

Oktopus

Der Oktopus Algorithmus ist Teil des Taskmanagers und für die Positionierung von Offensiven Spielern verantwortlich. Ziel dabei ist es, dass die sekundären Angreifer sich freilaufen und anspielbar machen, um den Ball zu empfangen und somit einen Fortschritt Richtung Gegnertor schaffen. Der Algorithmus wurde erstmals beim RoboCup 2023 im Rahmen des Taskmanager_Bod23 eingesetzt.

Der Oktopus Algorithmus war Teil des eTDP 2024 (siehe Anhang links).

Motivation

Der Oktopus Algorithmus wurde speziell für die Zusammenarbeit mit Passbäume_Bordeaux_2023 entwickelt. Die Passbäume analysieren das Spielfeld und die Position der sekundären Angreifer, also Roboter, die zwar an der Offensive beteiligt jedoch nicht im Ballbesitz sind. Der Passbaum evaluiert mögliche Pässe, wobei auch die Abfangwahrscheinlichkeit der gegnerischen Roboter berücksichtigt wird. Aus diesen Pässen wird die bestmögliche Passkette ausgewählt, die auch aus mehreren Pässen in Folge bestehen kann. Da der Passbaum nicht nur den Roboter an sich, sondern auch dessen Position bestimmt, sind für die Roboter die in einer Passkette eingeplant sind keine weiteren Schritte zur Positionierung notwendig.

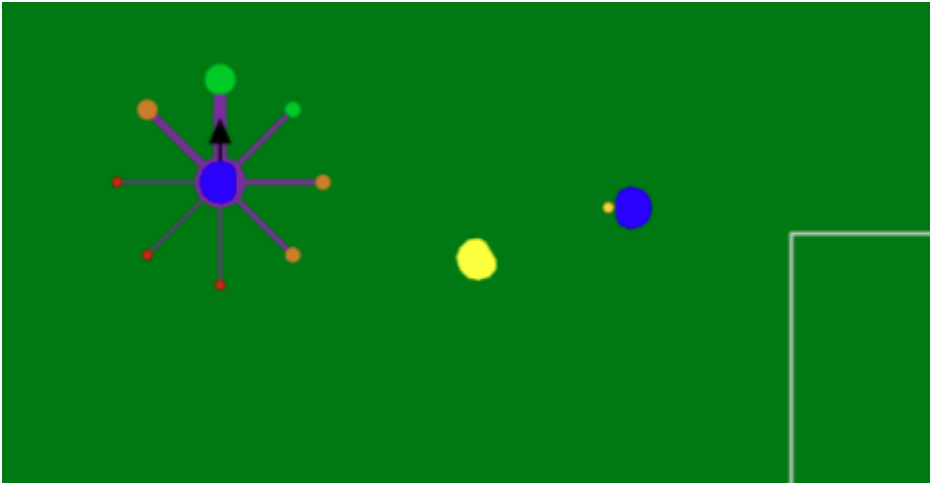
Anders sieht es allerdings mit Robotern aus, die nicht in einer Passkette berücksichtigt werden. Diese Roboter sind an ihrer aktuellen Position nicht gut genug positioniert und sollten sich bewegen, um die Position zu verbessern und nach Möglichkeit in zukünftigen Passketten miteinbezogen werden. Position verbessern heißt dabei, dass die Roboter einerseits sich freilaufen sollten, sodass sich keine Gegner auf der Passlinie befinden und andererseits in eine gute Schussposition finden sollten, sodass nach einem erfolgreichen Pass ein Torschuss unternommen werden kann.

Algorithmus

Der Oktopus-Algorithmus wurde aufgrund seiner Ähnlichkeit in der Visualisierung nach dem gleichnamigen Tier benannt. Wie ein echter Oktopus besitzt ein Oktobot acht Tentakel, mit denen er seine Umgebung abtastet. Dabei wird für jedes Tentakel an der Spitze ein Score berechnet, der sich aus der Tor-&Passwahrscheinlichkeit an dieser Position zusammensetzt. Diese werden miteinander multipliziert, um einen Gesamtscore zu erhalten. Außerdem kann noch ein Bias eingeführt werden, dass der Roboter seine aktuelle Bewegungsrichtung möglichst weiter behält, um einen häufigen Wechsel der Bewegungsrichtung zu verhindern. Dies kann sinnvoll sein, um

einen Hysterese-Effekt einzuführen, der instabile Zustände vermeidet, bei denen der Roboter sehr schnell zwischen zwei Zuständen oszilliert und damit effektiv stecken bleibt.

Nachdem für jedes Tentakel der Score berechnet wurde, kann das beste Tentakel ausgewählt und als Richtung gewählt werden. Da der Oktobot nicht den Score seiner aktuellen Position kennt, sondern nur Scores in seiner Umgebung, ist er dazu gezwungen sich zu bewegen. Dies ist sinnvoll, da der Passbaum evaluiert hat, dass die aktuelle Position nicht ideal ist.



Da der Oktobot stets versucht in eine Richtung zu fahren, die die Pass- und Torwahrscheinlichkeit verbessert, kann er sich prinzipiell komplett selbstständig bewegen und positionieren. Dies hat jedoch auch den Nachteil, dass evtl Positionen angefahren werden, an denen der Roboter seine Anspielbarkeit maximiert, dabei jedoch die Torwahrscheinlichkeit verschlechtert. Außerdem könnte es passieren, dass mehrere Oktobots dasselbe lokale Optimum anfahren. Das ist allgemein zu vermeiden, da sich die Roboter sonst gegenseitig behindern und eine bessere Verteilung der Roboter auch dazu führt, dass der Gegner seine Verteidiger aufteilen muss, wodurch Lücken frei werden, die ausgenutzt werden können.

Daher sollte die Position der Oktobots auf Bereiche beschränkt werden können, die sicherstellen, dass die Position einigermaßen sinnvoll ist. Innerhalb dieser Bereiche sollte der Oktobot jedoch genug Freiheit haben, um seine Position selbst optimieren zu können. Durch eine geschickte Wahl der Bereiche lässt sich außerdem Domänenwissen einbauen, also die menschliche Intuition, wo etwa eine Positionierung sinnvoll wäre. Gleichzeitig hat der Roboter jedoch die Freiheit, seine Position innerhalb des Bereiches zu optimieren, wodurch sich der Algorithmus auch gut auf unbekannte Spielsituationen anwenden lässt, ohne dass man als Programmierer hierfür extra Fälle definieren muss.

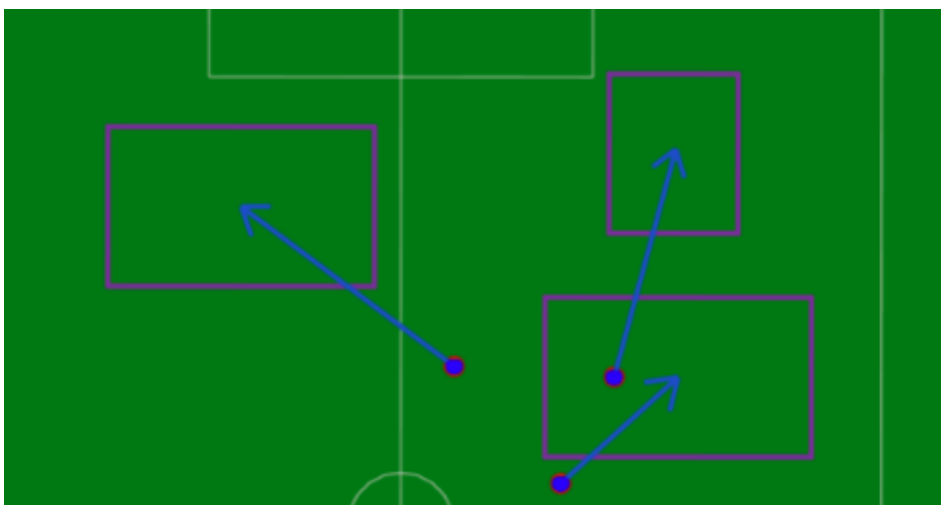
Die angesprochenen Bereiche werden im Allgemeinen als Aquarium bezeichnet. Befindet sich der Oktobot aktuell außerhalb seines zugewiesenen Aquariums, so soll er in dessen Mitte fahren. Sobald er sich im Aquarium befindet, beginnt die Abtastung und damit verbundene Positionsoptimierung. Außerdem lässt sich verhindern dass der Oktopus das Aquarium verlässt, indem für Tentakel außerhalb des Aquariums negative Scores gesetzt werden.

In der folgenden Abbildung wird gezeigt, wie sich die Aquarien definieren lassen. hier werden je nach Ballposition drei unterschiedliche Fälle definiert, die einem defensiven, ausgeglichenen oder

aggressiven Spielstil entsprechen. Jedes Aquarium hat dabei eine Priorität, die angibt, welche Aquarien besetzt werden sollten, falls es nicht genug Roboter gibt. Kleinere Zahlen werden dabei zuerst besetzt. Im defensiven Fall ist die Priorität, denn Ball schnell nach vorne zu bekommen, um die aktuelle Situation zu entschärfen und die Gefahr eines Gegentores zu verringern. Im Ausgeglichenen und Aggressiven Fall sind Aquarien nahe des gegnerischen Strafraums, um die eigenen Torchancen zu maximieren. Außerdem werden auch Aquarien hinter dem Ball definiert, um Rückpässe zuzulassen. Diese erlauben einerseits die Durchführung von Reflexschüssen, bei denen der Ball ohne Annahme schnell aufs Tor geschossen wird, was schwer zu halten ist. Andererseits lassen sich durch Rückpässe auch die gegnerischen Mauern effektiv umspielen, da der Ball schnell an die andere Seite des Strafraums weitergeleitet werden kann.



Nachdem die Lage der Aquarien bestimmt wurde, müssen die Oktobots diesen zugeordnet werden. Dabei ist das Ziel, dass alle Aquarien so schnell wie möglich besetzt werden. Um dies zu gewährleisten wird der Munkres Algorithmus verwendet. Dabei werden als Kosten die quadratischen euklidischen Abstände verwendet, damit lange Wege besonders bestraft werden. Der Fokus liegt somit darauf, dass Bewegungen parallelisiert werden. Das bedeutet auch, dass bereits besetzte Aquarien wieder verlassen werden, damit andere Oktobots geringere Wege fahren müssen.



Erfahrungen und Kritik

In Bordeaux 2023 hat sich der Oktopus Algorithmus in Kombination mit den Passbäume_Bordeaux_2023 als äußerst effektiv erwiesen. Viele der Spiele konnten vor Ablauf der vollen 10 Spielminuten durch ein 10:0 frühzeitig beendet werden. Insbesondere durch den Einsatz der Passbäume war ein schnelles und effektives Ausspielen der gegnerischen Verteidigung möglich.

Der Oktopus Algorithmus ist dabei besser als andere Ansätze wie die kraftbasierte Positionierung oder die Restricted Radial Positioning, wie sie in [Offense_Bod23](#) beschrieben werden, da die Roboter neben der Anspielbarkeit auch die Torwahrscheinlichkeit maximieren, statt sich nur auf die Anspielbarkeit zu fokussieren. Der Kompromiss aus Beschränkung durch die Aquarien sowie Freiheit ist dabei gut geeignet, um einerseits Domänenwissen miteinfließen zu lassen und gleichzeitig eine ausreichende Abstraktion für unbekannte Szenarien zu erreichen.

Weiterentwicklung

Für Eindhoven 2024 soll der Algorithmus weiterverwendet werden, dabei allerdings auf mehr Roboter ausgelegt sein, um auch in Div A verwendet werden zu können.

Revision #2

Created 7 August 2025 15:00:22 by Malte Sparenborg

Updated 7 August 2025 15:05:26 by Malte Sparenborg