

Coalition skill games with incomplete information and over-abundance of tasks

Mihail Bogojeski, Tobias Pockrandt, Mitja Phillip Richter, Björn Fischer

Abstract

In diesem Paper stellen wir einen Lösungsansatz für das Matching von Agenten mit gewissen Fertigkeiten auf Aufgaben welche besagte Fertigkeiten benötigen. Das Ziel ist es bei fehlen von vollständiger Information eine Maximierung der sozialen Wohlfahrt zu erreichen. Hierzu werden verschiedene Methoden und Konzepte aus Bereichen wie der Spieltheorie angewandt. Der vorgestellte Lösungsansatz wird zirka 80% der theoretisch möglichen optimalen Lösung erreichen und wir liefern eine Reihe von Maßnahmen die eine Verbesserung dieses Wertes ermöglichen sollten, beziehungsweise die Rechenzeit bei minimalem Verlust verbessern.

Keywords

coalition games — skills — incomplete information — matching problem

Contents

	Introduction	1
1	Versuchsumgebung und Beschreibung	1
2	Algorithmusbeschreibung	2
3	Methoden und Konzepte	3
4	Evaluation	4
5	Future Work	5
6	Conclusion	7
	Acknowledgments	7

Introduction

Dieses Paper stellt einen Abschlussbericht für den Kurs *Agententechnologien in der Forschung* dar und repräsentiert die im Semester erarbeiteten Resultate. Das Zusammenführen von Agenten und Aufgaben im Rahmen eines Allokationsspiels ist ein äußerst komplexes Problem. Aus diesem Grund werden einige Einschränkungen vorgenommen. Es existiert ein Pool aus Fertigkeiten und zugehörigen Intensitäten. Aus diesem Pool werden die Aufgaben und die Fertigkeiten der Agenten entnommen. Zwischen der Menge von Agenten und der, der Aufgaben soll eine möglichst effiziente Zusammenführung gefunden werden. Effizienz definiert hierbei den Grad in dem die soziale Wohlfahrt maximiert wurde. Um einen erhöhten Bezug zu Realität zu erreichen besitzen die Agenten nur ein beschränktes Wissen über ihre Umgebung. Dies führt zusätzlich dazu das die beste Lösung nicht durch einfaches propagieren aller Permutationen ermittelbar ist.

Für die beschriebene Spezifikation des Problems wird in diesem Paper eine Lösung vorgestellt. Hierfür wird zunächst auf die genauen Rahmenbedingungen des Versuchs eingegangen (KAPITEL 1). Im Anschluss erfolgt eine genauere Be-

trachtung des angewandten Algorithmus (KAPITEL 2) sowie der genutzten Methoden und Konzepte (KAPITEL 3). Um die Qualität der präsentierten Lösung zu bewerten wurde eine Evaluation durchgeführt (KAPITEL 4). Den Abschluss bilden ein Ausblick über mögliche Verbesserungen (KAPITEL 5), sowie eine Zusammenfassung (KAPITEL 6).

1. Versuchsumgebung und Beschreibung

Um eine spannende Umgebung zu erschaffen, bezieht sich der gewählte Algorithmus nicht einfach auf Agenten und Aufgaben beziehungsweise Auftragsnehmer und Firma. Ein attraktiveres und intuitives Szenario meinen wir mit Abenteurern (Agenten) und Abenteuern (Aufgaben) gefunden zu haben. Die Abenteurer werden in einer Kaschemme angeheuert, um gewisse Abenteuer gegen die Zahlung einer bestimmten Menge Gold zu erfüllen. Im folgenden wird daher auch von Abenteurern und Abenteuern gesprochen. Die Abenteurer besitzen einen von drei möglichen Skills (Fertigkeiten). Der Pool aus Skills besteht aus *Kämpfen*, *Verhandeln* und *Schleichen*. Jeder Fertigkeit eines Agenten ist ein Power-Wert (Intensität) zugeordnet, die beschreibt wie gut er diese beherrscht. Die Abenteurer benötigen von jedem Skill einen gewissen Wert um erfüllt zu werden. Eine Abenteurer kann nicht abgeschlossen werden, solange nicht alle Anforderungen komplett erfüllt worden sind.

Aufgrund des Überangebots von Aufgaben existieren mehr Abenteurer als Abenteuer, beziehungsweise benötigen die Abenteurer mehr Power um geschlossen zu werden als die Agenten in ihrer Summe verfügen. Sowohl die Abenteurer als auch die Abenteuer werden in einem gewissen Rahmen zufällig generiert. Für erstere wird ungefähr die Hälfte der Gesamtpower aller Abenteuer auf die Abenteurer verteilt. Die Abenteurer unterscheiden sich in ihrer Größe (klein, mittel, groß), wobei viele kleine und wenig große Abenteurer

existieren. —TODO—Der Erlös einer Abenteuer im Vergleich zu ihrer Größe ist allerdings nicht linear, sodass das ein großes Abenteuer proportional mehr Gold ergibt als die Summe kleinerer mit gleicher Power. Dies entspricht auch der Realitätsnähe des Versuchs, da Großprojekte in der Regel einen größeren Gewinn ausstoßen als kleinere.

Wie im Titel der Arbeit erwähnt, stehen in diesem Szenario nur unvollständigen Informationen zur Verfügung. Speziell bedeutet dies, dass zwar jeder Abenteuerer alle Abenteuer kennt, inklusive deren Bedarf und Erlös, allerdings keine konkreten Informationen darüber besitzt, wie viele andere Agenten im Spiel sind und insbesondere nicht, welche Skills und wie viel Power sie besitzen.

In der Planungsphase des Versuchsaufbaus wurde diskutiert, mit wie vielen Skill-Arten ein Abenteuerer ausgestattet werden soll: Ein Skill pro Abenteuerer oder zwei bis drei. Für den Besitz von zwei bis drei Skills sprachen eine interessantere Verhandlungsphase zwischen den Abenteuerern. Dagegen standen der stark erhöhte Aufwand, sowie die Realitätsnähe, da eine Firma zumeist auf eine Aufgabe spezialisiert ist. Weiterhin bestand die unbewiesene Annahme, dass sich ein Abenteuerer mit mehreren Skills auch als mehrere Abenteuerer mit jeweils einem Skill darstellen lässt. So führte auch nicht zuletzt die Tatsache, dass hier lediglich ein relativ ausgereifter Lösungsansatz vorgestellt werden soll, zu der Entscheidung Abenteuerer mit jeweils einem Skill zu verwenden.

Weiterhin besitzen die Agenten für jedes Abenteuer gewisse Grundkosten die anfallen und mit dem Gewinn abgedeckt werden müssen. In unserem Beispiel ist das die Beschaffung von Tränken und die Instandhaltung von Waffen und Ausrüstung. In der Realität könnte sich dies mit Fahrtkosten oder Wartung von Maschinen vergleichen lassen. Ein weiteres Ziel welches damit verfolgt wird, ist eine Erhöhung der Individualität der einzelnen Agenten. Die Abenteuerer besitzen alle das gleiche Grundverhalten, sodass sich ihre Handlungen nur durch ihre eigene Power und die Grundkosten unterscheidet. Ergo gibt es hier keine risikoscheuen oder risikoaversen Agenten. Für alle Agenten existiert ein Zeitlimit (Deadline) welches bekannt ist, nach welchem keine Bewerbungen auf Abenteuer mehr möglich sind. Somit werden die Abenteuerer bei fortgeschrittener Zeit auch in weniger profitablere Alternativen wechseln um noch Gewinne zu realisieren.

Initiale Variationen sind in diesem Szenario: Die Anzahl an Abenteuerern und die der Abenteuer. Die Einhaltung der Rahmenbedingungen erfolgen anschließend automatisch. Eine genaue Beschreibung des Ablaufes des Algorithmus erfolgt im nächsten Kapitel.

2. Algorithmusbeschreibung

Zunächst erfolgt eine grobe Zusammenfassung des Algorithmus, wobei die einzelnen Bestandteile in den folgenden Unterkapiteln ausführlicher beleuchtet werden. Zu Beginn berechnen die Agenten für jedes Abenteuer ihren Nutzen. Anschließend dürfen sie sich auf bis zu vier Abenteuer be-

werben. Anhand dieser Bewerbungen werden Koalitionen gebildet und die Beste anhand des Banzhaf Power Index ausgewählt. Potentieller Überschuss in den Koalitionen wird entfernt und möglichst sinnvoll an die Agenten zurückgezahlt. Erst dadurch wird ein Erreichen des maximal möglichen Erlöses möglich. Die Information über den zu erwartenden Gewinn wird an die Agenten weitergegeben, welche sich auf dieser Grundlage auf Abenteuer beziehungsweise Koalitionen festlegen. Letztlich werden die geschlossenen Abenteuer zusammen mit der eingesetzten Power aus dem Spiel entfernt und die entsprechenden Belohnungen werden an die Agenten ausgezahlt. Sofern theoretisch noch ein Abenteuer geschlossen werden kann und die Zeit noch nicht abgelaufen ist beginnt der Algorithmus in einer neuen Iteration.

Bewerbungsphase: Jeder Agent berechnet seinen Nutzen für jedes Abenteuer und bewirbt sich für maximal vier Abenteuer, die den höchsten Nutzen für ihn haben. Die Agenten bewerben sich mit ihren gesamten verfügbaren Power. Sollte die von dem Abenteuer benötigte Power überschritten werden, bewirbt sich der Agent mit der exakt benötigten Power. Die Bewerbungen aller Agenten werden gesammelt und als Grundlage für die Koalitionsermittlung verwendet.

Koalitionsermittlung: Für jedes Abenteuer, werden zunächst alle möglichen Koalitionen aus den Agenten, die sich für dieses Abenteuer beworben haben, gebildet. Danach werden alle Koalitionen die keine gewinnende Koalitionen sind (die Koalitionen, die die Anforderungen des Abenteuers nicht erfüllen), eliminiert. Falls es keine Gewinnerkoalition gibt, wird die große Koalition als die Koalition, die der Erfüllung der Anforderungen am Nächsten ist, gespeichert. Schließlich wird aus der Gewinnerkoalitionen die Banzhaf Power für jeden Agent in diesem Abenteuer berechnet. In dieser Phase werden auch die Veto-Spieler in jedem Abenteuer ermittelt und ihr ihre jeweilige Banzhaf Power mit einem bestimmten Faktor multipliziert (in unser Implementierung verdoppelt), um die Macht der Veto-Spieler im Vergleich zu den anderen Spieler zu verdeutlichen.

Beste Koalition: Die beste Koalition für eines Abenteuer wird mit einer leicht geänderter Version des verteilten Algorithmus ermittelt. Zuerst werden die Koalitionen eliminiert, die einen nicht minimalen Überschuss an investierter Power haben. Unter Überschuss wird hier die Differenz zwischen der investierten Power der Koalition und der benötigten Power des Abenteuers für jeden Skill verstanden. Eine Koalition die einen hohen Überschuss hat ist nicht effizient, da die überflüssige Power potentiell bei anderen Abenteuer eingesetzt werden könnte. So werden auch Dummy-Spieler aus der potentiell besten Koalition entfernt, da es immer eine Koalition ohne Dummy-Spielern und mit kleinerem Überschuss gibt. Der verteilte Algorithmus wird dann auf die Koalitionen mit minimalem Überschuss angewandt, wo die beste Koalition nach elitärer Gewinnmaximierung ausgewählt wird. Genauer gesagt, die beste Koalition ist die Koalition, für die irgendein Agent die größte Gewinnschätzung hat.

Überschusstilgung: In dieser Phase wird die überflüssige Power aus der besten Koalition entfernt und an den jeweiligen Agenten zurückgegeben. Da Power ein ganzzahliger Wert ist, kann die überschüssige Power meistens nicht gleichmäßig auf alle beteiligten Agenten aufgeteilt werden. In unsere Implementierung wird die Aufteilung des Überschusses durch eine iterierte randomisierte Prozedur realisiert. Für jeden Skill bei dem es einen Überschuss gibt, wird in jeder Iteration eine Power-Einheit einem Agenten zurückgegeben. Jeder Agent, der Power von einem Skilltyp investiert hat, kann mit einer gewissen Wahrscheinlichkeit eine Power-Einheit von der Überschuss in diese Skilltyp zurückbekommen. Diese Wahrscheinlichkeit ist gleich der Anteil der investierten Power des Agenten aus der gesamten investierten Power aller Agenten. Das heißt, dass Agenten die mehr Power aus einen Skilltyp investiert haben, mit einer höheren Wahrscheinlichkeit eine Power-Einheit aus dem Überschuss zurückbekommen. Diese Prozedur wird so oft wiederholt, bis den ganzen Überschuss aus der besten Koalition entfernt wird.)

Gewinnverteilung: Der Gewinn wird abhängig von der Banzhaf Power der Agenten aufgeteilt. Jeder Agent in der besten Koalition bekommt den Anteil am Gesamtgewinn, der dem Anteil seiner Banzhaf Power an der aufsummierten Banzhaf Power aller Agenten entspricht. Aufgrund der Verwendung des Banzhaf Power Index ist diese Verteilung fair bezüglich der Macht der Agenten und zusätzlich auch pareto-effizient, da der gesamte Gewinn verteilt wird.

Festlegung auf Abenteurer/Koalition: Da sich die Agenten auf maximal vier Abenteurer mit ihrer gesamten Power bewerben können, muss sich danach jeder Agent auf nur ein Abenteurer festlegen. Die Entscheidung erfolgt in zwei Phasen: In der ersten Phase, wählt jeder Agent das Abenteurer aus, in welchem seine Gewinnschätzung am höchsten ist. Für die zweite Phase werden diese Entscheidungen den Agenten bekannt gegeben, die sich ebenfalls für die jeweiligen Abenteurer beworben haben. Jetzt haben die Agenten die Möglichkeit ihre Entscheidung zu ändern. Diesmal entscheiden die Agenten nicht anhand ihrer Gewinnschätzung, sondern anhand ihrer Nutzenfunktion, die die Entscheidungen der anderen Agenten aus der ersten Phase miteinbezieht. Die Agenten wählen das Abenteurer mit dem größten Nutzen und diese Entscheidung ist für die aktuelle Runde endgültig.

Schliessung erfüllter Abenteurer: Nachdem sich jeder Agent für eine einzige Abenteurer entschieden hat, werden die Abenteurer, wo die Gewinnerkoalition vollständig ist (d.h. jeder Agent in der Koalition hat sich auf dieses Abenteurer festgelegt). Falls einen Abenteurer geschlossen wird, wird das Gewinn an der Agenten ausgezahlt, die investierte Power der Agenten aus ihren Power abgezogen und das geschlossene Abenteurer entfernt. Ab der nächste Runde kann sich kein Agent mehr für das geschlossene Abenteurer anmelden.

3. Methoden und Konzepte

Hier erfolgt eine Zusammenfassung der Methoden und Konzepte, welche unter anderem aus der Spieltheorie kommen um einen Überblick zu erhalten für potentielle Veränderungen aus KAPITEL 5 oder Diskussionen für und gegen die angewandten bzw. nicht angewandten Mittel.

Gewinnschätzung: Die Gewinnschätzung des Agenten für einen Abenteurer ist eine ganz wichtige Funktion, die in viele Teile der Implementierung benutzt wird. Für ein bestimmtes Abenteurer, wird die Gewinnschätzung wie folgt berechnet: Falls der Agent kein Mitglied der aktuellen Gewinnerkoalition des Abenteurers ist, schätzt er, dass er seinen fairen Anteil des Gewinns bekommt, d.h. er erwartet den gleichen Anteil des Gewinns, wie der Anteil seine investierte Power von der gesamten benötigten Power des Abenteurers. Falls der Agent Teil der großen Koalition ist, die aber nicht alle Anforderungen des Abenteurers erfüllen kann, schätzt der Agent das er der gleichen Anteil des Gewinns bekommt, wie der Anteil seine investierte Power von der gesamten Power der Koalition. Wenn der Agent ein Mitglied der aktuellen Gewinnerkoalition ist, kennt er seinen Banzhaf Power für dieses Abenteurer und schätzt sein Gewinn als der gleichen Anteil des gesamten Gewinns, wie der Anteil seiner Banzhaf Power aus der Summe der Banzhaf Power aller Agenten in dieser Koalition.

Nutzenfunktion: Die Nutzenfunktion ist das wichtigste Mittel, um das Verhalten der Agenten zu steuern. Die Basis des Nutzen eines Agenten von einem bestimmten Abenteurer wird aus der Gewinnschätzung minus die Kosten für das Abenteurer gebildet. Dann wird dieser Wert abhängig von den Eigenschaften des Abenteurers angepasst und daraus der endgültige Nutzen erhalten. Es werden vier Eigenschaften eines Abenteurers und die zugehörige beste Koalition unterschieden:

- E1 Die Anzahl der bisherigen Fehlversuche bei diesem Abenteurer. Eine frühere Bewerbung zählt als Fehlversuch, wenn der Agent mit dieser Bewerbung kein Mitglied der Gewinnerkoalition geworden ist.
- E2 Diese Eigenschaft spielt nur dann eine Rolle, wenn die große Koalition nicht alle Anforderungen des Abenteurers erfüllt. Es wird der Anteil der fehlenden Power der große Koalition zur Schließung des Abenteurers aus der gesamten benötigten Power für das Abenteurer berechnet.
- E3 Diese Eigenschaft spielt auch nur dann eine Rolle, wenn die große Koalition nicht alle Anforderungen des Abenteurers erfüllt. Es wird die Anzahl an Skills gezählt, bei denen die große Koalitionen einen Mangel an Power hat. Dieser Wert wird dann durch den Anzahl an insgesamt benötigten Skills für das Abenteurer dividiert.
- E4 Diese Eigenschaft spielt in der zweite Phase des Festlegungsprotokolls eine Rolle. Sie wird genau wie E2

berechnet, aber statt der großen Koalition, wird hier die Anteil der fehlenden Power der Koalition aus der bereits festgelegten Agenten berechnet. Diese Eigenschaft drückt also aus, wie viel Power benötigen die bereits festgelegten Agenten um das Abenteuer abzuschließen.

Wie bei der Gewinnschätzung, wird in der Nutzenfunktion zwischen drei Fälle unterschieden, wobei in jedem Fall unterschiedliche Eigenschaften des Abenteurers einen Einfluss auf der ursprünglichen Gewinnschätzung haben. Im ersten Fall ist der Agent kein Mitglied der Gewinnerkoalition des Abenteurers und die Gewinnschätzung wird von Eigenschaft E1 und der Anzahl an vergangenen Runden negativ beeinflusst, d.h. der Nutzen für so einen Abenteurer wird später im Spiel und mit steigendem Anzahl an Fehlversuche immer kleiner.

Der zweite Fall tritt auf, wenn der Agent Teil der großen Koalition ist, die aber nicht alle Anforderungen des Abenteurers erfüllen kann. Die Gewinnschätzung wird in diesem Fall von E1 und der Anzahl an vergangenen Runden auch negativ beeinflusst, kann sich aber in Abhängigkeit von E2 und E3 vergrößern oder verkleinern. Die Nutzen ist daher in diesem Fall größer, falls die große Koalition nur wenig Power benötigt um das Abenteuer abzuschließen.

Im dritten Fall, wenn der Agent einen Mitglied der Gewinnerkoalition ist, wird die Gewinnschätzung von der Anzahl an vergangenen Runden stark positiv beeinflusst, d.h. der Nutzen eines Abenteurers im diesen Fall steigt in den letzten Runden sehr stark. Außerdem wird in den letzten Runden der positive Einfluss von Eigenschaft E4 auch viel stärker. Also die Nutzen in dritten Fall wird größer, wenn die bereits festgelegten Agenten wenig Power benötigen, um das Abenteuer zu schliessen, und diesen Nutzen steigt in den letzten Runden besonders viel.

So einen Nutzenfunktion stellt sicher, dass die Agenten am Anfang mit ihren Bewerbungen flexibel sind und versuchen, einen maximalen Gewinn zu erzielen. Andererseits, in den späteren Runden, versucht jeder Agent vor dem Deadline irgendein Gewinn zu erzielen, auch wenn dieser für ihn nicht optimal ist. In diese weise wird ein realistisches Verhalten der Agenten modelliert.

Banzhaf-Power: Um die Banzhaf-Power eines Agenten zu bestimmen wird gezählt in wie vielen möglichen Koalitionen dieser ein kritischer Spieler ist. Wie vorher erwähnt, wird die Banzhaf-Power der Veto-Spieler zusätzlich mit einem Faktor multipliziert, um die Tatsache auszudrücken, dass Veto-Spieler viel mächtiger als den anderen Spielern sind.

Egalitäre und elitäre Funktionen: Für die Ermittlung der besten Koalition haben wir auch eine egalitäre Version des Algorithmus getestet, wo wir die Koalition, mit dem größten minimalen Nutzen für einen Agent, als die beste Koalition ausgewählt haben. Diese Version des Algorithmus hat schlechtere, aber nicht signifikant unterschiedliche Ergebnisse als die elitäre Version erzielt.

Deadline: Die Deadline war ein ganz wichtiges Konzept in unser Implementierung, das das Verhalten der Agenten stark beeinflusst hat. Dank der Deadline ist der Spielzustand in jeder Runde ein bisschen anders, was ein zyklisches Verhalten der Agenten verhindert. Die Deadline hat auch einen großen Einfluss auf der Konvergenz unser Algorithmus, weil das Verhalten der Agenten in den letzten Runden vor der Deadline viel stabiler wird. So wird sicher gestellt, dass die Agenten in den letzten Runden so viele Abenteuer wie möglich abschließen, obwohl diese Abenteuer für jeden einzelnen Agent nicht einen optimalen Gewinn liefern.

4. Evaluation

Testaufbau und Initialisierung: Die Evaluation unser Implementieren war mit verschiedenen Kombinationen von 10-15 Agenten, 10-20 Abenteuer und 10-50 Runden durchgeführt. Abenteuer mit Anforderungen von verschiedenen Größen werden von einer Beta-Verteilung mit Parametern $\alpha = 2.3$ und $\beta = 4.6$ generiert. Für jedes Abenteuer wird das so generierte Power an zwei oder drei Skills beliebig verteilt. Schließlich, wird der Gewinn eines Abenteurers mit eine superlineare Funktion aus der benötigten Power zusammen mit einem gewissen randomisierten Rauschen ausgerechnet. So ist eine Power-Einheit, die in einem Abenteuer mit höheren Anforderungen investiert ist, im Schnitt mehr wert als eine Power-Einheit in einem Abenteuer mit niedrigen Anforderungen.

Sobald alle Abenteuer generiert sind, wird der angegebenen Anzahl der Agenten generiert, sodass für jedes Skill, die gesamte Power aller Agenten 40-50% der gesamten benötigten Power aller Abenteuer beträgt. So wird sichergestellt, dass es eine Überangebot an Abenteuer gibt. Die Anzahl der Agenten, die einen bestimmten Skill besitzen, wird in Abhängigkeit von der Höhe der benötigten Power für dieses Skill in allen Abenteuern bestimmt. Das heißt, wenn für einen bestimmten Skill in allen Abenteuern mehr Power als den anderen Skills benötigt wird, werden auch mehr Agenten diesen Skill besitzen. So werden alle Agenten im Schnitt den gleichen Power haben, unabhängig von welchen Skill sie besitzen. Als letztes werden die Kosten der Agenten für jedes Abenteuer in einem bestimmten Intervall beliebig generiert.

Greedy Bound und Upper Bound: Die optimale Lösung, und damit der maximal erreichbaren Gewinn für einen Spiel auszurechnen ist sehr aufwändig (äquivalent zu einer Lösung der multidimensionalen Rucksackproblem, was NP-schwer ist). Deswegen haben wir haben wir zwei heuristische Werte für jedes Spiel ausgerechnet: die Greedy und Upper Bound, die wir mit dem von unser Implementierung erzielten Gewinn vergleichen, um unser Algorithmus zu evaluieren.

Die Greedy Bound ist der Gewinn, der mit einem Greedy Algorithmus mit vollständiger Information im Spiel erzielen kann. Der Greedy Algorithmus schliesst in jede Iteration das Abenteuer mit dem höchsten Gewinn das geschlossen werden kann, bis es kein Abenteuer gibt das geschlossen werden kann. Der Greedy Algorithmus liefert natürlich nicht immer

die optimale Lösung, ist aber in der Regel relativ nah dem Optimum.

Die Upper Bound ist dann eine echte Obergrenze, d.h. die Gewinn von der Upper Bound ist immer gleich oder höher als dem maximal erreichbaren Gewinn. Dieser Gewinn wird wie folgt berechnet: In jeder Iteration wird versucht, das Abenteuer mit dem höchsten Gewinn zu schließen. Falls das möglich ist, wird das Abenteuer geschlossen und der Algorithmus geht weiter. Wenn die Agenten nicht genug Power haben dieses Abenteuer zu schließen, wird so viel Power wie möglich in das Abenteuer investiert, und der gleichen Anteil des Gewinns an den Agenten ausgezahlt, wie der Anteil der investierten Power von der benötigten Power des Abenteuers. Dieser Schritt wird dann so lange wiederholt, bis die Agenten kein Power mehr zu Investieren haben. Die so ausgerechnete Upper Bound, ist deswegen höher als dem maximal erreichbaren Gewinn, weil der Gewinn eines Abenteuers superlinear von der benötigten Power abhängig ist.

Ergebnisse: Wir haben hauptsächlich vier verschiedenen Konfigurationen getestet:

- K1 10 Agenten, 10 Abenteuer, 50 Runden - 1000 Spiele getestet
- K2 10 Agenten, 15 Abenteuer, 50 Runden - 1000 Spiele getestet
- K3 15 Agenten, 10 Abenteuer, 50 Runden - 50 Spiele getestet
- K4 15 Agenten, 20 Abenteuer, 50 Runden - 50 Spiele getestet

Wir könnten Konfigurationen mit mehr als 15 Agenten nicht effizient testen, da der Rechenaufwand für Spiele mit vielen Agenten wegen der Koalitionsermittlung sehr hoch ist. Die Ergebnisse von der Evaluation aller vier Konfigurationen können auf den Abbildungen 1-4 gesehen werden. Die Abbildungen zeigen der mittleren Gewinn aller Agenten für jede Runde über alle getesteten Spiele, sowie die mittlere Upper und Greedy Bound. Außerdem, wird in der Beschreibung eingeführt, wie viel Prozent aus der Upper und Greedy Bound erzielt im Mittel unsere Implementierung. fig:

5. Future Work

Wie bereits aus der Evaluation des vorherigen Kapitels, bekannt erreicht der von uns entwickelte Algorithmus eine recht akzeptable Leistung, wobei durchaus noch Verbesserungen bzw. Veränderungen möglich und notwendig sind. Diese weiteren Arbeiten können und sollen unter anderem die Effizienz (nähe zum Optimum) als auch die Geschwindigkeit verbessern. Es sind unter anderem auf Maßnahmen aufgelistete, welche den Informationsgehalt des Versuchs erhöhen, indem z.B. mehr (spezifischere) Daten betrachtet und damit auch berücksichtigt und ausgelesen werden können.

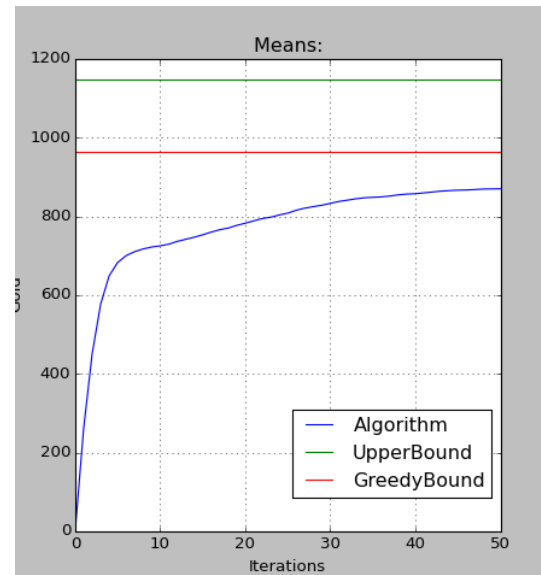


Figure 1. Konfiguration K1: 76% Upper Bound, 90% Greedy Bound

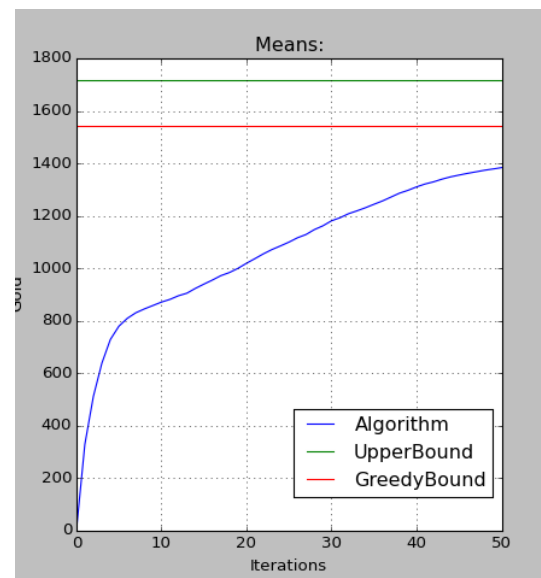


Figure 2. Konfiguration K2: 81% Upper Bound, 90% Greedy Bound

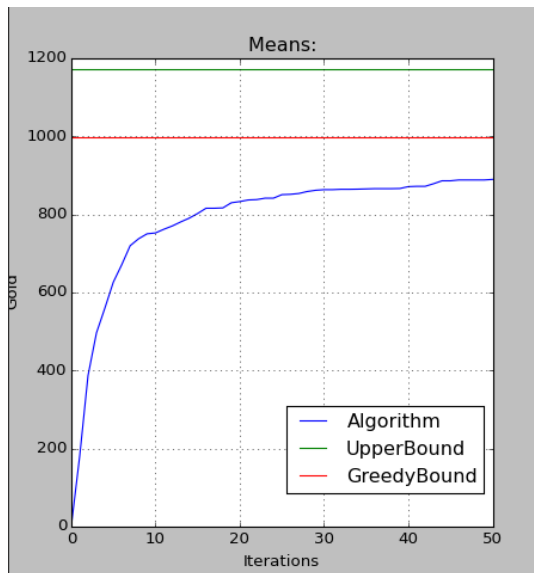


Figure 3. Konfiguration K3: 75% Upper Bound, 89% Greedy Bound

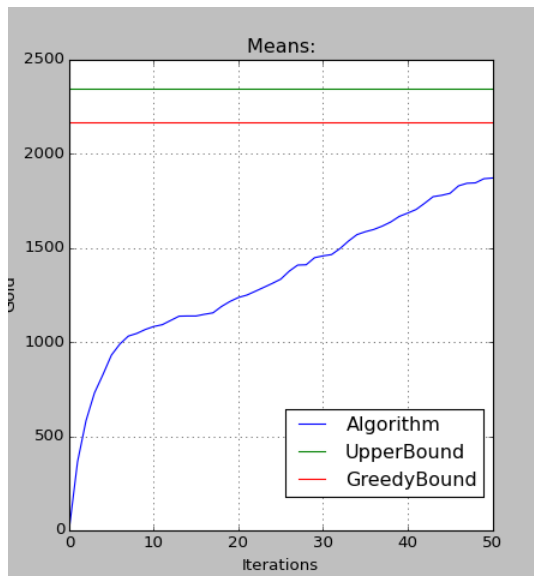


Figure 4. Konfiguration K4: 80% Upper Bound, 87% Greedy Bound

Verteilung des Algorithmus: Die hier gewählte Implementation ist kein echter verteilter Algorithmus. Die Hauptarbeit wird von einer zentralen Instanz durchgeführt, welche die Berechnungen und Entscheidungen für die Agenten (in deren Sinne) durchführt. Wenn jeder Agent als Einzelne Entität in einem verteilten Algorithmus realisiert ist, lassen sich sehr einfach individuelle Verhaltensmuster für jeweilige Abenteurer festlegen. Dann können unter anderem verschiedene Risikoaspekte mit in den Aufbau eingewoben werden. Was bedeutet risikoscheue und averse Agenten können aufeinander Treffen.

Ein weiterer Aspekt könnte sein, dass ein Agent A unter keinen Umständen mit einem Agenten B eine Koalition bilden möchte, aufgrund schlechter Erfahrungen oder ähnliches. Hierbei könnte untersucht werden in wie weit die Effizienz des gesamten Konstruktes, durch den Ausschluss der Zusammenarbeit zweier Agenten, in Mitleidenschaft gezogen wird. Der Kreativität im Design vom Verhalten einzelner Agenten ist hierbei kein Limit gesetzt.

Optimierung der Koalitionsbildung: Da aktuell jeder Koalition berechnet werden muss, aufgrund des Banzhafpowerindex befindet sich der Aufwand in einem Exponentiellen Bereich, was dazu führt, dass der Algorithmus schon bei einer vergleichsweise geringen Anzahl an Abenteuern extrem viel Zeit benötigt. Es muss demnach eine Möglichkeit gefunden werden ein im besten Fall identisches Ergebnis in geringerer Zeit zu finden. Der intuitive Ansatz ist hierbei nicht alle Koalitionen zu betrachten, sondern eine simple Vorauswahl zu treffen welche die Anzahl einschränkt. Wie diese auszusehen hat und wie groß der Effizienzverlust ist muss ermittelt und evaluiert werden.

Maschinelles Lernen: In der von uns entwickelten Lösung ist Maschinelles Lernen nur sehr marginal vertreten. Agenten merken sich unter anderem wie oft sie bereits an einer Koalition gescheitert sind. Wenn dieser Anteil erheblich ausgebaut werden würde und sich jede Aktion und ihr Resultat quasi gespeichert wird, lassen sich unter Umständen ganz neue Lösungsverhalten finden, welche das Optimum näher erreichen. Hierbei sind allerdings nicht Lernphase über die Dauer eines Spiels gemeint, sondern lernen anhand von 1000 und oder mehr Spielen das anzustrebende Ergebnis. Sodass sich mittels Lernen unter Umständen neue oder unerwartete Lösungen mit ähnlichem bzw. besserem Ergebnis finden lassen. Dieser Lernende Agent kann gegen "gewöhnliche" Agenten oder sogar Menschen Spielen, diese schnell Anhand ihres Verhaltens identifizieren bzw. kategorisieren und eine geeignete Strategien anzuwenden. Dies sei nur als Anreiz für die Einbettung maschinellen Lernens zu verstehen und erhebt nicht den Anspruch vollständig zu sein.

Untersuchung aller Parameter: Das großflächige Untersuchen sämtlicher Parameter könnte aufschlussreiche Ergebnisse liefern. In dieser Evaluation wurde nur an der Menge von Agenten und Abenteuern manipuliert. Weitere Stellschrauben an denen zu drehen ist, sind:

- *Belohnung*: Die Verteilung von Auszahlungen könnte variiert werden. Den Anstieg noch mehr Quadratische, oder linear oder mittels einer Beliebigen Fkt darstellen. Auch ein stagnierender Anstieg der Auszahlung für größere Abenteuer ist denkbar.
- *Skillmengen*: Statt 3 Skills, könnte eine Beliebige Anzahl an Skills vorhanden sein (z.B. 5). Eine weitere denkbare Alternative ist, das nicht jede Baustelle alle Skills braucht, oder die Agenten mehr als nur einen Skill besitzen, wodurch die Verhandlung eine weitere Dimension erreicht.
- *Powergrößen*: Die Power die jeder Agent hat bzw. alle Agenten haben oder die Abenteuer haben, kann weiterhin variiert werden.
- *Angebot*: Zum einen kann das Überangebot der Aufgaben noch verschärft oder zu ein Unterangebot abgewandelt werden. Dann würde sich auch die Frage nach dem Ergebnis abwandeln, da bei dieser Veränderung nicht von Interesse ist, wie viel Gold alle Agenten erwirtschaften, sondern wer bekommt grundsätzlich Gewinn und wie viel.
Ein weiteres Szenario ist eine komplette Gleichverteilung. Es ist demnach möglich alle Abenteuer zu lösen ohne das ein Skillpunkt eines Abenteurers übrig bleibt. Hier stellt sich die Frage, welcher Agent wie viel vom Gewinn erwirtschaften kann.
- *Agenten*: Die verschiedenen Verhalten die ein Agent aufweisen kann wurden bereits erläutert.

6. Conclusion

Acknowledgments

So long and thanks for all the fish [?].