

# **Institut für Produktion und Industrielles Informationsmanagement**

Universität Duisburg-Essen / Campus Essen  
Fakultät für Wirtschaftswissenschaften  
Universitätsstraße 9, 45141 Essen  
Tel.: ++ 49 (0) 201 / 183 – 4007  
Fax: ++ 49 (0) 201 / 183 – 4017

Arbeitsbericht Nr. 39

## **Implementierung des $\tau$ -Werts**

Daniel Bergenrodt

Susanne Jene

Univ.-Prof. Dr. Stephan Zelewski



daniel.bergenrodt@pim.uni-due.de; susanne.jene@pim.uni-due.de;

stephan.zelewski@pim.uni-due.de

Internet: <http://www.pim.wiwi.uni-due.de>

ISSN 1614-0842

Essen 2013  
Alle Rechte vorbehalten.

## **Zusammenfassung**

Der  $\tau$ -Wert ist ein Lösungskonzept der kooperativen Spieltheorie. Es kann eingesetzt werden, um Kosten oder Gewinne unter mehreren Akteuren fair aufzuteilen. Dies gilt insbesondere im Hinblick auf Effizienzgewinne, die in einer Kooperation gemeinsam erwirtschaftet wurden. Bisher wurde dieses Lösungskonzept in der betrieblichen Praxis jedoch nur selten angewendet. Um die Chancen des Einsatzes dieses Lösungskonzepts in der betrieblichen Praxis zu erhöhen, wird in diesem Arbeitsbericht ein Programm zur Berechnung des  $\tau$ -Werts erläutert, das die praktische Anwendung des  $\tau$ -Werts erheblich vereinfacht. Dieses Programm wird vorgestellt und anhand einiger Testfälle erprobt.

## Inhaltsüberblick

Inhaltsüberblick.....	II
Abkürzungs- und Akronymverzeichnis .....	III
Symbolverzeichnis .....	IV
1    Problemstellung .....	6
2    Der $\tau$ -Wert .....	9
2.1  Erläuterung des $\tau$ -Werts.....	9
2.2  Pseudocode zur Berechnung des $\tau$ -Werts.....	15
3    Implementierung des $\tau$ -Werts.....	25
3.1  Zielsetzung.....	25
3.2  Programmcode .....	25
3.3  Anwendung des Programms zur Berechnung des $\tau$ -Werts.....	33
4    Test des Programms zur Berechnung des $\tau$ -Werts .....	36
5    Fazit .....	42
Literatur.....	43

## Abkürzungs- und Akronymverzeichnis

bzw.	beziehungsweise
Dr.	Doktor
E-Mail	Electronic Mail
et al.	et alii
f.	folgende
ff.	fortfolgende
Hrsg.	Herausgeber
ISSN	International Standard Serial Number
Jg.	Jahrgang
No.	Number
Nr.	Nummer
S.	Seite
Univ.-Prof.	Universitätsprofessor
vgl.	vergleiche
Vol.	Volume
z. B.	zum Beispiel

## Symbolverzeichnis

$A$	Akteurmenge
$A_n$	Akteur aus der Akteurmenge
$AC_{n,q}$	q-te-Außenseiterkoalition, die vom Akteur $A_n$ angeführt wird
$\gamma$	Linearfaktor
$c$	charakteristische Funktion
$c_n$	Wert der charakteristischen Funktion $c$ für einen Akteur $A_n$
$C_m$	Koalition mit dem Index $m$
$C_0$	große Koalition bestehend aus allen Akteuren
$G$	Effizienzgewinn
$L$	Lösungspunkt
$IN_{n,q}$	Indexmenge aller Indizes von Akteuren, die zur Außenseiterkoalition $AC_{n,q}$ gehören
$MB_n$	marginaler Beitrag eines Akteurs $A_n$
$MC_n$	Marginalkoalition bezüglich des Akteurs $A_n$
$N$	Anzahl von Akteuren
$\mathbb{N}$	Menge aller natürlichen Zahlen
$OG$	obere Grenze für die Gewinnanteile aller Akteure
$\mathbb{R}$	Menge aller reellen Zahlen
$\mathbb{R}_{>0}$	Menge aller positiven reellen Zahlen
$\mathbb{R}_{\geq 0}$	Menge aller nicht-negativen reellen Zahlen
$\tau$	Symbol für den $\tau$ -Wert
$U$	Utopiapunkt
$UG$	untere Grenze für die Gewinnanteile aller Akteure
$v$	Verteilungsfunktion

---

$v_n$	Auszahlung, die dem Akteur $A_n$ als Ergebnis durch die Verteilungsfunktion $v$ zugeordnet wird
$v_{n.max}$	maximal zurechenbarer Gewinnanteil des Akteurs $A_n$
$v_{n.min}$	minimal zurechenbarer Gewinnanteil des Akteurs $A_n$

# 1 Problemstellung

Die Spieltheorie erfreut sich bereits seit mehreren Jahrzehnten großer Beliebtheit in der wissenschaftlichen Forschung. In diversen Gebieten, insbesondere der wirtschaftswissenschaftlichen Forschung, wird auf spieltheoretische Konzepte zurückgegriffen. Zwar existieren auch zahlreiche wissenschaftliche Beiträge, die spieltheoretische Konzepte auf praktische Probleme anwenden, jedoch ist die Anwendung der Spieltheorie in der betrieblichen Praxis insgesamt gering. Gründe dafür sind zum einen die komplexe und meist formale Darstellung und Beschreibung spieltheoretischer Konzepte in der Literatur. Zum anderen fehlt es für zahlreiche Konzepte an der in der betrieblichen Praxis erwünschten softwaretechnischen Unterstützung.<sup>1</sup> Eine Implementierung eines spieltheoretischen Konzepts in Form einer Software würde den letzten Kritikpunkt beheben und den ersten zumindest lindern.

Im Fokus dieses Arbeitsberichts steht die Implementierung eines Konzepts der kooperativen Spieltheorie, des  $\tau$ -Werts. Die kooperative Spieltheorie<sup>2</sup> befasst sich mit Entscheidungssituationen, bei denen die Beteiligten durch Absprache Vereinbarungen<sup>3</sup> treffen können, um stabile Kooperationen zu bilden.<sup>4</sup>

In der betrieblichen Praxis kooperieren Unternehmen, um Effizienzgewinne zu erzielen. Trotz wirtschaftlichen Erfolgs bestehen Kooperationen nicht immer langfristig. Ein Grund dafür kann eine Verteilung der Effizienzgewinne sein, die von zumindest einem Teil der

---

<sup>1</sup> Bisher ist nur für wenige spieltheoretische Konzepte eine softwaretechnische Unterstützung vorhanden.

<sup>2</sup> Die Spieltheorie wird unter anderem in die kooperative und die nicht-kooperative Spieltheorie unterschieden. Die kooperative Spieltheorie unterscheidet sich von der nicht-kooperativen Spieltheorie dadurch, dass bei der kooperativen Spieltheorie die Akteure, also in einer Kooperation die Kooperationspartner, die Möglichkeit haben zu kommunizieren und Vereinbarungen zu treffen. Bei der nicht-kooperativen Spieltheorie wird eine Kommunikation zwischen den Akteuren und somit die Möglichkeit, Vereinbarungen zu treffen, ausgeschlossen, vgl. z.B. HOLLER/ILLING (2006), S. 4, OSBORNE/RUBINSTEIN (1994), S. 2 f., und HARSANYI (1979), S. 7 f.

Ursprünglich wurde die Unterscheidung in kooperative und nicht-kooperative Spieltheorie von NASH eingeführt, vgl. NASH (1951), S. 286 ff. Zwar lässt sich zeigen, dass sich die Lösungskonzepte der kooperativen Spieltheorie auf die Lösungskonzepte der nicht-kooperativen Spieltheorie zurückführen lassen; vgl. z.B. BINMORE (1998), S. 42 ff. Von dieser Rückführungsmöglichkeit wird in dem hier vorliegenden Arbeitsbericht jedoch abgesehen, weil sie keinen Beitrag zur Lösung des konkreten Realproblems einer fairen Verteilung von Effizienzgewinnen erkennen lässt.

<sup>3</sup> Diese Vereinbarungen werden in der wissenschaftlichen Fachliteratur häufig als verbindlich oder glaubwürdig bezeichnet, vgl. z.B. MCCAIN (2010), S. 403 und 421, HOLLER/ILLING (2009), S. 189, RASMUSEN (2007), S. 21, PELEG/SUDHÖLTER (2007), S. 1, SIEG (2005), S. 76, AUMANN/MASCHLER (1995), S. 229, und ZARAGARE (1991), S. 71.

<sup>4</sup> Vgl. z.B. MCCAIN (2010), S. 403, PETERS (2008), S. 121, SIEG (2005), S. 76, OSTMANN (2002), S. 144, und AUMANN/MASCHLER (1995), S. 227 u. 229.

Kooperationspartner als nicht fair empfunden wird.<sup>5</sup> Eine von mindestens einem Kooperationspartner als unfair empfundene Verteilung der gemeinsam erwirtschafteten Effizienzgewinne führt in der Regel zu Problemen zwischen den Kooperationspartnern, die von beinträchtigttem Vertrauen<sup>6</sup> und ineffizienten Verhandlungen bis hin zum Scheitern der Kooperation führen können.<sup>7</sup> Die faire Verteilung der Effizienzgewinne ist somit ein wesentlicher Erfolgsfaktor einer Kooperation<sup>8</sup>; er beeinflusst maßgeblich den nachhaltigen Erfolg und das langfristige Bestehen einer Kooperation.<sup>9</sup>

Das Realproblem, den Effizienzgewinn einer Kooperation unter den Kooperationspartnern so zu verteilen, dass das Ergebnis der Verteilung von allen Kooperationspartnern als fair empfunden wird (kurz: das Verteilungsproblem) stellt ein strategisches Entscheidungsproblem dar. Da solche Entscheidungsprobleme Gegenstand der Spieltheorie<sup>10</sup> sind, liegt der Versuch nahe, das Verteilungsproblem mithilfe von spieltheoretischen Konzepten zu lösen.

In der vorliegenden Arbeit werden nur bewusst eingegangene, interorganisationale Kooperationen betrachtet. Es wird vorausgesetzt, dass die verschiedenen Entscheidungsträger untereinander kommunizieren<sup>11</sup> und Abmachungen treffen oder Verträge schließen können. Daher wird für die Lösung des Verteilungsproblems ein Konzept der kooperativen Spiel-

---

<sup>5</sup> An dieser Stelle sei darauf hingewiesen, dass unter dem Begriff Fairness keine objektive Eigenschaft einer Verteilung gemeint ist, sondern Fairness einen Imputationsbegriff darstellt. Fairness kann somit von den einzelnen Kooperationspartnern aufgrund ihrer subjektiven Empfindung einer Verteilung zugeschrieben werden, vgl. KLENDAUER ET AL. (2006), S. 189, und LIND/TYLER (1988), S. 3 f.

<sup>6</sup> Vertrauen zwischen Kooperationspartnern wird als einer der wichtigsten Erfolgsfaktoren von Kooperationen angesehen, vgl. z.B. TJADEN (2003), S. 104, HEROWITSCH (1997), S. 113 ff., EISELE (1995), S. 151 ff. (bezogen auf die Kooperationsform Joint Venture), und BAIRD ET AL. (1990), S. 129. Fairness wird als wichtiger Faktor zur Sicherung und zum Ausbau von Vertrauen angesehen, vgl. z.B. KOLLOGE (2010), S. 202.

<sup>7</sup> Vgl. z.B. FROMEN (2004), S. 38, GARBISCH/EDLER-PAIN (2004), S. 104, BALLING (1998), S. 133, YOUNG (1994), S. 1195 f., BLECKE/BULL-LARSEN/ERNST (1992), S. 116, und LEMAIRE (1984), S. 63. Auch MARWELL und SCHMITT bestätigen dies. Sie führen aber auch an, dass zahlreiche Studien zu dem Schluss gekommen sind, dass ein deutlich bevorzogter Kooperationspartner, bevor eine Kooperation aus den oben aufgeführten Gründen auseinanderbricht, stark benachteiligten Kooperationspartnern einen Ausgleich anbietet, um die Kooperation beizubehalten, vgl. MARWELL/SCHMITT (1975), S. 43 f. Für JÄHN ist eine Kooperation ohne gegenseitiges Vertrauen nicht denkbar, vgl. JÄHN (2008), S. 32 f. Zu dem Schluss, dass „unfaire“ Kooperationen zum Scheitern führen, kommen auch BLEEKE, BULL-LARSEN und ERNST anhand ihrer empirischen Untersuchung von strategischen Allianzen, vgl. BLEEKE/BULL-LARSEN/ERNST (1992), S. 116.

<sup>8</sup> Vgl. KOLLOGE (2010), S. 202.

<sup>9</sup> Vgl. FISCHER (2008), S. 24, BALLING (1998), S. 133, sowie JARILLO (1988), S. 36.

<sup>10</sup> Eine ausführliche Erläuterung des Gegenstandsbereichs der Spieltheorie findet sich beispielsweise bei HOLLER/ILLING (2006), S. 1.

<sup>11</sup> Es ist jedoch darauf hinzuweisen, dass eine Kommunikation zwischen Kooperationspartnern nicht immer stattfinden muss oder gewollt ist. Somit ist auch ein der Kooperation schädigendes Verhalten einzelner oder aller Kooperationspartner möglich. Zur Modellierung dieses Verhaltens wären Konzepte der nicht-kooperativen Spieltheorie erforderlich. Hier wird jedoch davon ausgegangen, dass alle Kooperationspartner die Kooperation bewusst eingegangen sind und ein langes Fortbestehen, sofern wirtschaftlich sinnvoll, anstreben.



theorie benötigt. Im Fokus des vorliegenden Arbeitsberichts liegt der  $\tau$ -Wert, ein Lösungskonzept der kooperativen Spieltheorie.

Es existieren zahlreiche Konzepte der kooperativen Spieltheorie. Jedoch werden diese kaum in der betrieblichen Praxis eingesetzt. Um die Wahrscheinlichkeit zu erhöhen, dass der  $\tau$ -Wert in der betrieblichen Praxis verwendet wird, wird in diesem Arbeitsbericht ein Programm vorgestellt, das die Berechnung des  $\tau$ -Werts durchführt und so dessen Einsatz in der Praxis vereinfacht.

## 2 Der $\tau$ -Wert

### 2.1 Erläuterung des $\tau$ -Werts

Das Konzept des  $\tau$ -Werts<sup>12</sup> wurde 1980 von TIJS auf dem „Seminar on game theory and mathematical economics“ erstmalig vorgestellt.<sup>13</sup> Zusammen mit DRIESSEN entwickelte er dieses Konzept in den darauf folgenden Jahren weiter.<sup>14</sup> Im Folgenden wird der  $\tau$ -Wert kurz vorgestellt. Da der Fokus des vorliegenden Arbeitsberichts auf der Implementierung des  $\tau$ -Werts liegt, werden nur die dafür notwendigen Eigenschaften des  $\tau$ -Werts erläutert.

Die Unternehmen, die miteinander kooperieren, werden in der Spieltheorie als Akteure bezeichnet. Ein Spiel hat  $N$  autonome Akteure  $A_n$  – oder auch synonym Spieler genannt – mit  $n = 1, \dots, N$ ,  $N \in \mathbb{N}$  und  $N \geq 2$ <sup>15</sup>, wobei  $\mathbb{N}$  die Menge aller natürlichen Zahlen ist.<sup>16</sup> Die Menge aller Akteure<sup>17</sup> ist  $A$ . Akteure können kooperieren und so Koalitionen bilden. Eine Koalition  $C_m$  stellt eine beliebige, nicht-leere<sup>18</sup> Teilmenge der Menge  $A$  aller Akteure dar.<sup>19</sup> Es gilt folglich:<sup>20</sup>

$$\emptyset \subset C_m \subseteq A \text{ mit } A = \{A_1, \dots, A_N\} \quad 2.1$$

<sup>12</sup> Der  $\tau$ -Wert wird in seltenen Fällen auch als TIJS-Wert oder im Englischen als TIJS value bezeichnet, vgl. z.B. BILBAO/LEBRÓN/JIMÉNEZ-LOSADA/TIJS (2001), S. 24.

<sup>13</sup> Vgl. TIJS (1980) S. 123 ff.

<sup>14</sup> Vgl. z.B. TIJS (2005), S. 7, DRIESSEN/TIJS (1990), S. 7 ff., DRIESSEN (1988), S. z.B. 57 ff. u. 79 ff., TIJS (1987), S. 187 ff., DRIESSEN (1987a), S. 209 ff., DRIESSEN/TIJS (1986), S. 13 ff., TIJS/DRIESSEN (1986), S. 10 ff., DRIESSEN/TIJS (1984), S. 252 ff., DRIESSEN/TIJS (1983), S. 1 ff., und DRIESSEN/TIJS (1983), S. 397 ff.

<sup>15</sup> Es ist ausgeschlossen, dass ein Spiel aus nur einem Akteur besteht, da in einem solchen Szenario das Problem der Verteilung von Effizienzgewinnen in Kooperationen nicht auftreten würde. Gibt es nur einen Akteur, liegt zum einen kein Verteilungsproblem vor, da dieser Akteur den gesamten Effizienzgewinn erhalten würde. Zum anderen würde kein Effizienzgewinn im Sinne eines „Mehrgewinns“ gegenüber alleinigem Handeln entstehen. Daher gilt hier  $N \geq 2$ , vgl. z.B. JENE/ZELEWSKI (2012), S. 170.

<sup>16</sup> Vgl. z.B. PETERS (2008b), S. 123.

<sup>17</sup> Synonym wird die Menge aller Akteure auch als Akteurmenge bezeichnet.

<sup>18</sup> Im Folgenden wird die sogenannte Null-Koalition, eine Koalition bestehend aus der leeren Menge, nicht zugelassen. Zwar würde die Zulassung einer Null-Koalition die charakteristische Funktion  $c$  vereinfachen, jedoch handelt es sich bei einer Null-Koalition um ein rein artifizielles Konstrukt, das keine Relevanz in der ökonomischen Realität hat. Aus diesem Grund wird die Null-Koalition nicht zur Menge der möglichen Koalitionen gezählt.

Ähnlicher Ansicht sind z.B. auch ZELEWSKI, SIEG, MYERSON GÜTH, OWEN und MOULIN, vgl. ZELEWSKI (2009), S. 79, SIEG (2005), S. 82, MYERSON (2001), S. 422, GÜTH (1999), S. 216 u. S. 247, OWEN (1995), S. 253, und MOULIN (1988), S. 90.

<sup>19</sup> Vgl. z.B. DRECHSEL (2010), S. 9 f., PARRACHINO/ZARA/PATRONE (2006), S. 7, DERKS/KUIPERS (2004), S. 88, SLIKKER/VAN DEN NOUWELAND (2001), S. 3, und SZÉP/FORGÓ (1983), S. 198.

<sup>20</sup> Vgl. z.B. JENE/ZELEWSKI (2012), S. 170, JENE/ZELEWSKI (2011a), S. 320, JENE/ZELEWSKI (2011b), S. 301, JENE/ZELEWSKI (2011c), S. 119, ZELEWSKI/JENE (2011), S. 3 und ZELEWSKI (2009), S. 79.

Eine Koalition, der sich alle Akteure angeschlossen haben, wird als große Koalition  $C_0$  bezeichnet.<sup>21</sup> Die charakteristische Funktion  $c$  verweist auf alle möglichen Koalitionen  $C_m$ .<sup>22</sup> Auch Koalitionen, die aus nur einem Akteur  $A_n$  bestehen, gehören zu diesen möglichen Koalitionen.<sup>23</sup> Die charakteristische Funktion  $c$  ordnet jeder dieser Koalitionen einen Betrag  $c(C_m)$  zu, den diese Koalition „begründet“ für sich in Anspruch nehmen kann.<sup>24</sup> Eine Auszahlung  $v(C_m)$  beschreibt, was eine Koalition  $C_m$  als Ergebnis des Spiels erhält. Der gemeinsam erwirtschaftete Effizienzgewinn einer Kooperation ist durch  $G$  angegeben, wobei gilt:  $G \in \mathbb{R}_{>0}$ .<sup>25</sup> Da der Effizienzgewinn  $G$  von allen Akteuren gemeinsam erwirtschaftet wurde, gilt folglich  $c(C_0)=G$ .

Der  $\tau$ -Wert ist ein sogenannter „Compromise Value“<sup>26</sup>. Compromise Values sind Lösungskonzepte der kooperativen Spieltheorie, deren Lösung auf der Definition von einer unteren und einer oberen Grenze für Auszahlungen der Akteure beruht.<sup>27</sup> Die Lösung ist die Kombination dieser beiden Grenzen.<sup>28</sup> Die Lösung kann somit als ein Kompromiss zwischen der unteren und der oberen Grenze angesehen werden.

Die obere Grenze des  $\tau$ -Werts wird durch den Anteil  $v_{n,max}$  des Effizienzgewinns  $G$  definiert, den ein Akteur  $A_n$  höchstens erhält. Dieser Anteil ist gleich dem marginalen Beitrags des Akteurs  $A_n$ . Ein marginaler Beitrag eines Akteurs  $A_n$  ist der Betrag der Wertsteigerung,

---

<sup>21</sup> Vgl. DRECHSEL (2010), S. 10, MCCAIN (2010), S. 409, MCCAIN (2009), S. 18 u. 31, PARRACHINO/ZARA/PATRONE (2006), S. 7, BRANZEI/DIMITROV/TIJS (2005b), S. 1, und SZÉP/FORGÓ (1983), S. 198 f.

<sup>22</sup> Vgl. DRECHSEL (2010), S. 10, PETERS (2008), S. 123, PARRACHINO/ZARA/PATRONE (2006), S. 7, SIEG (2005), S. 82, FROMEN (2004), S. 83, MYERSON (2001), S. 422, SLIKKER/VAN DEN NOUWELAND (2001), S. 4, und GÜTH (1999), S. 216.

<sup>23</sup> Vgl. JENE/ZELEWSKI (2012), S. 170, JENE/ZELEWSKI (2011a), S. 320, JENE/ZELEWSKI (2011b), S. 301, JENE/ZELEWSKI (2011c), S. 119, und ZELEWSKI (2009), S. 80.

<sup>24</sup> Vgl. PLATA-PÉREZ/SÁNCHEZ-PÉREZ (2011), S. 64, DRECHSEL (2010), S. 10, PETERS (2008), S. 229, WINTER (2002), S. 2028, SLIKKER/VAN DEN NOUWELAND (2001), S. 4, ZARAGARE (1991), S. 72, und SZÉP/FORGÓ (1983), S. 199.

<sup>25</sup> Vgl. JENE/ZELEWSKI (2012), S. 170, JENE/ZELEWSKI (2011a), S. 320, JENE/ZELEWSKI (2011b), S. 301, JENE/ZELEWSKI (2011c), S. 119, ZELEWSKI/JENE (2011), S. 3, und ZELEWSKI (2009), S. 74.

<sup>26</sup> Vgl. TIJS/OTTEN (1993), S. 2.

<sup>27</sup> Vgl. GILLIES (2010), S. 105 und TIJS/OTTEN (1993), S. 2.

<sup>28</sup> Vgl. GILLIES (2010), S. 105.

den die Koalition  $C_m$  durch den Beitritt von  $A_n$  erfährt. Der marginale Beitrag  $MB$  eines Akteurs  $A_n$  bezüglich der Koalition  $C_m$  kann formal wie folgt ausgedrückt werden:<sup>29</sup>

$$MB_n(c) = c(C_m) - c(C_m - \{A_n\}) \quad 2.2$$

Diese obere Grenze kann auch mithilfe der marginalen Koalition beschrieben werden. Die marginale Koalition wird definiert durch:

$$MC_n = C_0 - \{A_n\} = \{A_1, \dots, A_{n-1}, A_{n+1}, \dots, A_N\} \quad 2.3$$

Verwendet man diesen Term zur Definition der oberen Grenze  $OG$  des  $\tau$ -Werts und geht man dabei von der großen Koalition als Ausgangskoalition aus, folgt daraus:

$$\forall n = 1, \dots, N \quad \forall v_n \in \mathbb{R}_{\geq 0}: \quad v_{n,max} = c(C_0) - c(MC_n) = G - c(MC_n) \quad 2.4$$

Bei der Festlegung der unteren Grenze des  $\tau$ -Werts werden alle möglichen sogenannten Außenseiterkoalitionen betrachtet. Einem Akteur  $A_n$  wird mindestens der Anteil am Effizienzgewinn zugeteilt, mit dem er außerhalb der großen Koalition durch die Gründung einer Außenseiter-Koalition  $AC_{n,q}$ , glaubwürdig zu drohen vermag.<sup>30</sup> Dieser Mindestanteil am Effizienzgewinn  $G$  wird als  $v_{n,min}$  bezeichnet. Eine Außenseiter-Koalition ist eine Koalition  $AC_{n,q}$ , die aus ehemaligen Akteuren der großen Koalition besteht, welche glaubwürdig damit drohen können, die große Koalition zu verlassen, um außerhalb der großen Koalition eine größere Auszahlung zu erhalten. Jede Außenseiterkoalition besteht mindestens aus einem Akteur  $A_n$ , der als „Anführer“ agiert. Da ein Akteur  $A_n$  mehrere Außenseiterkoalitionen anführen kann, wird der zweite Index  $q$  benötigt, um die verschiedenen vom Akteur  $A_n$  angeführten Außenseiterkoalitionen zu differenzieren. Der Akteur  $A_n$ , der als Anführer einer Außenseiterkoalition  $AC_{n,q}$  agiert, bietet allen anderen Akteuren der Außenseiterkoalition  $AC_{n,q}$  eine Motivation, die große Koalition  $C_0$  zu verlassen. Diese Motivation besteht aus den Auszahlungen, die alle anderen Mitglieder der Außenseiterkoalition erhalten. Diese Auszahlungen der anderen Akteure in der betrachteten Außenseiterkoalition

<sup>29</sup> Vgl. LEYTON-BROWN/SOHAM (2008), S. 74, MORETTI/PATRONE (2008), S. 5, PETERS (2008), S. 241, PELEG/SÜDHÖLTER (2007), S. 161, HOLLER/ILLING (2006), S. 305, PARRACHINO/ZARA/PATRONE (2006), S. 18, BRANZEI/DIMITROV/TIJS (2005a), S. 4, BRANZEI/DIMITROV/TIJS (2005b), S. 6, WIESE (2005), S. 63 u. 197, DERKS/KUIPERS (2004), S. 89, NÚÑEZ/RAFELS (2002), S. 413, WINTER (2002), S. 2028, STUART (2001), S. 191, MYERSON (2001), S. 439, GÜTH (1992), S. 285, ROTH (1988), S. 6, DRIESSEN (1985), S. 15, DRIESSEN/TIJS (1985), S. 230, ROSENMÜLLER (1981), S. 257, sowie ROSENMÜLLER (1977), S. 11.

<sup>30</sup> Die Glaubwürdigkeit hängt immer von rein subjektiven Bewertungsgrundlagen und -maßstäben der anderen Akteure der großen Koalition ab. Ob die Drohung eines Akteurs glaubwürdig ist, kann folglich nicht objektiv bestimmt werden. Die mangelnde Objektivierbarkeit der Glaubwürdigkeit einer Drohung kann als ein Schwachpunkt des  $\tau$ -Werts festgehalten werden.

$AC_{n,q}$  entsprechen genau dem Nutzen, den die Akteure in der großen Koalition  $C_0$  maximal erhalten würden. Alle Akteure  $A_m$ , die Mitglied der Außenkoalition und nicht deren Anführer  $A_n$  sind, erhalten somit jeweils  $v_{m,max}$ . So haben die Akteure keinen Anreiz in der großen Koalition  $C_0$  zu verweilen. Der Anführer  $A_n$  der Außenseiterkoalition  $AC_{n,q}$  erhält den Betrag, den die betrachtete Außenseiterkoalition außerhalb der großen Koalition  $C_0$  erwirtschaften vermag, abzüglich der Seitenzahlungen an die weiteren Mitglieder  $A_m$  der Außenseiterkoalition.

Aus diesen Überlegungen ergibt sich die folgende Formel für untere Grenze  $UG$  des  $\tau$ -Werts:<sup>31</sup>

$$\forall n=1,\dots,N \forall v_n \in \mathbb{R}_{\geq 0} : v_n \geq v_{n,min} \wedge v_{n,min} = \max\{c_1; c_2; 0\} \quad 2.5$$

mit:

$$c_1 = c(\{A_n\} | AC_{n,q}) = c(\{A_n\}) \quad \text{für } AC_{n,q} = \{A_n\}$$

$$c_2 = \max \left\{ \begin{array}{l} c(\{A_n\} | AC_{n,q}) = c(AC_{n,q}) - \sum_{m \in (I_{n,q} \setminus \{n\})} v_{m,max} \quad \dots \\ \emptyset \subset AC_{n,q} \subset A \quad \wedge \quad \{A_n\} \subset AC_{n,q} \end{array} \right\} \quad 2.6$$

Der Term  $c_1$  stellt den Fall dar, dass die Außenseiterkoalition nur aus einem Akteur besteht. Sie kann aus der Formel (2.2.1) für die Bedingung der individuellen Rationalität abgeleitet werden. Aus diesem Grund müsste die Bedingung der individuellen Rationalität streng genommen nicht separat aufgeführt werden, da sie in der Formel für die untere Grenze des  $\tau$ -Werts implizit enthalten ist. Der Term  $c_2$  stellt alle anderen möglichen Fälle dar.

Es muss sichergestellt werden, dass der gemeinsam erwirtschaftete Effizienzgewinn  $G$  unter allen Akteuren  $A_n$  der großen Koalition  $C_0 = \{A_1, \dots, A_n\}$  also exakt aufgeteilt werden muss.<sup>32</sup> D.h., dass weder weniger noch mehr als der gemeinsam erwirtschaftete Effizienz-

<sup>31</sup> Vgl. JENE/ZELEWSKI (2012), S. 173 f., JENE/ZELEWSKI (2011a), S. 324, JENE/ZELEWSKI (2011b), S. 306, JENE/ZELEWSKI (2011c), S. 122, ZELEWSKI/PETERS (2010), S. 1, und ZELEWSKI (2009), S. 125 und 127.

<sup>32</sup> Vgl. JENE/ZELEWSKI (2012), S. 172, JENE/ZELEWSKI (2011a), S. 322, JENE/ZELEWSKI (2011b), S. 303, JENE/ZELEWSKI (2011c), S. 120, GILLIES (2010), S. 30, KOLOKOLTSOV/MALAFEYEV (2010), S. 168, ZELEWSKI/PETERS (2010), S. 7, ZELEWSKI (2009), S. 102, PETERS (2008b), S. 124 u. 230, ZELEWSKI/PETERS (2008), S. 6, WIESE (2005), S. 145, FROMEN (2004), S. 93 f., BORWARDT (2001), S. 596, und MYERSON (2001), S. 428.

gewinn  $G$  verteilt werden kann. Denn zum einen wäre es irrational, weniger als den Effizienzgewinn  $G$  zu verteilen, da alle Akteure ihren Nutzen maximieren wollen und aus diesem Grund keinen noch so kleinen Rest von  $G$  unverteilt lassen würden. Weniger zu verteilen würde somit gegen die Prämisse verstoßen, dass alle Akteure rational handeln. Anders betrachtet würden rational handelnde Akteure eine solche Verteilung nicht akzeptieren. Übertragen auf das zugrundeliegende Realproblem bedeutet dies, dass der Effizienzgewinn eines Ermittlungszeitraums und auch kein Teil davon gespart werden kann, sondern vollständig zeitnah an die Kooperationspartner verteilt werden muss. Zum anderen ist es unmöglich, mehr zu verteilen als vorhanden ist.<sup>33</sup> Eine exakte Verteilung des Effizienzgewinns  $G$  wird auch als effizient bezeichnet.

Daraus resultiert, dass für jede zulässige Lösung  $L$  des Verteilungsproblems und den Wert  $c(C_0)$  der charakteristischen Funktion  $c$  für die große Koalition  $C_0$  folgende Effizienzbedingung gelten muss:<sup>34</sup>

$$\forall n=1,\dots,N \forall v_n \in \mathbb{R}_{\geq 0} : \sum_{n=1}^N v_n = c(C_0) = G \quad 2.7$$

Zusätzlich muss beachtet werden, dass Lösungspunkte nur zwischen der unteren Grenze  $UG$  und der Oberen Grenze  $OG$  liegen können. Die Integritätsbedingung sichert die Existenz einer Lösung. Daraus folgt in abgekürzter Schreibweise folgende Bedingung für alle möglichen Lösungspunkte:<sup>35</sup>

$$\forall L \in \mathbb{R}_{\geq 0}^N : UG \leq L \leq OG \quad 2.8$$

---

<sup>33</sup> Vgl. z.B. PLATA-PÉREZ/SÁNCHEZ-PÉREZ (2011), S. 62.

In diesem Fall müsste entweder ein Kredit aufgenommen oder auf Rücklagen zurückgegriffen werden. In diesem Fall würde mindestens ein Akteur, und zwar derjenige, der einen Kredit aufnimmt oder Teile seiner Rücklagen nutzt, einen Verlust machen. Dies verstößt gegen die Annahme, dass alle Akteure rational handeln.

<sup>34</sup> Vgl. JENE/ZELEWSKI (2012), S. 172, JENE/ZELEWSKI (2011a), S. 322, JENE/ZELEWSKI (2011b), S. 304, JENE/ZELEWSKI (2011c), S. 120, GILLIES (2010), S. 30, KOLOKOLTSOV/MALAFEYEV (2010), S. 168, ZELEWSKI/PETERS (2010), S. 7, PETERS (2008), S. 124 u. 230, ZELEWSKI/PETERS (2008), S. 6, ZELEWSKI (2008), S. 102 f., WIESE (2005), S. 145, FROMEN (2004), S. 93 f., BILBAO/LEBRÓN/JIMÉNEZ-LOSADA/TIJS (2001), S. 23, BORGWARDT (2001), S. 596, und MYERSON (2001), S. 428.

<sup>35</sup> Vgl. ZELEWSKI (2009), S. 128.

Aus dieser Bedingung kombiniert mit der Effizienzbedingung folgt die folgende Integritätsbedingung:<sup>36</sup>

$$\begin{aligned} & \forall UG, OG \in \mathbb{R}_{\geq 0}^N \forall G \in \mathbb{R}_{> 0} : \\ & \left( UG = \begin{pmatrix} v_{1.min} \\ \dots \\ v_{N.min} \end{pmatrix} \wedge OG = \begin{pmatrix} v_{1.max} \\ \dots \\ v_{N.max} \end{pmatrix} \wedge c(C_0) = G \right) \\ & \rightarrow \left( \sum_{n=1}^N v_{n.min} \leq G \leq \sum_{n=1}^N v_{n.max} \wedge UG \leq OG \right) \end{aligned} \quad 2.9$$

Die Berechnung des  $\tau$ -Werts kann mithilfe der folgenden aus den zuvor aufgeführten Bedingungen resultierenden Formel erfolgen:<sup>37</sup>

$$\forall n=1, \dots, N: v_{n,\tau} = \gamma \cdot v_{n,max} + (1-\gamma) \cdot v_{n,min}$$

mit:

$$\begin{aligned} \gamma &= \frac{G - \sum_{n=1}^N v_{n.min}}{\sum_{n=1}^N v_{n.max} - \sum_{n=1}^N v_{n.min}} ; & \text{falls } \sum_{n=1}^N v_{n.max} \neq \sum_{n=1}^N v_{n.min} \\ \gamma &\in [0; 1]; & \text{falls } \sum_{n=1}^N v_{n.max} = \sum_{n=1}^N v_{n.min} \end{aligned} \quad 2.10$$

---

<sup>36</sup> Vgl. JENE/ZELEWSKI (2012), S. 174, JENE/ZELEWSKI (2011a), S. 324, JENE/ZELEWSKI (2011b), S. 306, JENE/ZELEWSKI (2011c), S. 123, ZELEWSKI/PETERS (2010), S. 9, ZELEWSKI (2009), S. 138, und ZELEWSKI (2007), S. 564.

<sup>37</sup> Vgl. JENE/ZELEWSKI (2012), S. 175, JENE/ZELEWSKI (2011a), S. 328, JENE/ZELEWSKI (2011b), S. 307, JENE/ZELEWSKI (2011c), S. 123, ZELEWSKI/PETERS (2010), S. 11, und ZELEWSKI (2009), S. 154.

## 2.2 Pseudocode zur Berechnung des $\tau$ -Werts

Der nachfolgende Pseudocode wurde von ZELEWSKI<sup>38</sup> übernommen. Die Pseudocode-Notation basiert auf einfachen Operatoren, die zum Teil in verschiedenen Programmiersprachen Anwendung finden. Zur Hervorhebung der Operatoren werden diese in Großbuchstaben dargestellt. Die hier verwendeten Operatoren werden im Folgenden kurz erläutert:

- **BEGIN:** BEGIN bezeichnet den Anfang einer Anweisung.
- **END:** END bezeichnet das Ende einer Anweisung.
- **INPUT:** INPUT gibt an, welche Informationen zu Beginn der Durchführung des Berechnungsalgorithmus bekannt sein müssen. Diese Spezifizierung des Inputs beginnt mit dem Operator INPUT, gefolgt vom Algorithmusnamen und dem Operator BEGIN. Das Ende der Spezifizierung des Inputs wird durch den Operator INPUT, gefolgt vom Algorithmusnamen und dem Operator END festgelegt.
- **OUTPUT:** OUTPUT gibt an, welche Informationen der Berechnungsalgorithmus ausgibt. Diese Spezifizierung des Outputs beginnt mit dem Operator OUTPUT, gefolgt vom Algorithmusnamen und dem Operator BEGIN. Das Ende der Spezifizierung des Outputs wird durch den Operator OUTPUT, gefolgt vom Algorithmusnamen und dem Operator END festgelegt.
- **TASK:** TASK identifiziert eine Anweisung, die unabhängig von anderen Anweisungen ausgeführt werden kann. Eine solche Anweisung beginnt mit dem Operator TASK, gefolgt vom Anweisungsnamen und dem Operator BEGIN.
- **ENDTASK:** Das Ende einer Anweisung wird durch dem Operator ENDTASK, gefolgt vom Anweisungsnamen festgelegt.

---

<sup>38</sup> Vgl. ZELEWSKI (2009), S. 250 ff.

Nur im Bereich des Integritätstests (Existenz einer Lösung) wurden Anpassungen im Vergleich zum Original vorgenommen.



- DO: DO gefolgt vom Anweisungsnamen und BEGIN identifiziert Anweisungen, die sequenziell ausgeführt werden.
- ENDDO: ENDDO gefolgt vom Anweisungsnamen bezeichnet das Ende einer DO-Anweisung.
- DO-PARALLEL: DO-PARALLEL gefolgt vom Anweisungsnamen und dem Operator BEGIN zeigt an, dass mehrere darauffolgende TASK-Anweisung unabhängig von anderen Anweisungen, also nebenläufig ausgeführt werden können.
- LOOP: LOOP gefolgt vom Anweisungsnamen und dem Operator BEGIN und identifiziert den Anfang einer Iteration.
- FOR: FOR definiert den Zähler in einer LOOP-Iteration. Auf dem Operator FOR folgt die Festlegung einer Variablen gefolgt von dem Startwert dieser Variablen, dem Wert, um den sich diese Variable bei jeder Iteration erhöht, und dem Endwert, bei dessen Erreichen die Iteration abgebrochen wird.
- ENDLOOP: ENDLOOP identifiziert das Ende einer Iteration.
- IF-THEN: IF wird gefolgt von einer Bedingung. Wenn diese Bedingung erfüllt wird, wird die Anweisung, die auf den Operator THEN folgt, ausgeführt.
- ELSE: ELSE gefolgt von einer Anweisung ist eine optionale Erweiterung der Bedingung, die durch IF-THEN entsteht. Die Anweisung wird ausgeführt, wenn die IF-Bedingung nicht erfüllt wird.
- ENDIF: ENDIF gefolgt von einer Bedingung beendet eine IF-Anweisung.
- PRINT: Die Anweisung, die auf den Operator PRINT folgt, gibt an, welche Informationen ausgegeben werden.

Der Berechnungsalgorithmus wird von oben nach unten durchlaufen, d.h., dass die Anweisungen sequenziell ausgeführt werden.<sup>39</sup> Die Erläuterung der einzelnen Anweisungen erfolgt in den jeweiligen Fußnoten.

---

<sup>39</sup> Dies gilt nicht für ineinander verschachtelte Anweisungen.

```

INPUT     $\tau$ -Wert-Berechnung BEGIN

    Akteuremenge:  $A = \{A_1, \dots, A_N\}$ 

    zu verteiler Effizienzgewinn: G

    notwendige Werte der charakteristischen Funktion:  $c(C_0), c(C_0 \setminus \{A_1\}), \dots, c(C_0 \setminus \{A_N\})$ 

INPUT     $\tau$ -Wert-Berechnung END

Fehleranzeiger := „nein“40

Fehlermeldung := []

DO PARALLEL Integritätstests-Inputwerte BEGIN41

    TASK    Integritätstest-Akteuremenge BEGIN

        N := |A|

        IF N < 242

            THEN Fehleranzeiger := „ja“

                IF Fehlermeldung = []

                    THEN Fehlermeldung := ...

                        Fehlermeldung  $\oplus$  [„zu kleine Akteuremenge“]

                    ELSE Fehlermeldung := ...

                        Fehlermeldung  $\oplus$  [„/ zu kleine Akteuremenge“]

                ENDIF Fehlermeldung = []

            ENDIF N < 2

        ENDTASK Integritätstest-Akteuremenge

    TASK    Integritätstest-Effizienzgewinn BEGIN43

        IF G  $\leq$  0

```

---

<sup>40</sup> Die Variable „Fehleranzeiger“ wird zu Beginn des Berechnungsalgorithmus auf „nein“ gesetzt. Ebenso ist die Variable Fehlermeldung zu Beginn leer.

<sup>41</sup> Die Anweisung Integritätstests-Inputwerte überprüft, ob die eingegebenen Werte die notwendigen Voraussetzungen einhalten, die sicherstellen, dass eine Lösung existiert. Welche Voraussetzungen das sind, wird in den folgenden Fußnoten erläutert.

<sup>42</sup> Der erste Integritätstest bezüglich der Inputwerte bezieht sich auf die Anzahl der Akteure. Es müssen mindestens zwei Akteure beteiligt sein, damit ein Verteilungsproblem vorliegt. Falls weniger als zwei Akteure beteiligt sind, wird der Fehleranzeiger auf „ja“ gesetzt.

<sup>43</sup> Der zweite Integritätstest bezüglich der Inputwerte bezieht sich auf den Effizienzgewinn G. Der Effizienzgewinn G muss größer als 0 sein, damit überhaupt ein Effizienzgewinn G vorliegt, der verteilt werden kann. Falls der Effizienzgewinn G kleiner als oder gleich 0 ist, wird der Fehleranzeiger auf „ja“ gesetzt.

```

THEN      Fehleranzeiger := „ja“

      IF Fehlermeldung = []

      THEN      Fehlermeldung := ...
                Fehlermeldung  $\oplus$  [„kein verteilter Effizienzgewinn“]

      ELSE Fehlermeldung := ...
                Fehlermeldung  $\oplus$  [„ / kein verteilter Effizienzgewinn“]

      ENDIF Fehlermeldung = []

    ENDIF G < 0

  ENDTASK Integritätstest-Effizienzgewinn

TASK      Integritätstest-charakteristische-Funktion-große-Koalition BEGIN44

  IF  $c(C_0) \neq G$ 

  THEN      Fehleranzeiger := „ja“

            IF Fehlermeldung = []

            THEN      Fehlermeldung := Fehlermeldung  $\oplus$  ...
                      [„große Koalition erhält nicht exakt den Effizienzgewinn“]

            ELSE Fehlermeldung := Fehlermeldung  $\oplus$  ...
                      [„ / große Koalition erhält nicht exakt den Effizienzgewinn“]

            ENDIF Fehlermeldung = []

        ENDIF  $c(C_0) \neq G$ 

    ENDTASK Integritätstest-charakteristische-Funktion-große-Koalition

  ENDDO Integritätstests-Inputwerte

  IF Fehleranzeiger = „ja“

  THEN      QUIT

  ENDIF Fehleranzeiger = „ja“

```

---

<sup>44</sup> Der dritte Integritätstest bezüglich der Inputwerte bezieht sich auf den Wert der großen Koalition  $c(C_0)$ . Der Wert der großen Koalition  $c(C_0)$  muss gleich dem Effizienzgewinn  $G$  sein. Falls der Wert der großen Koalition  $c(C_0)$  nicht gleich dem Effizienzgewinn  $G$  ist, wird der Fehleranzeiger auf „ja“ gesetzt.

DO Vorsortierung-Akteure BEGIN<sup>45</sup>

Menge-maßgeblicher-Akteure :=  $\emptyset$

Menge-nicht-maßgeblicher-Akteure :=  $\emptyset$

LOOP Ermittlung-maßgeblicher-Akteure BEGIN

FOR  $n = 1 < I > N$

Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur- $n$  :=  $\emptyset$

Menge-unwesentlicher-Außenseiterkoalitionen-für-Akteur- $n$  :=  $\emptyset$

ermittle die Menge  $\{AC_{n,q} | q=1, \dots, Q_n\}$  aller Außenseiterkoalitionen  $AC_{n,q}$ , die aus mehreren Akteuren bestehen und vom Akteur  $A_n$  angeführt werden

LOOP Ermittlung-wesentlicher-Außenseiterkoalitionen-für-Akteur- $n$  BEGIN

UNTIL  $\{AC_{n,q} | q=1, \dots, Q_n\} = \emptyset$

wähle eine beliebige Außenseiterkoalition  $AC_{n,r}$  aus  $\{AC_{n,q} | q=1, \dots, Q_n\}$  aus

schätze für den Wert  $c(\{A_n\} | AC_{n,r})$  der charakteristischen Funktion  $c$

grob die Größenordnung  $\hat{c}(\{A_n\} | AC_{n,r})$  ab, ob diese Größenordnung

entweder  $\hat{c}(\{A_n\} | AC_{n,r}) > 0$  oder aber  $\hat{c}(\{A_n\} | AC_{n,r}) \leq 0$  erfüllt

IF  $\hat{c}(\{A_n\} | AC_{n,r}) > 0$

THEN Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur- $n$  := ...

Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur- $n \cup \{AC_{n,r}\}$

ELSE Menge-unwesentlicher-Außenseiterkoalitionen-für-Akteur- $n$  := ...

Menge-unwesentlicher-Außenseiterkoalitionen-für-Akteur- $n \cup \{AC_{n,r}\}$

ENDIF  $\hat{c}(\{A_n\} | AC_{n,r}) > 0$

<sup>45</sup> Diese Anweisung dient dazu, den Aufwand der Ermittlung der Werte der charakteristischen Funktion  $c$  zu verringern. Dazu werden die maßgeblichen Akteure identifiziert. Maßgebliche Akteure sind Akteure, die überhaupt glaubwürdig damit drohen können, eine Außenseiterkoalition zu gründen. Akteure, die nicht als maßgeblich identifiziert, wurden werden als nicht-maßgebliche Akteure bezeichnet.

Zur Identifizierung der maßgeblichen Akteure werden für jeden Akteur alle möglichen Außenseiterkoalitionen durchlaufen, um die wesentlichen Außenseiterkoalitionen zu identifizieren. Eine Außenseiterkoalition wird als wesentliche Außenseiterkoalition bezeichnet, wenn nach einer Überschlagsrechnung der geschätzte Wert  $\hat{c}(\{A_n\} | AC_{n,q}) > 0$  ist. In diesem Fall handelt es sich um eine Außenseiterkoalition mit der der Akteur  $A_n$  glaubwürdig drohen kann. Außenseiterkoalitionen die nicht als wesentlich identifiziert wurden, werden als unwesentliche Außenseiterkoalitionen bezeichnet.

Bei der Identifizierung der maßgeblichen Akteure sowie der wesentlichen Außenseiterkoalitionen wird auf Schätzungen für die jeweiligen Werte der charakteristischen Funktion  $c$  zurückgegriffen. Diese Schätzungen erfolgen durch den Anwender. Der Einfachheit halber werden diese durch den Anwender getätigten Eingaben nicht durch zusätzliche Input-Anweisungen dargestellt, sondern werden als bekannt vorausgesetzt.

$$\{AC_{n,q} \mid q=1, \dots, Q_n\} := \{AC_{n,q} \mid q=1, \dots, Q_n\} \setminus \{AC_{n,r}\}$$

ENDLOOP Ermittlung-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  
 IF Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\neq \emptyset$   
 THEN Menge-maßgeblicher-Akteure :=  
     Menge-maßgeblicher-Akteure  $\cup \{A_n\}$   
 ELSE Menge-nicht-maßgeblicher-Akteure :=  
     Menge-nicht-maßgeblicher-Akteure  $\cup \{A_n\}$   
 ENDIF Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\neq \emptyset$   
 ENDLOOP Ermittlung-maßgeblicher-Akteure  
 ENDDO Vorsortierung-Akteure  
 LOOP Ermittlung-obere-Grenze- $\tau$ -Wert BEGIN<sup>46</sup>  
     FOR n = 1<1>N  
          $v_{n,max} := G - c(C_0 \setminus \{A_n\})$   
     ENDLOOP Ermittlung-obere-Grenze- $\tau$ -Wert  
 LOOP Ermittlung-untere-Grenze- $\tau$ -Wert BEGIN<sup>47</sup>  
     FOR n = 1<1>N  
         DO Ermittlung-untere-Grenze-für-Akteur-n BEGIN  
             IF  $A_n \in$  Menge-maßgeblicher-Akteure  
                 THEN DO Ermittlung-individueller-Drohwert BEGIN  
                      $c_n := -\infty$   
                     IF  $\{A_n\} \in$  Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  
                         THEN ermittle<sup>48</sup> exakten Wert  $c(\{A_n\})$   
                              $c_n := c(\{A_n\})$   
                             Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n := ...  
                             Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\setminus \{A_n\}$   
                         ENDIF  $\{A_n\} \in$  Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  
             END DO Ermittlung-individueller-Drohwert BEGIN  
         END DO Ermittlung-untere-Grenze-für-Akteur-n BEGIN  
     END FOR n = 1<1>N  
 END LOOP Ermittlung-untere-Grenze- $\tau$ -Wert

<sup>46</sup> Diese Anweisung dient zu Ermittlung der oberen Grenze OG gemäß Formel 2.4.

<sup>47</sup> Diese Anweisung dient zu Ermittlung der unteren Grenze UG gemäß Formel 2.5.

<sup>48</sup> Mit „ermittle“ ist hier die Ermittlung des „exakten“ Werts  $c(\{A_n\})$  durch den Anwender gemeint, der dann durch den Anwender eingegeben wird. Der Einfachheit halber wird hier auf einer weitere Input-Anweisung verzichtet.

```

ENDDO Ermittlung-individueller-Drohwert
DO Ermittlung-kollektiver-Drohwert BEGIN
   $d_n := -\infty$ 
  LOOP Ermittlung-für-wesentliche-Außenseiterkoalitionen BEGIN
    UNTIL Menge-wesentlicher-Außenseiterkoalitionen-
           für-Akteur-n =  $\emptyset$ 
    wähle eine beliebige Außenseiterkoalition
     $AC_{n,q} \in$  Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n
    ermittle49 exakten Wert  $c(AC_{n,q})$ 
    
$$c(\{A_n\} | AC_{n,q}) := c(AC_{n,q}) - \sum_{m \in (N_{n,q} \setminus \{n\})} v_{m,max}$$

    Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n := ...
    Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\setminus$   $\{AC_{n,q}\}$ 
  ENDLOOP Ermittlung-für-wesentliche-Außenseiterkoalitionen
  IF Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\neq \emptyset$ 
  THEN  $d_n := \max \{ c(\{A_n\} | AC_{n,q}) \mid \dots$ 
        $AC_{n,q} \in$  Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n
  ENDIF Menge-wesentlicher-Außenseiterkoalitionen-für-Akteur-n  $\neq \emptyset$ 
ENDDO Ermittlung-kollektiver-Drohwert
 $v_{n,min} := \max \{0, c_n, d_n\}$ 
ELSE  $v_{n,min} := 0$ 
ENDIF  $A_n \in$  Menge-maßgeblicher-Akteure
ENDDO Ermittlung-untere-Grenze-für-Akteur-n
ENDLOOP Ermittlung-untere-Grenze- $\tau$ -Wert
Fehleranzeiger := „nein“50

```

<sup>49</sup> Mit „ermittle“ ist hier die Ermittlung des „exakten“ Werts  $c(AC_{n,q})$  durch den Anwender gemeint, der dann durch den Anwender eingegeben wird. Der Einfachheit halber wird hier auf einer weitere Input-Anweisung verzichtet.

<sup>50</sup> Hier wurde eine Abweichung zum Original vorgenommen. Im Original Pseudo-Code wird der Fehleranzeiger auf ja gesetzt. Dies führt jedoch dazu, dass selbst wenn im Folgenden kein Fehler auftritt, der Berechnungsalgorithmus bei der nächsten Überprüfung des Fehleranzeigers abgebrochen wird. Aus diesem Grund wurde der Fehleranzeiger hier auf nein gesetzt.

Fehlermeldung := []

DO PARALLEL Integritätstest-Existenz-einer-Lösung BEGIN

TASK Integritätstest-Verhältnis-Summe-untere-Grenzen-und-Effizienzgewinn BEGIN<sup>51</sup>

IF  $\sum_{n=1}^N v_{n,\min} > G$ <sup>52</sup>

THEN Fehleranzeiger := „ja“

IF Fehlermeldung = []

THEN Fehlermeldung := Fehlermeldung  $\oplus$  ...

[„Summe untere Grenzen größer als der Effizienzgewinn“]

ELSE Fehlermeldung := Fehlermeldung  $\oplus$  ...

[„ / Summe untere Grenzen größer als der Effizienzgewinn“]

ENDIF Fehlermeldung = []

ENDIF  $\sum_{n=1}^N v_{n,\min} > G$

ENDTASK Integritätstest-Verhältnis-Summe-untere-Grenzen-und-Effizienzgewinn

TASK Integritätstest-Verhältnis-Summe-obere-Grenzen-und-Effizienzgewinn BEGIN<sup>53</sup>

IF  $\sum_{n=1}^N v_{n,\max} < G$ <sup>54</sup>

THEN Fehleranzeiger := „ja“

IF Fehlermeldung = []

THEN Fehlermeldung := Fehlermeldung  $\oplus$  ...

[„Summe obere Grenzen kleiner als der Effizienzgewinn“]

ELSE Fehlermeldung := Fehlermeldung  $\oplus$  ...

[„ / Summe obere Grenzen kleiner als der Effizienzgewinn“]

ENDIF Fehlermeldung = []

<sup>51</sup> Dieser weitere Integritätstest überprüft, ob die Summe der Werte  $v_{n,\min}$  aller Akteure größer ist als der Effizienzgewinn  $G$ . Ist dies der Fall, wird der Fehleranzeiger auf „ja“ gesetzt. Denn die Summe der Mindestanteile  $v_{n,\min}$  aller Akteure darf nicht größer sein als der zu verteilende Effizienzgewinn  $G$ .

<sup>52</sup> An dieser Stelle wurde eine weitere Änderung gegenüber dem Original vorgenommen. Im Original wurde folgende Bedingung  $v_{n,\max} < G$  überprüft. Gemäß der Integritätsbedingung (Formel 2.9) muss jedoch  $\sum_{n=1}^N v_{n,\max} < G$  überprüft werden.

<sup>53</sup> Dieser weitere Integritätstest überprüft, ob die Summe der Werte  $v_{n,\max}$  aller Akteure kleiner ist als der gemeinsam erwirtschaftete Effizienzgewinn  $G$ . Ist dies der Fall, wird der Fehleranzeiger auf „ja“ gesetzt. Denn die Summe der maximal zurechenbaren Anteile  $v_{n,\max}$  aller Akteure, darf nicht kleiner sein als der zu verteilende Effizienzgewinn  $G$ .

<sup>54</sup> An dieser Stelle wurde eine weitere Änderung gegenüber dem Original vorgenommen. Im Original wurde folgende Bedingung  $v_{n,\max} <$  überprüft. Gemäß der Integritätsbedingung (Formel 2.9) muss jedoch  $\sum_{n=1}^N v_{n,\max} < G$  überprüft werden.

```

ENDIF  $\sum_{n=1}^N v_{n,\max} < G$ 

ENDTASK Integritätstest-Verhältnis-Summe-obere-Grenzen-und-Effizienzgewinn

TASK Integritätstest-Verhältnis-untere-zu-obere-Grenze-für-jeden-Akteur BEGIN55

    LOOP    Test-für-jeden-Akteur BEGIN

        FOR n = 1<1>N

            DO    Integritätstest-Verhältnis-untere-und-obere-Grenze BEGIN

                IF  $v_{n,\min} > v_{n,\max}$ 

                    THEN    Fehleranzeiger := „ja“

                        IF Fehlermeldung = []

                            THEN    Fehlermeldung := Fehlermeldung  $\oplus$  ...
                                    [„untere Grenze größer als obere Grenze für Akteur n“]

                            ELSE Fehlermeldung := Fehlermeldung  $\oplus$  ...
                                    [„ / untere Grenze größer als obere Grenze für Akteur n“]

                            ENDIF Fehlermeldung = []

                        ENDIF  $v_{n,\min} > v_{n,\max}$ 

                    ENDDO Integritätstest-Verhältnis-untere-und-obere-Grenze

                ENDLLOOP Test-für-jeden-Akteur

            ENDTASK Integritätstest-Verhältnis-untere-zu-obere-Grenze-für-jeden-Akteur

        ENDDO Integritätstest-Existenz-einer-Lösung

    IF Fehleranzeiger := „ja“

        THEN    QUIT

    ENDIF Fehleranzeiger := „ja“

```

---

<sup>55</sup> Dieser weitere Integritätstest überprüft für jeden Akteur  $A_n$ , ob der mindestens zurechenbare Anteil  $v_{n,\min}$  größer als der maximal zurechenbare Anteil  $v_{n,\max}$  ist, ob also folgende Bedingung erfüllt wird:  $v_{n,\min} > v_{n,\max}$ . Ist diese Bedingung erfüllt, widerspricht dies gegen die Integritätsbedingung gemäß Formel 2.8 und der Fehleranzeiger wird auf „ja“ gesetzt.



DO Berechnung- $\tau$ -Wert BEGIN<sup>56</sup>

$$\gamma := \begin{cases} \frac{G - \sum_{n=1}^N v_{n,\min}}{\sum_{n=1}^N v_{n,\max} - \sum_{n=1}^N v_{n,\min}} & ; \text{ falls } \sum_{n=1}^N v_{n,\max} \neq \sum_{n=1}^N v_{n,\min} \\ 0,5 & ; \text{ falls } \sum_{n=1}^N v_{n,\max} = \sum_{n=1}^N v_{n,\min} \end{cases}$$

LOOP Berechnung- $\tau$ -Wert-Komponenten BEGIN

FOR n = 1 < 1 > N

$$v_{n,\tau} := \gamma \cdot v_{n,\max} + (1 - \gamma) \cdot v_{n,\min}$$

ENDLOOP Berechnung- $\tau$ -Wert-Komponenten BEGIN

ENDDO Berechnung- $\tau$ -Wert

OUTPUT  $\tau$ -Wert-Berechnung BEGIN<sup>57</sup>

IF Fehleranzeiger = „nein“

THEN PRINT „Komponenten des  $\tau$ -Werts für die Instanz des generischen Verteilungsproblems mit der Akteuremenge “ A „ und “ N „ Akteuren: “

DO Ausgabe-der-Komponenten-des- $\tau$ -Werts BEGIN

FOR n = 1 < 1 > N

PRINT „ $v_{n,\tau}$  = “  $v_{n,\tau}$

ENDDO Ausgabe-der-Komponenten-des- $\tau$ -Werts

ELSE PRINT Fehlermeldung

ENDIF Fehleranzeiger = „nein“

OUTPUT  $\tau$ -Wert-Berechnung END

Dieser im Vorangegangenen vorgestellte Pseudo-Code dient als Grundlage für die im Folgenden vorgestellte Implementierung des  $\tau$ -Werts.

<sup>56</sup> Abweichend von Formel 2.10 wird hier für den Sonderfall  $\sum_{n=1}^N v_{n,\max} = \sum_{n=1}^N v_{n,\min}$  der Wert 0,5 festgelegt, um eine eindeutige Anweisung zu formulieren. Statt 0,5 hätte auch jeder andere Wert des Intervalls  $[0;1]$  gewählt werden können.

<sup>57</sup> Die Output-Anweisung gibt an, dass die Auszahlung  $v_{n,\tau}$  jedes Akteurs  $A_n$  als Ergebnis der Berechnung ausgegeben wird.

## 3 Implementierung des $\tau$ -Werts

### 3.1 Zielsetzung

Ziel der Implementierung des zuvor dargestellten Pseudocodes zur Berechnung des  $\tau$ -Werts ist die Entwicklung eines Programms, das durch die Eingabe der zu Beginn der Pseudocodes unter Input aufgeführten Daten (Anzahl der Akteure, zu verteiler Effizienzgewinn und notwendige Werte der charakteristischen Funktion) den  $\tau$ -Wert ermittelt. Der Programmcode orientiert sich weitestgehend an dem in Kapitel 2.2 vorgestellten Pseudocode. Es wurde abweichend vom Pseudo-Code davon ausgegangen, dass alle Werte der charakteristischen Funktion vorliegen und zu Beginn durch den Anwender eingegeben werden. Daher basiert die Unterscheidung in maßgebliche und nicht-maßgebliche Außen-seiterkoalitionen sowie die Unterscheidung in wesentliche und unwesentliche Akteure nicht basierend auf Schätzungen durchgeführt, sondern anhand der zu Beginn eingegebenen Werte für die charakteristische Funktion.

### 3.2 Programmcode

```
/*  
 * AkteureTauwert  
 */  
  
package tauwert.algo;  
  
import java.util.ArrayList;  
  
public class AkteureTauwert extends TauwertAlgorithmus {  
  
    Akteur[] akteure;  
    Double effizienzGewinn;  
  
    public AkteureTauwert(Integer[] akteureMenge, Double effGewinn,  
        Double[] charWert, CharacterFunction charakterFunktion) {  
  
        integritaetstest(effGewinn, akteureMenge.length, charWert);  
  
        effizienzGewinn = effGewinn;  
        Akteur.akteure.clear();  
        akteure = new Akteur[akteureMenge.length];  
  
        for(int i = 0; i < akteureMenge.length; i++) {  
            akteure[i] = new Akteur(i+1, charWert[i+1], charakterFunktio-  
on, effGewinn, akteureMenge.length);  
        }  
  
        for(Akteur akteur : akteure) {  
            akteur.vorsortiere(charakterFunktion);  
        }  
    }  
}
```

```

        for(Akteur akteur : akteure) {
            akteur.calcVmin();
        }

        for(Akteur akteur : akteure) {
            System.out.println(akteur);
        }

        balanceTest(akteure, effGewinn);
    }

    private void integritaetstest(Double effizienzGewinn, int akteure,
    Double[] charWert) {
        String errstr = "";
        boolean errocc = false;

        if(akteure < 2) {
            errocc = true;
            errstr += (errstr.length() == 0)?
                "Zu kleine Akteuremenge" :
                " / zu kleine Akteuremenge";
        }

        if(effizienzGewinn <= 0) {
            errocc = true;
            errstr += (errstr.length() == 0)?
                "Kein verteilter Effizienzgewinn" :
                " / kein verteilter Effizienzgewinn";
        }

        if(charWert[0] != effizienzGewinn) {
            errocc = true;
            errstr += (errstr.length() == 0)?
                "Große Koalition erhält nicht exakt den Effizienzge-
winn" :
                " / große Koalition erhält nicht exakt den Effizienz-
gewinn";
        }

        if(errocc)
            throw new IllegalArgumentException(errstr);
    }

    private void balanceTest(Akteur[] akteure, Double effizienzGewinn) {
        String errstr = "";
        boolean errocc = false;

        for(Akteur akteur : akteure) {
            if(akteur.getVmin() > akteur.getVmax()) {
                errocc = true;
                errstr += ((errstr.length() == 0)?
                    "Untere Grenze größer als obere Grenze für Akteur
" :
                    " / untere Grenze größer als obere Grenze für Ak-
teur ")
                    + akteur.getIndex();
            }
        }
    }

```

```

        if(akteur.getVmin() > effizienzGewinn) {
            errocc = true;
            errstr += ((errstr.length() == 0)?
                "Untere Grenze größer als Effizienzgewinn für Ak-
teur " :
                " / untere Grenze größer als Effizienzgewinn für
Akteur ")
                + akteur.getIndex();
        }

        if(akteur.getVmax() > effizienzGewinn) {
            errocc = true;
            errstr += ((errstr.length() == 0)?
                "Obere Grenze größer als Effizienzgewinn für Ak-
teur " :
                " / obere Grenze größer als Effizienzgewinn für
Akteur ")
                + akteur.getIndex();
        }
    }

    if(errocc)
        throw new IllegalArgumentException(errstr);
}

@Override
public ArrayList<Double> berechneTauwert() {
    Double sumVmin = 0.0d;
    Double sumVmax = 0.0d;
    for(Akteur akteur : akteure) {
        sumVmin += akteur.getVmin();
        sumVmax += akteur.getVmax();
    }

    Double y = (sumVmin != sumVmax)?
        (effizienzGewinn - sumVmin) / (sumVmax - sumVmin) : 0.5d;

    ArrayList<Double> tau = new ArrayList<Double>();
    for(Akteur akteur : akteure)
        tau.add((y * akteur.getVmax() + (1 - y) * akteur.getVmin()));

    return tau;
}
}

/*****
 * Akteur
 *****/

package tauwert.algo;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

```

```
class Akteur {

    public static final ArrayList<Akteur> akteure = new ArrayList<Akteur>();

    int index;
    Double charWert;
    Double vmax;
    Double vmin;
    boolean massgeblich;

    // Wesentliche Aussenseiterkoalitionen
    ArrayList<Koalition> wKoalition = new ArrayList<Koalition>();
    // Unwesentliche Aussenseiterkoalitionen
    ArrayList<Koalition> uKoalition = new ArrayList<Koalition>();

    public Akteur(int index, Double marginal, CharacterFunction charFunc,
        double effizienzGewinn, int numAkteure) {
        this.index = index;
        this.charWert = charFunc.calculate(index);

        this.vmax = this.calcVmax(effizienzGewinn, charFunc, numAkteure);

        akteure.add(this);
    }

    public Double calcVmax(double effizienzGewinn, CharacterFunction
charFunc, int numAkteure) {
        //Stelle eine Koalition mit allen Akteuren ausser dem
        //Aktuellen zusammen.
        ArrayList<Integer> koalition = new ArrayList<Integer>();
        for(int i = 1; i <= numAkteure; i++) {
            if((i) == this.getIndex())
                continue;
            koalition.add(i);
        }

        return effizienzGewinn - charFunc.calculate(koalition);
    }

    public int getIndex() {
        return index;
    }

    public void setIndex(int index) {
        this.index = index;
    }

    public boolean isMassgeblich() {
        return this.massgeblich;
    }

    public double getVmax() {
        return this.vmax;
    }

    public double getVmin() {
        return this.vmin;
    }
}
```

```

public void vorsortiere(CharacterFunction charFunction) {
    Permuter<Akteur> perm = new Permuter<Akteur>(
        akteure.toArray(new Akteur[akteure.size()]), 1,
        akteure.size() - 1);

    Koalition tmpKoalition;
    while (perm.peek() != null) {
        tmpKoalition = new Koalition(perm.next(), charFunction);
        if (!tmpKoalition.inKoalition(this))
            continue;

        System.out.println(tmpKoalition);

        if (tmpKoalition.getCharWert() > 0)
            this.wKoalition.add(tmpKoalition);
        else
            this.uKoalition.add(tmpKoalition);
    }
    this.massgeblich = (wKoalition.size() > 0);
}

public Double calcIndividualDrohwert() {

    for(Koalition k : this.wKoalition) {
        if(k.size() != 1) continue;
        if(k.inKoalition(this))
            return this.charWert;
    }

    return -Double.MAX_VALUE;
}

public void calcVmin() {
    if (!this.massgeblich) {
        this.vmin = 0.0d;
        return;
    }

    //Double iDrohwert = this.charWert;
    Double iDrohwert = this.calcIndividualDrohwert();
    Double kDrohwert = -Double.MAX_VALUE;

    for (Koalition koalition : this.wKoalition) {
        // Wenn die Koalitionsgroesse < 2 so wurde
        // diese schon als individueller drohwert
        // berechnet
        if (koalition.size() < 2 || !koalition.inKoalition(this))
            continue;

        // Ist der neue kollektive Drohwert größer als
        // alle anderen, setze diesen
        kDrohwert = Math.max(kDrohwert,
            koalition.getCharWert());
    }
    this.vmaxSumKoalition(koalition));
}

// Die untere Grenze ergibt sich aus max(0, iDrohwert, kDrohwert)
this.vmin = Math.max(Math.max(0, iDrohwert), kDrohwert);
}

```

```
// Berechne die summe der oberen Grenzen einer Koalition
// unter Auslassung des aktuellen Akteurs
private Double vmaxSumKoalition(Koalition koalition) {
    Double sum = 0.0d;
    for (Akteur akteur : koalition.asList()) {
        if (akteur.getIndex() != this.getIndex())
            sum += akteur.getVmax();
    }
    return sum;
}

@Override
public String toString() {
    return "Akteur " + this.index + " vmin: " + this.vmin + " vmax: "
        + this.vmax;
}

class Koalition {

    ArrayList<Akteur> koalition = new ArrayList<Akteur>();

    Double charWert = 0.0d;

    public Koalition(Akteur[] koalition, CharacterFunction charFunc-
tion) {
        ArrayList<Akteur> al = new ArrayList<Akteur>(
            Arrays.asList(koalition));
        this.initialize(al, charFunction);
    }

    public Koalition(ArrayList<Akteur> koalition,
        CharacterFunction charFunction) {
        this.initialize(koalition, charFunction);
    }

    private void initialize(ArrayList<Akteur> koalition,
        CharacterFunction charFunction) {
        this.koalition = koalition;

        // Berechne wert der char. Funktion für diese Kombination
        ArrayList<Integer> aArr = new ArrayList<Integer>();
        for (int i = 0; i < koalition.size(); i++)
            aArr.add(koalition.get(i).getIndex());

        this.charWert = charFunction.calculate(aArr);
    }

    public boolean inKoalition(Akteur akteur) {
        return (koalition.indexOf(akteur) >= 0);
    }

    public Double getCharWert() {
        return this.charWert;
    }

    public int size() {
        return koalition.size();
    }

    public List<Akteur> asList() {
        return this.koalition;
    }
}
```

```

@Override
public String toString() {
    String out = "[";
    for (Akteur akteur : koalition) {
        out += akteur.getIndex() + ", ";
    }
    return (out.substring(0, out.length() - 1) + "] C(" +
this.charWert + ")");
}
}

```

Die Berechnung des  $\tau$ -Werts findet ausschließlich in den beiden Klassen *AkteureTauwert* sowie *Akteur* statt. In der erstgenannten Klasse werden hauptsächlich Integritätstests sowie akteurübergreifende Berechnungen, in der letztgenannten Klasse akteurspezifische Berechnungen durchgeführt.

Ein *Akteur*-Objekt wird mit der Nummer des Akteurs ( $1 \dots N$ ), der akteurspezifischen Marginalkoalition, der charakteristischen Funktion sowie dem zu verteilenden Effizienzgewinn initialisiert. Dabei wird bereits bei der Erzeugung des Objekts die Obergrenze *OG* des Akteurs berechnet.

In der Methode *vorsortiere* der *Akteur*-Klasse werden durch Permutation einer Liste aller Akteure die jeweiligen wesentlichen und unwesentlichen Außenseiterkoalitionen ermittelt. Anhand dieser Liste wird bestimmt, ob es sich bei dem Akteur um einen maßgeblichen oder einen nicht-maßgeblichen Akteur handelt, was sich wiederum später auf die Berechnung der Untergrenze des  $\tau$ -Werts auswirkt.

In der Methode *calcVmin* wird die untere Grenze des  $\tau$ -Werts für den jeweiligen Akteur ermittelt. Dazu wird zunächst der individuelle Drohwert als der Wert festgelegt, den die charakteristische Funktion für eine Koalition annimmt, die nur aus dem aktuellen Akteur besteht. Darauf folgt die Berechnung des kollektiven Drohwerts. Dazu wird über die in der Methode *vorsortiere* erzeugte Liste von wesentlichen Außenseiterkoalitionen iteriert. Der kollektive Drohwert der jeweiligen Koalition ergibt sich aus dem Wert der charakteristischen Funktion für die Koalition abzüglich der Summe der Obergrenzen aller Akteure der Koalition exklusive der Obergrenze für den jeweiligen Akteur („Anführer“), für den der Drohwert bestimmt wird. Der kollektive Drohwert für den Akteur ergibt sich aus dem maximalen kollektiven Drohwert aller für ihn wesentlichen Außenseiterkoalitionen. Die Un-



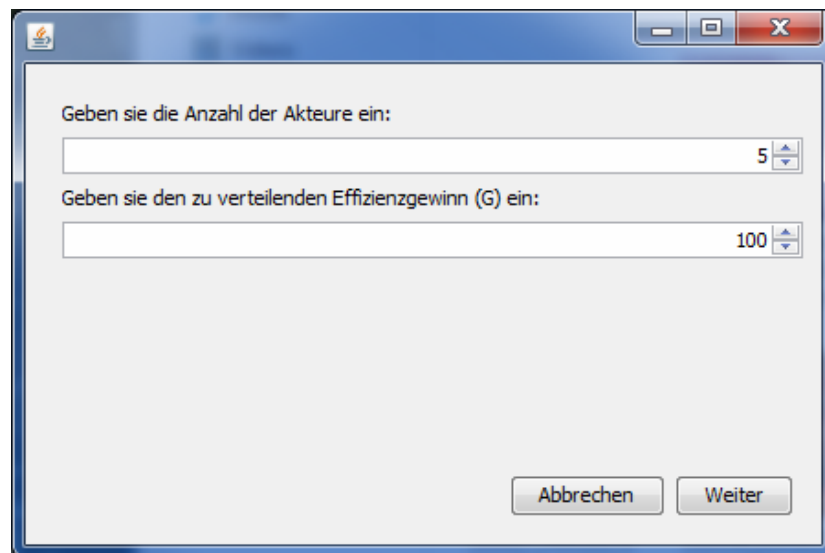
tergrenze wird dann als das Maximum aus Null, dem individuellen und dem kollektiven Drohwert festgesetzt.

In der Klasse *AkteureTauwert* werden zunächst die Parameter einem Integritätstest unterzogen. Dabei wird überprüft, ob eine ausreichende Anzahl von Akteuren sowie ein positiver zu verteiler Effizienzgewinn vorliegt und, ob der Effizienzgewinn mit dem Wert der charakteristischen Funktion für eine Koalition aus allen Akteuren übereinstimmt. Im Fehlerfall wird eine Ausnahmemeldung mit einer Fehlermeldung generiert, die an die nächst höhere Methode weitergereicht wird. Ist der Test erfolgreich verlaufen, werden die *Akteur*-Objekte erzeugt und durch Aufruf der Methoden *vorsortiere* und *calcVmin* initialisiert. Abschließend wird die Integrität der in den *Akteur*-Objekten berechneten Werte überprüft. Dabei wird für jeden Akteur getestet, ob die untere Grenze nicht oberhalb der oberen Grenze liegt und ob die untere Grenze nicht größer sowie die obere Grenze nicht kleiner als der zu verteilende Effizienzgewinn ist. Sollte dieser Test scheitern, wird wiederum eine Fehlermeldung generiert.

In der Methode *berechneTauwert* werden letztendlich die  $\tau$ -Werte der einzelnen Akteure aus deren oberen und unteren Grenzen berechnet und als Liste zurückgegeben.

### 3.3 Anwendung des Programms zur Berechnung des $\tau$ -Werts

Nach Aufruf des Programms zur Berechnung der  $\tau$ -Werts durch einen Doppelklick auf den Programm-Icon erscheint der folgende Dialog.

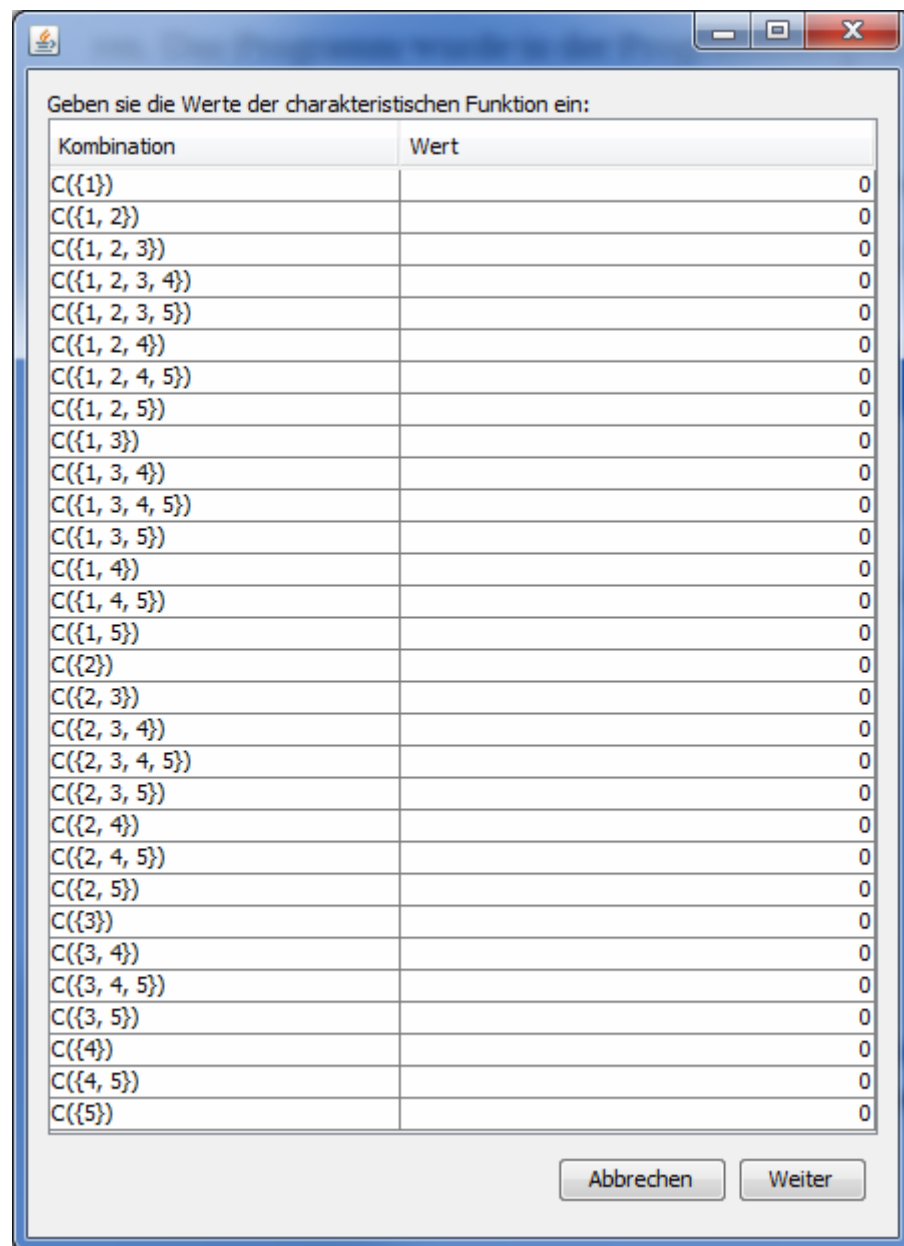


The image shows a standard Windows-style dialog box. The title bar is blue and contains the minimize, maximize, and close buttons. The main area is white and contains two text labels with corresponding input fields. The first label is 'Geben sie die Anzahl der Akteure ein:' followed by a text box containing the number '5'. The second label is 'Geben sie den zu verteilenden Effizienzgewinn (G) ein:' followed by a text box containing the number '100'. At the bottom right of the dialog, there are two buttons: 'Abbrechen' and 'Weiter'.

**Abbildung 1: Startdialog**

Zur Berechnung des  $\tau$ -Werts müssen hier die Anzahl  $N$  der Akteure, unter denen der Effizienzgewinn  $G$  aufgeteilt werden soll, und der zu verteilende Effizienzgewinn angegeben werden. Unter den Eingabefeldern befinden sich zwei Buttons in diesem Startdialog. Das Anklicken des Buttons „Abbrechen“ öffnet einen zusätzlichen Dialog. In diesem Dialog wird der Anwender gefragt, ob er die Berechnung wirklich abbrechen will. Wird diese Frage mit „Ja“ bestätigt, schließt sich das Programm. Wird die Frage mit „Nein“ beantwortet, schließt sich der Dialog und der Startdialog ist erneut im Vordergrund.

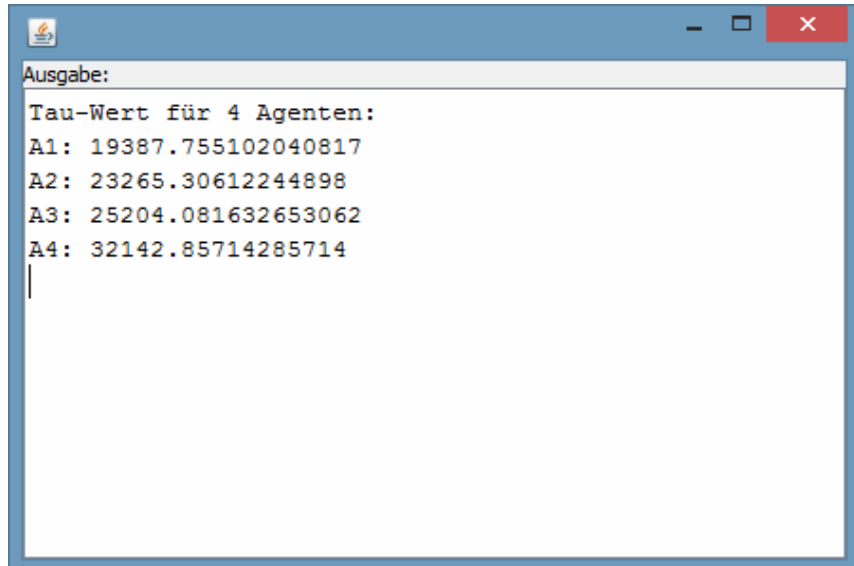
Wählt man im Startdialog statt „Abbrechen“ den Button „Weiter“, wird das folgende Fenster aufgerufen, in dem die Werte der charakteristischen Funktion angegeben werden müssen (siehe Abbildung 2). Die möglichen Kombinationen wurden bereits vom Programm ermittelt und werden in einer Tabelle aufgelistet. Der Funktionswert  $c(\{\emptyset\})$  wird nicht abgefragt, da dieser immer gleich null ist. Auch der Wert für die große Koalition  $C_0$  wird nicht abgefragt, da dieser bereits im ersten Dialog in als zu verteilerender Effizienzgewinn abgefragt wurde. Der Anwender muss nun lediglich die jeweiligen Werte der charakteristischen Funktion ergänzen. Das Programm setzt zunächst alle Werte auf null.



Kombination	Wert
$C(\{1\})$	0
$C(\{1, 2\})$	0
$C(\{1, 2, 3\})$	0
$C(\{1, 2, 3, 4\})$	0
$C(\{1, 2, 3, 5\})$	0
$C(\{1, 2, 4\})$	0
$C(\{1, 2, 4, 5\})$	0
$C(\{1, 2, 5\})$	0
$C(\{1, 3\})$	0
$C(\{1, 3, 4\})$	0
$C(\{1, 3, 4, 5\})$	0
$C(\{1, 3, 5\})$	0
$C(\{1, 4\})$	0
$C(\{1, 4, 5\})$	0
$C(\{1, 5\})$	0
$C(\{2\})$	0
$C(\{2, 3\})$	0
$C(\{2, 3, 4\})$	0
$C(\{2, 3, 4, 5\})$	0
$C(\{2, 3, 5\})$	0
$C(\{2, 4\})$	0
$C(\{2, 4, 5\})$	0
$C(\{2, 5\})$	0
$C(\{3\})$	0
$C(\{3, 4\})$	0
$C(\{3, 4, 5\})$	0
$C(\{3, 5\})$	0
$C(\{4\})$	0
$C(\{4, 5\})$	0
$C(\{5\})$	0

Abbildung 2: Eingabe der Werte der charakteristischen Funktion

Wurden die Werte der charakteristischen Funktion eingegeben und die Eingabe mit Klicken des Buttons „Weiter“ bestätigt, erscheint im nächsten Fenster als Ausgabe des Programms das Ergebnis der  $\tau$ -Wert Berechnung. Es wird für jeden einzelnen Akteur der  $\tau$ -Wert ausgegeben.



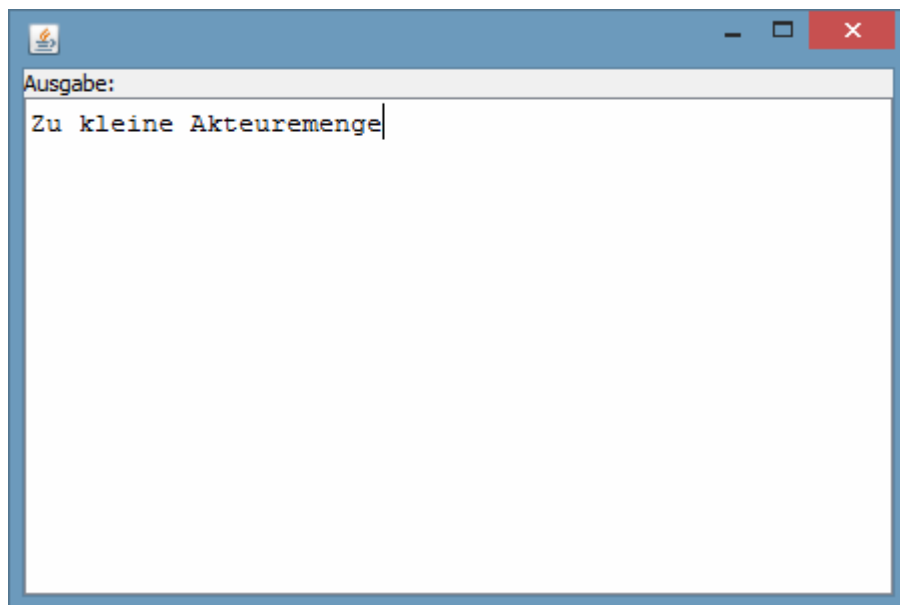
```
Ausgabe:  
Tau-Wert für 4 Agenten:  
A1: 19387.755102040817  
A2: 23265.30612244898  
A3: 25204.081632653062  
A4: 32142.85714285714  
|
```

**Abbildung 3: Ausgabe des Ergebnisses**

## 4 Test des Programms zur Berechnung des $\tau$ -Werts

Im Folgenden werden Testfälle vorgestellt, die Ergebnisse, die das Programm zur Berechnung des  $\tau$ -Werts ausgibt, auf Korrektheit überprüft. Neben Testfällen, die eine korrekte Berechnung des  $\tau$ -Werts nachweisen, werden zusätzlich Testfälle konstruiert, die Fehlermeldungen hervorrufen sollen, um zu testen, ob das Programm die Fehler erkennt und den Anwender entsprechend darauf hinweist.

Der erste Testfall besteht daraus, dass in den Startdialog als Anzahl der Akteure 0 eingegeben wird und als zu verteiler Effizienzgewinn 100. Die Eingabe von 0 für die Anzahl der Akteure wird nicht zugelassen. Die Anzahl wird automatisch auf 1 korrigiert. Wird durch Klicken auf den Button „Weiter“ diese Eingabe bestätigt, folgt ein leerer Dialog. Bei erneutem Klick auf „Weiter“ erscheint der Hinweis, dass die Akteurmenge, sprich die Anzahl der Akteure, zu klein ist.



**Abbildung 4: Ergebnis des Testfalls 1**

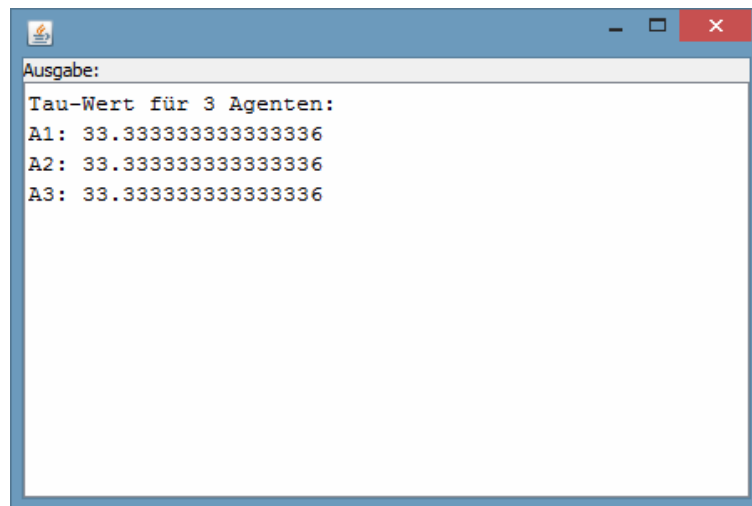
Das Ergebnis dieses ersten Tests ist positiv. Das Programm erwartet eine Eingabe von mindestens 2 Akteuren, denn ist nur ein Akteur am Spiel beteiligt, liegt kein Verteilungsproblem vor.

Im zweiten Testfall werden als Anzahl der Akteure 3 angegeben und als zu verteiler Effizienzgewinn 100. Die Eingaben werden mit Klicken auf den Button „Weiter“ bestätigt. Des Weiteren werden im nächsten Schritt die folgenden Werte für die charakteristische Funktion eingegeben:

$C_m$	$c(C_m)$
$C_1=\{A_1\}$	0
$C_2=\{A_2\}$	0
$C_3=\{A_3\}$	0
$C_4=\{A_1,A_2\}$	20
$C_5=\{A_1,A_3\}$	20
$C_6=\{A_2,A_3\}$	20

**Tabelle 1: Daten des Testfalls 2**

Durch das Betätigen des Button „Weiter“ wird die Berechnung durchgeführt. Das Ergebnis des zweiten Testfalls, das das Programm ausgibt, zeigt die folgende Abbildung:



**Abbildung 5: Ergebnis des Testfalls 2**

Das Ergebnis des zweiten Tests ist positiv. Jeder Akteur erhält laut Ausgabe des Programms einen Anteil von 33,33 am Effizienzgewinn. Das ausgegebene Ergebnis der Berechnung ist korrekt. Dies kann leicht nachvollzogen werden. Da für alle drei Akteure die gleichen Ausgangswerte eingegeben wurden, haben alle eine identische Verhandlungsposition. Es resultiert eine Gleichverteilung bei der jeder Akteur einen gleichgroßen Anteil am Effizienzgewinn erhält.

Durch Berechnung der oberen Grenze  $OG$  und der unteren Grenze  $UG$  durch Einsetzen der Werte aus Tabelle 1 in die Formeln 2.4 und 2.5 sowie das Einsetzen der Oberen Grenze  $OG$  und der unteren Grenze  $UG$  in Formel 2.10 kann dieses Ergebnis bestätigt werden:

$A_n$	$v_{n,max}$	$v_{n,min}$
$A_1$	80	0
$A_2$	80	0
$A_3$	80	0

**Tabelle 2:  $v_{n,max}$  und  $v_{n,min}$  des Testfalls 2**

Die Ergebnisse aus Tabelle 2 werden verwendet um den Gewichtungsfaktor zu berechnen. Da hier der Normalfall vorliegt kann die folgende Formel verwendet werden:

$$\gamma = \frac{G - \sum_{n=1}^N v_{n,min}}{\sum_{n=1}^N v_{n,max} - \sum_{n=1}^N v_{n,min}}$$

Durch Einsetzen ergibt sich der Gewichtungsfaktor wie folgt:

$$\gamma = \frac{100 - 0}{240 - 0}$$

$$\Rightarrow \gamma = 0,416666$$

Wird dieser Gewichtungsfaktor in Formel 2.10 eingesetzt ergeben sich die folgenden Werte als Zwischenergebnisse bei der Berechnung des  $\tau$ -Werts:

$A_n$	$\gamma \cdot v_{n,max}$	$(1-\gamma) \cdot v_{n,min}$	$v_{n,\tau}$
$A_1$	33,3333	0	33,3333
$A_2$	33,3333	0	33,3333
$A_3$	33,3333	0	33,3333

**Tabelle 3: Berechnung des  $\tau$ -Werts für Testfall 2**

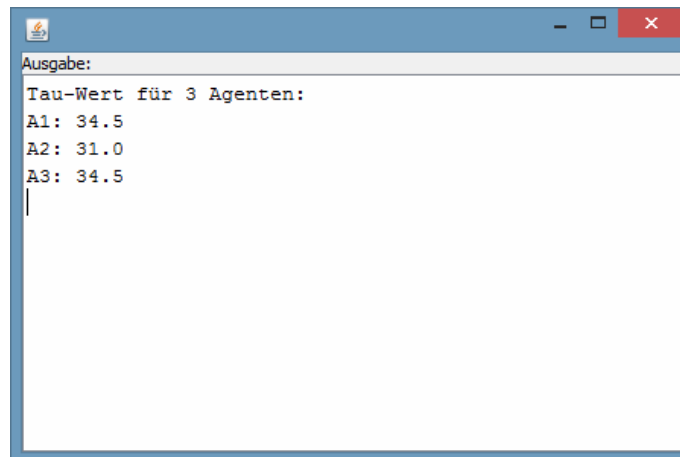
Die Summe der zweiten und der dritten Spalte ergibt den Anteil des jeweiligen Akteurs am Effizienzgewinn, der in der vierten Spalte dargestellt wird. Hier wird ein erster Schwachpunkt des Programms deutlich. Denn erhält jeder der drei Akteure einen Anteil von 33,33 am Effizienzgewinn, ergibt das insgesamt 99,99. Dies ist weniger als der Effizienzgewinn 100. Somit handelt es sich nicht um einer effiziente Verteilung.

Im nächsten Testfall werden wiederum als Anzahl der Akteure 3 und als zu verteiler Effizienzgewinn 100 angegeben. Die Eingaben werden mit Klicken auf den Button „Weiter“ bestätigt. Des Weiteren werden im nächsten Schritt die folgenden Werte als Werte für die charakteristische Funktion eingegeben:

$C_m$	$c(C_m)$
$C_1=\{A_1\}$	10
$C_2=\{A_2\}$	10
$C_3=\{A_3\}$	10
$C_4=\{A_1,A_2\}$	20
$C_5=\{A_1,A_3\}$	30
$C_6=\{A_2,A_3\}$	20

**Tabelle 4: Daten des Testfalls 3**

Das Ergebnis zeigt die folgende Abbildung:



**Abbildung 6: Ergebnis des Testfalls 3**

Das Ergebnis des zweiten Tests ist positiv. Das ausgegebene Ergebnis der Berechnung ist korrekt. Wie schon beim Testfall 2 kann die Berechnung den folgenden Tabellen entnommen werden:

$A_n$	$v_{n.max}$	$v_{n.min}$
$A_1$	80	0
$A_2$	70	0
$A_3$	80	0

**Tabelle 5:  $v_{n.max}$  und  $v_{n.min}$  des Testfalls 3**



Die Ergebnisse aus Tabelle 2 werden verwendet um den Gewichtungsfaktor zu berechnen. Da hier der Normalfall vorliegt kann die folgende Formel verwendet werden:

$$\gamma = \frac{G - \sum_{n=1}^N v_{n.min}}{\sum_{n=1}^N v_{n.max} - \sum_{n=1}^N v_{n.min}}$$

Durch Einsetzen ergibt sich der Gewichtungsfaktor wie folgt:

$$\gamma = \frac{100 - 30}{230 - 30}$$

$$\Rightarrow \gamma = 0,35$$

Wird dieser Gewichtungsfaktor in Formel 2.10 eingesetzt ergeben sich die folgenden Werte als Zwischenergebnisse bei der Berechnung des  $\tau$ -Werts:

$A_n$	$\gamma \cdot v_{n.max}$	$(1-\gamma) \cdot v_{n.min}$	$v_{n,\tau}$
$A_1$	28,0	6,5	34,5
$A_2$	24,5	6,5	31,0
$A_3$	28,0	6,5	34,5

**Tabelle 6: Berechnung des  $\tau$ -Werts für Testfall 3**

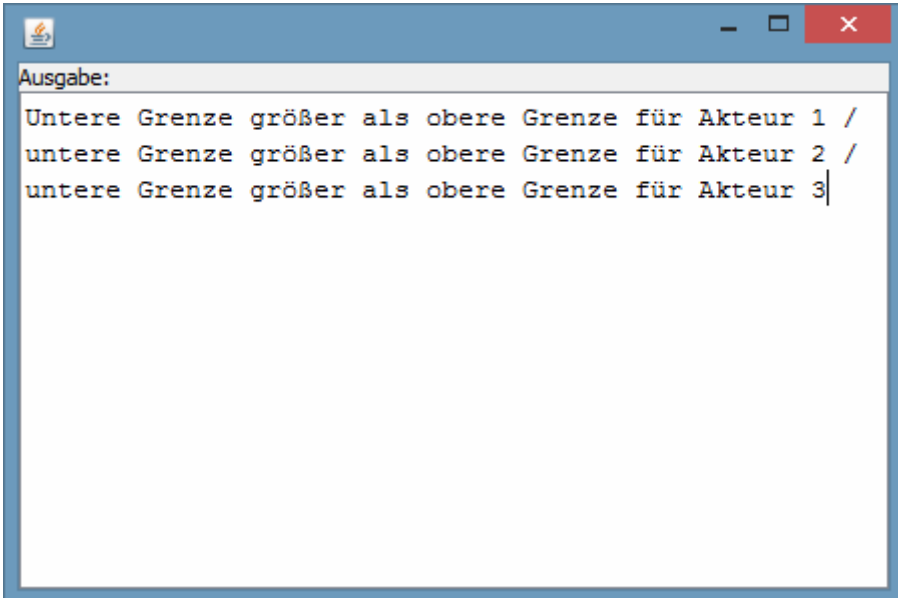
Die Summe der zweiten und der dritten Spalte ergibt den Anteil des jeweiligen Akteurs am Effizienzgewinn, der in der vierten Spalte dargestellt wird. Die Werte der rechten Spalte stimmen mit der Ausgabe des Programms überein.

Im vierten und letzten Testfall werden wiederum als Anzahl der Akteure 3 und abweichend von den vorangegangenen Testfällen als zu verteilender Effizienzgewinn 150 angegeben. Die Eingaben werden mit Klicken auf den Button „Weiter“ bestätigt. Des Weiteren werden im nächsten Schritt die folgenden Werte für die charakteristische Funktion eingegeben:

$C_m$	$c(C_m)$
$C_1=\{A_1\}$	0
$C_2=\{A_2\}$	0
$C_3=\{A_3\}$	0
$C_4=\{A_1,A_2\}$	100
$C_5=\{A_1,A_3\}$	120
$C_6=\{A_2,A_3\}$	100

**Tabelle 7: Daten Testfall 4**

Das Ergebnis zeigt die folgende Abbildung:



```

Ausgabe:
Untere Grenze größer als obere Grenze für Akteur 1 /
untere Grenze größer als obere Grenze für Akteur 2 /
untere Grenze größer als obere Grenze für Akteur 3

```

**Abbildung 7: Ergebnis des Testfalls 4**

Das Programm gibt eine Fehlermeldung aus, da es sich nicht um ein quasi-balanciertes Spiel handelt. Die Untere Grenze  $UG$  ist größer als die obere Grenze  $OG$ . Dass diese Ausgabe korrekt ist, zeigt die folgende Tabelle:

$A_n$	$v_{n.max}$	$v_{n.min}$
$A_1$	50	70
$A_2$	30	50
$A_3$	50	70

**Tabelle 8:  $v_{n.max}$  und  $v_{n.min}$  des Testfalls 3**

Alle Testfälle verliefen positiv, d.h., das Programm lieferte das korrekte Ergebnis oder in den Testfällen mit fehlerhafter Eingabe einen Hinweis auf den Fehler.

## 5 Fazit

Wie bereits in Kapitel 1.1 erläutert, war es Ziel dieses Arbeitsberichtes, ein Programm zur Berechnung des  $\tau$ -Werts zu erläutern, das den Einsatz des spieltheoretischen Lösungskonzepts in der Praxis vereinfacht, um die Wahrscheinlichkeit zu erhöhen, dass der  $\tau$ -Wert in der betrieblichen Praxis verwendet wird. Dieses Programm wurde in Kapitel 3.2 vorgestellt und in Kapitel 3.3 getestet. Das Ergebnis zeigt, dass das Programm die Berechnung des  $\tau$ -Werts für die vorgestellten Fälle korrekt durchführt und bei unzulässigen Eingaben, den Anwender darauf hinweist.<sup>58</sup>

Dennoch gibt es durchaus Weiterentwicklungspotential, denn es handelt sich bei dem vorgestellten Programm zur Berechnung des  $\tau$ -Werts lediglich um einen Prototyp, bei dessen Entwicklung die Umsetzung des Berechnungsalgorithmus im Vordergrund stand. Die graphische Oberfläche des vorgestellten Programms wurde bisher möglichst einfach gehalten. In einer Weiterentwicklung des Prototyps könnte z.B. die graphische Oberfläche weiterentwickelt werden. Auch die Eingabemöglichkeiten können in diesem Zusammenhang benutzerfreundlicher gestaltet werden. Zusätzlich könnte auch der dem  $\tau$ -Wert sehr ähnliche  $\chi$ -Wert<sup>59</sup> durch das Programm berechnet werden, sodass ein Vergleich dieser beiden spieltheoretischen Konzepte möglich ist.

---

<sup>58</sup> Es wurde lediglich die korrekte Funktionsweise für die zuvor erläuterten vier Testfälle nachgewiesen. Diese korrekte Funktionsweise kann jedoch aufgrund der geringen Anzahl der Testfälle nicht verallgemeinert werden. Dazu müsste eine größere Anzahl an Test durchgeführt werden.

<sup>59</sup> Vgl. BERGATIÑOS/MASSO (2002), S. 272 ff., BERGATIÑOS/MASSO (2000), S. 4 ff., BERGATIÑOS/MASSO (1994), S. 5 ff., und BERGATIÑOS/MASSO (1996), S. 279 ff.

## Literatur

### **AUMANN/MASCHLER (1995)**

Aumann, R. J.; Maschler, M. B.: Repeated Games with Incomplete Information. The MIT Press 1995.

### **BERGANTIÑOS/MASSÓ (1994)**

Bergantiños, G.; Massó, J.: Notes on a New Compromise Value: the  $\chi$ -Value. Working Paper W.P. 266.94, Department d'Economia i d'Història Econòmica and Institut d'Anàlisi Econòmica, Universitat Autònoma de Barcelona. Bellaterra (Barcelona) 1994.

### **BERGANTIÑOS/MASSÓ (1996)**

Bergantiños, G.; Massó, J.: Notes on a New Compromise Value: the  $\chi$ -Value. In: Optimization, Vol. 38 (1996), S. 277-286.

### **BERGANTIÑOS/MASSÓ (2002)**

Bergantiños, G.; Massó, J.: The Chi-compromise value for non-transferable utility games. Mathematical Methods of Operations Research, 56 (2002), S. 269-286.

### **BERGANTIÑOS/SÁNCHEZ (2002)**

Bergantiños, G.; Sánchez, J.: How to Distribute Costs Associated with a Delayed Project. Annals of Operations Research, 109 (2002), S. 159-174.

### **BILBAO/LEBRÓN/JIMÉNEZ-LOSADA/TIJS (2001)**

Bilbao, J. M.; Lebrón, E.; Jiménez-Losada, A.; Tijs, S.: A dual Approach to Compromise Values. In: Petrosjan, L. A.; Mazalov, V. V. (Hrsg.): Game Theory and Applications, Volume VII. Novapublishers 2001, S. 23-32.

### **BORGWARDT (2001)**

Borgwardt, K. H.: Optimierung, Operations Research, Spieltheorie – Mathematische Grundlagen. Birkhäuser 2001.

### **BRÂNZEI/DIMITROV/TIJS (2005a)**

Brânzei, R.; Dimitrov, D.; Tijs, S.: Convex Games, Clan Games and their Marginal Games. Working Paper No. 368, Institute of Mathematical Economics, Bielefeld University 2005.

### **BRÂNZEI/DIMITROV/TIJS (2005b)**

Brânzei, R.; Dimitrov, D.; Tijs, S.: Models in Cooperative Game Theory – Crisp, Fuzzy, and Multi-Choice Games. Springer Verlag 2005.

**DERKS/KUIPERS (2004)**

Derks, J.; Kuipers, J.: On the Number of Extreme Points of the Core of Transferable Utility Game. In: Borm, P.; Peters, H. (Hrsg.) Chapters in Game Theory – In honor of Stef Tijs. Kluwer Academic Publishers 2004, S. 83-98.

**DRECHSEL (2010)**

Drechsel, J.: Cooperative Lot Sizing Games in Supply Chains. Dissertation Universität Duisburg-Essen 2009. Springer Verlag 2010.

**DRIESSEN (1985)**

Driessen, T. S. H.: Contributions to the theory of cooperative games: the  $\tau$ -value and  $k$ -convex games. Kluwer Academic Publishers 1985.

**DRIESSEN/TIJS (1982)**

Driessen, T. S. H.; Tijs, S. H.: The  $\tau$ -Value, the Nucleolus and the Core for a Subclass of Games. In: Loeffel, H.; Stähly, P. (Hrsg.): Methods of Operations Research 46. VII. Symposium on Operations Research, 19.-21.08.1982 in St. Gallen, Tagungsbericht – Proceedings der Sektionen 4-9. Springer Verlag 1982, S. 395-406.

**DRIESSEN/TIJS (1984)**

Driessen, T. S. H.; Tijs, S. H.: Extensions and Modifications of the  $\tau$ -Value for Cooperative Games. In: Hammer, G.; Pallaschke, D. (Hrsg.): Selected Topics in Operations Research and Mathematical Economics. Springer Verlag 1984, S. 252-261.

**DRIESSEN/TIJS (1985)**

Driessen, T. S. H.; Tijs, S. H.: The  $\tau$ -Value, The Core and Semiconvex Games. In: International Journal of Game Theory, Vol. 14 (1985), Issue 4, S. 229-247.

**DRIESSEN/TIJS (1990)**

Driessen, T. S. H.; Tijs, S. H.: The Core and the  $\tau$ -Value for Cooperative Games with Coalition Structures. Memorandum No. 848, Faculty of Applied Mathematics, University of Twente. Enschede 1990.

**EISELE (1995)**

Eisele, J.: Erfolgsfaktoren des Joint-Venture-Managements. Dissertation Universität Mannheim 1995. Gabler Verlag 1995.

**FROMEN (2004)**

Fromen, B.: Faire Aufteilung in Unternehmensnetzwerken – Lösungsvorschläge auf der Basis der kooperativen Spieltheorie. Dissertation Universität Duisburg-Essen 2004. Deutscher Universitäts-Verlag 2004.

**GILLIES (2010)**

Gillies, R.: The Cooperative Game Theory of Networks and Hierarchies. Springer Verlag 2010.

**GÜTH (1992)**

Güth, W.: Spieltheorie und ökonomische (Bei)Spiele. Springer Verlag 1992.

**HEROWITSCH (1997)**

Herowitsch, J.: Organisatorische, personelle und kulturelle Erfolgsfaktoren von Unternehmenskooperationen. Dissertation Technische Universität Wien, Österreich 1997.

**HOLLER/ILLING (2009)**

Holler, M. J., Illing, G.: Einführung in die Spieltheorie. 7. Auflage, Springer Verlag 2009.

**JENE/ZELEWSKI (2011a)**

Jene, S.; Zelewski, S.: Fair Distribution of Added Values in Networks of Autonomous Actors. In: Jodlbauer, H.; Olhager, J.; Schonberger, R.J. (Hrsg.): Modelling Value. 1. International Conference on Value Change Management, 04.-05.03.2011 in Steyr, Pre-Proceedings, Volume 2. Shaker Verlag 2011, S. 317-337.

**JENE/ZELEWSKI (2011b)**

Jene, S.; Zelewski, S.: Fair Distribution of Profit in Supply Chains. In: Kersten, W.; Blecker, T.; Jahn, C. (Hrsg.): International Supply Chain Management and Collaboration Practices. Eul Verlag 2011, S. 299-313.

**JENE/ZELEWSKI (2011c)**

Jene, S., Zelewski, S.: Distributive Justice in Supply Chains – Fair Distribution of Collectively Earned Profits in Supply Chains. In: Sucky, E.; Asdecker, B.; Dobhan, A.; Haas, S.; Wiese, J. (Hrsg.): Logistikmanagement – Herausforderungen, Chancen und Lösungen. Band III.. Bamberg 2011, S. 115-132.

**JENE/ZELEWSKI (2012)**

Jene, S., Zelewski, S.: Fair Distribution of Added Values in Networks of Autonomous Actors. In: Jodlbauer, H.; Olhanger, R.; Schonberger, R.J. (Hrsg.): Modelling Value – Selected Papers of the 1st International Conference on Value Chain Management. Physica Verlag 2012, S. 167-186.

**KLENDAUER ET AL. (2006)**

Klendauer, R.; Streicher, B.; Jonas, E.; Frey, D.: Fairness und Gerechtigkeit. In: Bierhoff, H.-W.; Frey, D. (Hrsg.): Handbuch der Sozialpsychologie und Kommunikationspsychologie. Hogrefe Verlag 2006, S. 187-195.

**KOLLOGE (2010)**

Kolloge, K.: Erfolgsfaktoren internationaler Unternehmenskooperationen – Eine empirische Untersuchung am Beispiel von Vertriebskooperationen im deutschen Maschinenbau. Dissertation Universität Münster 2010. Shaker Verlag 2010.

**KOLOKOLTSOV/MALAFEYEV (2010)**

Kolokoltsov, V. N.; Malafeyev, o. A.: Understanding Game Theory – Introduction to the Analysis of Many Agent Systems with Competition and Cooperation. World Scientific 2010.

**LEYTON-BROWN/SHOHAM (2008)**

Leyton-Brown, K.; Shoham, Y.: Essentials of Game Theory – A Concise, Multidisciplinary Introduction. Morgan & Claypool Publishers 2008.

**LIND/TYLER (1988)**

Lind, E. A.; Tyler, T. R.: The Social Psychology of Procedural Justice. Plenum Press 1988.

**MCCAIN (2009)**

McCain, R.: Game Theory and Public Policy. Edward Elgar Publishing 2009.

**MCCAIN (2010)**

McCain, R.: Game Theory: Nontechnical Introduction to the Analysis of Strategy. 2. Auflage, World Scientific Publishing 2010.

**MORETTI/PATRONE (2008)**

Moretti, S.; Patrone, F.: Transversality of the Shapley Value. In: Top – An Official Journal of the Spanish Society of Statistics and Operations Research, Vol. 16 (2008), No. 1, S. 1-41.

**MOULIN (1988)**

Moulin, H.: Axioms of Cooperative Decision Making. Cambridge University Press 1988.

**MYERSON (2001)**

Myerson, R. B.: Game theory – Analysis of Conflict. 4. Auflage, Harvard University Press 2001.

**NÚÑEZ/RAFELS (2002)**

Núñez, M.; Rafels, C.: The assignment game – the  $\tau$ -value. In: International Journal of Game Theory, Vol. 31 (2002), No. 3, S. 411-422.

**OSTMANN (2002)**

Ostmann, A.: Cooperation in Environmental Commons. In: Bolle F.; Lehmann-Waffenschmidt, M. (Hrsg.): Surveys in Experimental Economics – Bargaining, Cooperation and Election Stock Markets. Physica Verlag 2002, S. 139-158.

**OWEN (1995)**

Owen, G.: Game Theory. 3. Auflage, Academic Press 1995.

**PELEG/SUDHÖLTER (2007)**

Peleg, B.; Sudhölter, P.: Introduction to the Theory of Cooperative Games. 2. Auflage, Springer Verlag 2007.

**PETERS (2008)**

Peters, H.: Game Theory – A Multi-Leveled Approach. Springer Verlag 2008.

**PLATA-PÉREZ/SÁNCHEZ-PÉREZ (2011)**

Plata-Pérez, L.; Sánchez-Pérez, J.: Convexity and marginal contributions in bankruptcy games. In: EconoQuantum Vol. 8 (2011), No. 1, S. 61-72.

**RASMUSEN (2007)**

Rasmusen, E.: Games and Information – An Introduction to Game Theory. 4. Auflage, Blackwell Publishing 2007.

**ROSENMÜLLER (1977)**

Rosenmüller, J.: Extreme Games and Their Solutions. Springer Verlag 1977.

**ROSENMÜLLER (1981)**

Rosenmüller, J.: The Theory of Games and Markets. North Holland Publishing Company 1981.



**ROSENMÜLLER (2000)**

Rosenmüller, J.: Games Theory – Stochastics, Information, Strategies and Cooperation. Kluwer Academic Publishers 2000.

**ROTH (2006)**

Roth, S.: Preismangement für Leistungsbündel – Preisbildung, Bündelung und Delegation. Habilitationsschrift Universität Bayreuth 2005. Deutscher Universitäts-Verlag 2006.

**SIEG (2005)**

Sieg, G.: Spieltheorie. 2. Auflage, Oldenbourg Verlag 2005.

**SLIKKER/VAN DEN NOUWELAND (2001)**

Slikker, M.; Van Den Nouweland, A.: Social and Economic Networks in Cooperative Game Theory. Kluwer Academic Publishers 2001.

**STUART (2001)**

Stuart, H. W.: Cooperative games and business strategy. In: Chatterjee, K.; Samuelson W. F. (Hrsg.): Game Theory and Business Applications. Kluwer 2001, S. 189-211.

**SZÉP/FORGÓ (1983)**

Szép, J.; Forgó, F.: Einführung in die Spieltheorie. Verlag Harri Deutsch 1983.

**TIJS (1981)**

Tijs, S. H.: Bounds for the Core and the  $\tau$ -Value. In: Moeschlin, O.; Pallaschke, D. (Hrsg.): Game Theory and Mathematical Economics. Proceedings of the Seminar on Game Theory and Mathematical Economics, 07.-10.10.1980 in Bonn/Hagen. North-Holland Publishing Company 1981, S. 123-132.

**TIJS (1987)**

Tijs, S. H.: An Axiomatization of the  $\tau$ -Value. In: Mathematical Social Sciences, Vol. 13 (1987), No. 2, S. 177-181.

**TIJS/DRIESSEN (1983)**

Tijs, S. H.; Driessen, T. S. H.: The  $\tau$ -Value as a Feasible Compromise Between Utopia and Disagreement. Report 8312, Department of Mathematics, University of Nijmegen 1983. Auch erschienen in: Mathematical Social Sciences, Vol. 12 (1986), No. 1, S. 9-20.

**TIJS/OTTEN (1993)**

Tijs, S. H.; Otten, G.-J.: Compromise Values in Cooperative Game Theory. In: Top– An Official Journal of the Spanish Society of Statistics and Operations Research, Vol. 1 (1993), No. 1, S. 1-51.

**WIESE (2005)**

Wiese, H.: Kooperative Spieltheorie. Oldenbourg Verlag 2005.

**WINTER (2002)**

Winter, E.: The Shapley Value. In: Aumann, R.; Hart, S. (Hrsg.): Handbook of Game Theory . With Economic Applications. Volume 3. North-Holland 2002, S. 2025-2054.

**ZARAGARE (1991)**

Zaragare, F. C.: Game Theory – Concepts and Applications. Sage Publications 1991.

**ZELEWSKI (2009)**

Zelewski, S.: Faire Verteilung von Effizienzgewinnen in Supply Webs – ein spieltheoretischer Ansatz auf Basis des  $\tau$ -Werts. Logos Verlag 2009.

**ZELEWSKI/PETERS (2008)**

Zelewski, S.; Peters, M. L.: Modeling of Fairness – Distribution of Efficiency Gains in Supply Webs from a Game-theoretic Point of View. In: o.V.: Proceedings of the Fifteenth International Working Seminar on Production Economics. 03.-07.03.2008 in Innsbruck. Pre-Prints, Vol. 3. Innsbruck 2008, S. 529-540.

**ZELEWSKI/PETERS (2010)**

Zelewski, S.; Peters, M. L.: Fair Distribution of Efficiency Gains in Supply Networks from a Cooperative Game Theory Point of View. In: International Journal of Information Systems and Supply Chain Management, Vol. 3 (2010), No. 2, S. 1-24.

**Institut für Produktion und  
Industrielles Informationsmanagement  
Universität Duisburg-Essen / Campus Essen**

---

**Verzeichnis der Arbeitsberichte  
(ISSN 1614-0842)**

- Nr. 1: Zelewski, S.: Stickels theoretische Begründung des Produktivitätsparadoxons der Informationstechnik. Universität Essen, Essen 1999.
- Nr. 2: Zelewski, S.: Flexibilitätsorientierte Koordinierung von Produktionsprozessen. Universität Essen, Essen 1999.
- Nr. 3: Zelewski, S.: Ontologien zur Strukturierung von Domänenwissen. Universität Essen, Essen 1999.
- Nr. 4: Siedentopf, J.; Schütte, R.; Zelewski, S.: Wirtschaftsinformatik und Wissenschaftstheorie. Universität Essen, Essen 1999.
- Nr. 5: Fischer, K.; Zelewski, S.: Ontologiebasierte Koordination von Anpassungsplanungen in Produktions- und Logistiknetzwerken mit Multi-Agenten-Systemen. Universität Essen, Essen 1999.
- Nr. 6: Weihermann, A. E.; Wöhlert, K.: Gentechnikakzeptanz und Kommunikationsmaßnahmen in der Lebensmittelindustrie. Universität Essen, Essen 1999.
- Nr. 7: Schütte, R.: Zum Realitätsbezug von Informationsmodellen. Universität Essen, Essen 2000.
- Nr. 8: Zelewski, S.: Erweiterungen eines Losgrößenmodells für betriebliche Entsorgungsprobleme. Universität Essen, Essen 2000.
- Nr. 9: Schütte, R.: Wissen, Zeichen, Information, Daten. Universität Essen, Essen 2000.
- Nr. 10: Hemmert, M.: The Impact of Internationalization and Externalization on the Technology Acquisition Performance of High-Tech Firms. Universität Essen, Essen 2001.
- Nr. 11: Hemmert, M.: Erfolgswirkungen der internationalen Organisation von Technologiegewinnungsaktivitäten. Universität Essen, Essen 2001.
- Nr. 12: Hemmert, M.: Erfolgsfaktoren der Technologiegewinnung von F&E-intensiven Großunternehmen. Universität Essen, Essen 2001.

- Nr. 13: Schütte, R.; Zelewski, S.: Epistemological Problems in Working with Ontologies. Universität Essen, Essen 2001.
- Nr. 14: Peters, M. L.; Zelewski, S.: Analytical Hierarchy Process (AHP) – dargestellt am Beispiel der Auswahl von Projektmanagement-Software zum Multiprojektmanagement. Universität Essen, Essen 2002.
- Nr. 15: Zelewski, S.: Wissensmanagement mit Ontologien. Universität Essen, Essen 2002.
- Nr. 16: Klumpp, M.; Krol, B.; Zug, S.: Management von Kompetenzprofilen im Gesundheitswesen. Universität Essen, Essen 2002.
- Nr. 17: Zelewski, S.: Der „non statement view“ – eine Herausforderung für die (Re-) Konstruktion wirtschaftswissenschaftlicher Theorien –. Universität Essen, Essen 2002.
- Nr. 18: Peters, M. L.; Zelewski, S.: A heuristic algorithm to improve the consistency of judgments in the Analytical Hierarchy Process (AHP). Universität Duisburg-Essen (Campus Essen), Essen 2003.
- Nr. 19: Peters, M. L.; Zelewski, S.: Fallstudie zur Lösung eines Standortplanungsproblems mit Hilfe des Analytical Hierarchy Process (AHP). Universität Duisburg-Essen (Campus Essen), Essen 2003.
- Nr. 20: Zelewski, S.: Konventionelle versus strukturalistische Produktionstheorie. Universität Duisburg-Essen (Campus Essen), Essen 2003.
- Nr. 21: Alparslan, A.; Zelewski, S.: Moral Hazard in JIT Production Settings. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 22: Dittmann, L.: Ontology-based Skills Management. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 23: Peters, M. L.; Zelewski, S.: Ein Modell zur Auswahl von Produktionsaufträgen unter Berücksichtigung von Synergien. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 24: Peters, M. L.; Zelewski, S.: Ein Modell zur Zuordnung ähnlicher Kundenbetreuer zu Kunden. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 25: Zelewski, S.: Kooperatives Wissensmanagement in Engineering-Netzwerken – (Vorläufiger) Abschlussbericht zum Verbundprojekt KOWIEN. Zugleich KOWIEN-Projektbericht 10/2004. Universität Duisburg-Essen (Campus Essen), Essen 2004.
- Nr. 26: Siemens, F.: Vorgehensmodell zur Auswahl einer Variante der Data Envelopment Analysis. Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 27: Alan, Y.: Integrative Modellierung kooperativer Informationssysteme – Ein Konzept auf der Basis von Ontologien und Petri-Netzen. Dissertation, Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 28: Akca, N.; Ilas, A.: Produktionsstrategien – Überblick und Systematisierung. Universität Duisburg-Essen (Campus Essen), Essen 2005.

- Nr. 29: Zelewski, S.: Relativer Fortschritt von Theorien – ein strukturalistisches Rahmenkonzept zur Beurteilung der Fortschrittlichkeit wirtschaftswissenschaftlicher Theorien (Langfassung). Universität Duisburg-Essen (Campus Essen), Essen 2005.
- Nr. 30: Peters, M. L.; Schütte, R.; Zelewski, S.: Erweiterte Wirtschaftlichkeitsanalyse mithilfe des Analytic Hierarchy Process (AHP) unter Berücksichtigung des Wissensmanagements zur Beurteilung von Filialen eines Handelsunternehmens. Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 31: Zelewski, S.: Beurteilung betriebswirtschaftlichen Fortschritts – ein metatheoretischer Ansatz auf Basis des „non statement view“ (Langfassung). Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 32: Kijewski, F.; Moog, M.; Niehammer, M.; Schmidt, H. ; Schröder, K.: Gestaltung eines Vorgehensmodells für die Durchführung eines Promotionsprojekts am Fachbereich Wirtschaftswissenschaften der Universität Duisburg-Essen, Campus Essen, zum Erwerb des „Dr. rer. pol.“ mithilfe von PETRI-Netzen. Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 33: Peters, M. L.; Zelewski, S.: Effizienz-Analyse unter Berücksichtigung von Satisfizierungsgrenzen für Outputs – Die Effizienz-Analysetechnik EATWOS. Universität Duisburg-Essen (Campus Essen), Essen 2006.
- Nr. 34: Häselhoff, I.; Meves, Y.; Munsch, D.; Munsch, S.; Schulte-Euler, D.; Thorant, C.: Anforderung an eine verbesserte Lehrqualität – Qualitätsplanung mittels House of Quality. Universität Duisburg-Essen (Campus Essen), Essen 2007.
- Nr. 35: Zelewski, S.: Das ADL-Modell der Prinzipal-Agent-Theorie für die Just-in-Time-Produktionssteuerung – Darstellung, Analyse und Kritik. Universität Duisburg-Essen (Campus Essen), Essen 2008.
- Nr. 36: Peters, M. L.; Zelewski, S.: Analyse der Effizienzentwicklung von Bankfilialen mithilfe des Operational Competitiveness Ratings (OCRA). Universität Duisburg-Essen (Campus Essen), Essen 2010.
- Nr. 37: Peters, M. L.; Zelewski, S.: Fallstudie zu PORTERS generischen Wettbewerbsstrategien im Kontext nachhaltigen Wirtschaftens. Universität Duisburg-Essen (Campus Essen), Essen 2010.
- Nr. 38: Peters, M. L.; Zelewski, S.: Erweiterung von EATWOS um die Berücksichtigung von Satisfizierungsgrenzen für Inputs. Universität Duisburg-Essen (Campus Essen), Essen 2012.
- Nr. 39: Bergenrodt, D.; Jene, S.; Zelewski, S.: Implementierung des  $\tau$ -Werts. Universität Duisburg-Essen (Campus Essen), Essen 2013.