, wenn nicht, werden alle e(s,a) Werte auf zurückgesetzt. Dies hat den Effekt, dass solange der Agent der greedy-Strategie folgt, alle bisher gewählten Zustandsaktionspaare aktualisiertwerden. Beginnt der Agent zu explorieren, wird nur der aktuelle e(s,a)-Wert aktualisiert, in Analogie zu dem oben beschriebenen Q-Learning.
Der zweite Ansatz zur Beschleunigung des Q-Learning adressiert das Aktionsset direkt, indem unmögliche Aktionen von vornherein ausgeschlossen werden [4]. Zum Beispiel kann der Agent nur ein Objekt aufnehmen in dessen Nähe er sich befindet. Kann er nicht zu dem Objekt hinlangen, kann er es auch nicht aufnehmen – die Aktion „Aufnehmen“ ist also in diesem Zustand unmöglich. Da diese Einschränkungen zustandsabhängig sind, wird diese Erweiterung mit „state conditions“ bezeichnet.

TECHNISCHE UNIVERSITÄT CHEMNITZ
Interdisziplinäres Kompetenzzentrum Virtual Humans
funded by:
Gefördert aus Mitteln der Europäischen Union
Europa fördert Sachsen. ESF Europäischer Sozialfonds
SACHSEN