



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Algorithmen für das Moving-Target Travelling Salesman Problem

Felix Greuling

27.01.2019

IM FOCUS DAS LEBEN



Überblick

Einleitung

Grundlagen

Zwei-orthogonale-Achsen im MT-TSP

Ergebnisse

Zusammenfassung und Ausblick

Einleitung

Travelling Salesman Problem (TSP)

- Optimierungsproblem aus der Kombinatorik
- Reihenfolge an Zielen, sodass die Tourzeit minimal ist
- Tour startet und endet im selben Ziel
- NP-vollständig

Travelling Salesman Problem (TSP)

- Optimierungsproblem aus der Kombinatorik
- Reihenfolge an Zielen, sodass die Tourzeit minimal ist
- Tour startet und endet im selben Ziel
- NP-vollständig

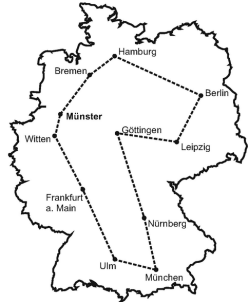
Travelling Salesman Problem (TSP)

- Optimierungsproblem aus der Kombinatorik
- Reihenfolge an Zielen, sodass die Tourzeit minimal ist
- Tour startet und endet im selben Ziel
- NP-vollständig

Travelling Salesman Problem (TSP)

- Optimierungsproblem aus der Kombinatorik
- Reihenfolge an Zielen, sodass die Tourzeit minimal ist
- Tour startet und endet im selben Ziel
- NP-vollständig

Einleitung



Moving-Target-TSP (MT-TSP)

- Im Jahre 1998 von Helvig et al. erwähnt
- Ziele sind nun nicht mehr stationär
- Problematik bleibt die selbe

Moving-Target-TSP (MT-TSP)

- Im Jahre 1998 von Helvig et al. erwähnt
- Ziele sind nun nicht mehr stationär
- Problematik bleibt die selbe

Moving-Target-TSP (MT-TSP)

- Im Jahre 1998 von Helvig et al. erwähnt
- Ziele sind nun nicht mehr stationär
- Problematik bleibt die selbe

Einleitung



Grundlagen

- Ziele

$$Z = (z_1, \dots, z_n)$$

- Startpositionen

$$P = (p_1, \dots, p_n)$$

- Geschwindigkeiten

$$V = (v_1, \dots, v_n)$$

- Verfolger

$$\kappa = (p_\kappa, v_\kappa)$$

MT-TSP in einer Dimension

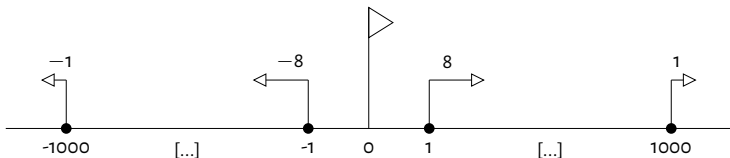


Abbildung: Eindimensionaler Fall mit jeweils zwei Zielen auf beiden Seiten des Ursprungs.

- Ziele $Z = \{(-1000, -1), (-1, -8), (1, 8), (1000, 1)\}$
- Verfolger $\kappa = (0, 10)$
- Betrachten für optimale Strategie Wendepunkte
- $Left = \{(-1, -8), (-1000, -1)\}$
- $Right = \{(1, 8), (1000, 1)\}$

MT-TSP in einer Dimension

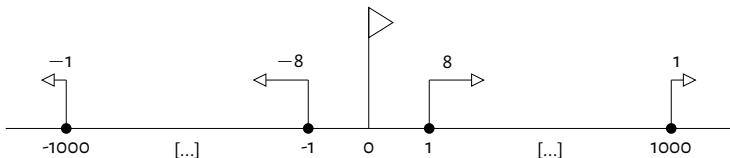


Abbildung: Eindimensionaler Fall mit jeweils zwei Zielen auf beiden Seiten des Ursprungs.

- Ziele $Z = \{(-1000, -1), (-1, -8), (1, 8), (1000, 1)\}$
- Verfolger $\kappa = (0, 10)$
- Betrachten für optimale Strategie Wendepunkte
- $Left = \{(-1, -8), (-1000, -1)\}$
- $Right = \{(1, 8), (1000, 1)\}$

MT-TSP in einer Dimension

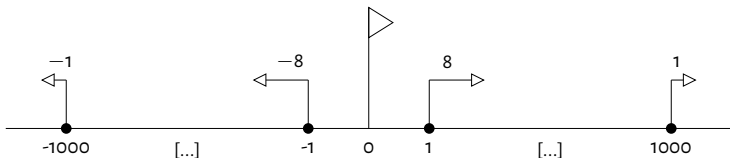


Abbildung: Eindimensionaler Fall mit jeweils zwei Zielen auf beiden Seiten des Ursprungs.

- Ziele $Z = \{(-1000, -1), (-1, -8), (1, 8), (1000, 1)\}$
- Verfolger $\kappa = (0, 10)$
- Betrachten für optimale Strategie Wendepunkte
- $Left = \{(-1, -8), (-1000, -1)\}$
- $Right = \{(1, 8), (1000, 1)\}$

MT-TSP in einer Dimension

- $Left = \{(-1, -8), (-1000, -1)\}$
- $Right = \{(1, 8), (1000, 1)\}$
- $A = (s_k, s_f)$

$$A_0 = \emptyset$$

$$A_1 = \{(-1, -8), (1, 8)\}$$

Index-Summenwert = 0

$$A_2 = \{(1, 8), (-1, -8)\}$$

Index-Summenwert = 0

$$A_3 = \{(-1, -8), (1000, 1)\}$$

Index-Summenwert = 1

$$A_4 = \{(1000, 1), (-1, -8)\}$$

Index-Summenwert = 1

$$A_5 = \{(-1000, -1), (1, 8)\}$$

Index-Summenwert = 1

$$A_6 = \{(1, 8), (-1000, -1)\}$$

Index-Summenwert = 1

$$A_7 = \{(-1000, -1), (1000, 1)\}$$

Index-Summenwert = 2

$$A_8 = \{(1000, 1), (-1000, -1)\}$$

Index-Summenwert = 2

$$A_9 = \emptyset$$

MT-TSP in einer Dimension

- $Left = \{(-1, -8), (-1000, -1)\}$
- $Right = \{(1, 8), (1000, 1)\}$
- $A = (s_k, s_f)$

$$A_0 = \emptyset$$

$$A_1 = \{(-1, -8), (1, 8)\}$$

Index-Summenwert = 0

$$A_2 = \{(1, 8), (-1, -8)\}$$

Index-Summenwert = 0

$$A_3 = \{(-1, -8), (1000, 1)\}$$

Index-Summenwert = 1

$$A_4 = \{(1000, 1), (-1, -8)\}$$

Index-Summenwert = 1

$$A_5 = \{(-1000, -1), (1, 8)\}$$

Index-Summenwert = 1

$$A_6 = \{(1, 8), (-1000, -1)\}$$

Index-Summenwert = 1

$$A_7 = \{(-1000, -1), (1000, 1)\}$$

Index-Summenwert = 2

$$A_8 = \{(1000, 1), (-1000, -1)\}$$

Index-Summenwert = 2

$$A_9 = \emptyset$$

MT-TSP in einer Dimension

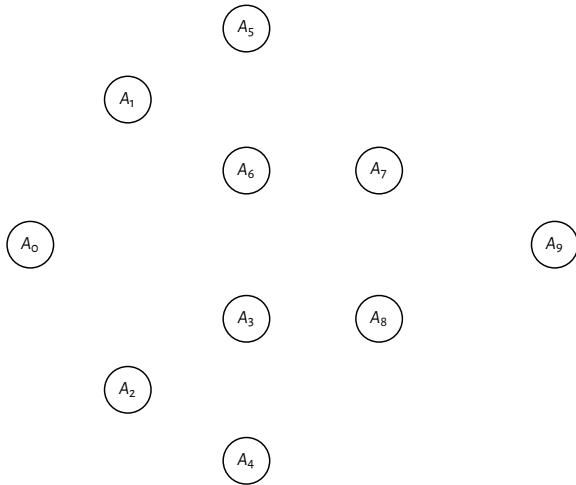


Abbildung: Zustände in aufsteigender Reihenfolge der Summenindizes.

MT-TSP in einer Dimension

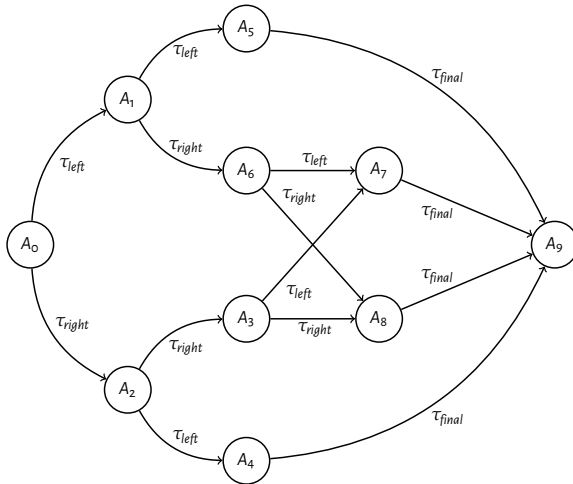


Abbildung: Zustände in aufsteigender Reihenfolge der Summenindizes.

Zwei-orthogonale-Achsen im MT-TSP

Zwei-orthogonale-Achsen im MT-TSP

- **Neue Modifikation mit zusätzlicher Achse**
- Ziele und Verfolger können sich ausschließlich auf dieser bewegen
- Dem Verfolger ist es möglich, die Achse zu wechseln

Zwei-orthogonale-Achsen im MT-TSP

- Neue Modifikation mit zusätzlicher Achse
- Ziele und Verfolger können sich ausschließlich auf dieser bewegen
- Dem Verfolger ist es möglich, die Achse zu wechseln

Zwei-orthogonale-Achsen im MT-TSP

- Neue Modifikation mit zusätzlicher Achse
- Ziele und Verfolger können sich ausschließlich auf dieser bewegen
- Dem Verfolger ist es möglich, die Achse zu wechseln

Lemma 1

In jeder optimalen Tour bei zwei orthogonalen Achsen im MT-TSP muss sich der Verfolger mit seiner maximalen Geschwindigkeit bewegen.

Lemma 2

In jeder optimalen Tour bei zwei orthogonalen Achsen im MT-TSP gelten für den Verfolger folgende Eigenschaften:

- Bewegt sich der Verfolger wegführend vom Ursprung, ändert dieser erst seine Richtung, sofern er das schnellste Ziel in seiner Richtung abgefangen hat.
- Bewegt sich der Verfolger in Richtung des Ursprungs, ändert dieser solange nicht seine Richtung, bis er den Ursprung erreicht hat.

Problem

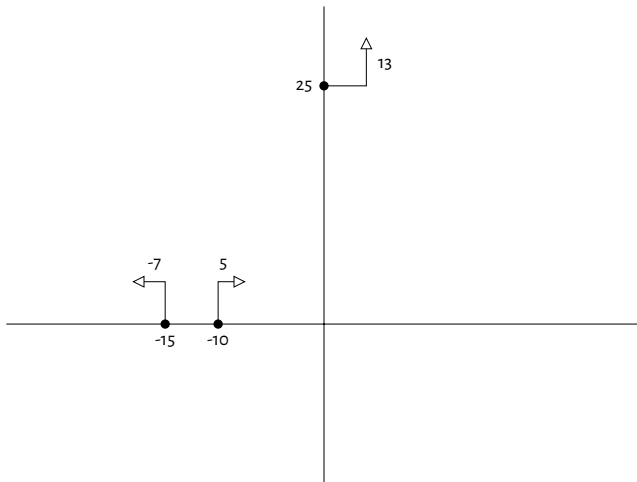


Abbildung: Bei zwei-orthogonale-Achsen-Fällen können Ziele den Ursprung überqueren, während der Verfolger auf einer Achse ein Ziel verfolgt.

Prioritätsansatz

■ Gewichte

$$\omega = (w_1, w_2, w_3)$$

■ Geschwindigkeitsfaktor

$$\varphi_1(z_i, w_1) = \frac{|v_i|}{v_K} \cdot w_1$$

■ Positionsfaktor

$$\varphi_2(z_i, w_2) = \frac{|p_i|}{v_K} \cdot a \cdot w_2$$

■ Distanzfaktor

$$\varphi_3(z_i, w_3) = \left| \frac{\|p_{\text{verfolger}}, p_i\|_1}{v_K - v_i} \right| \cdot w_3$$

■ Priorität

$$\alpha_i(z_i, \omega) := \varphi_1(z_i, w_1) + \varphi_2(z_i, w_2) - \varphi_3(z_i, w_3)$$

Prioritätsansatz

■ Gewichte

$$\omega = (w_1, w_2, w_3)$$

■ Geschwindigkeitsfaktor

$$\varphi_1(z_i, w_1) = \frac{|v_i|}{v_K} \cdot w_1$$

■ Positionsfaktor

$$\varphi_2(z_i, w_2) = \frac{|p_i|}{v_K} \cdot a \cdot w_2$$

■ Distanzfaktor

$$\varphi_3(z_i, w_3) = \left| \frac{\|p_{\text{verfolger}}, p_i\|_1}{v_K - v_i} \right| \cdot w_3$$

■ Priorität

$$\alpha_i(z_i, \omega) := \varphi_1(z_i, w_1) + \varphi_2(z_i, w_2) - \varphi_3(z_i, w_3)$$

Prioritätsansatz

■ Gewichte

$$\omega = (w_1, w_2, w_3)$$

■ Geschwindigkeitsfaktor

$$\varphi_1(z_i, w_1) = \frac{|v_i|}{v_K} \cdot w_1$$

■ Positionsfaktor

$$\varphi_2(z_i, w_2) = \frac{|p_i|}{v_K} \cdot a \cdot w_2$$

■ Distanzfaktor

$$\varphi_3(z_i, w_3) = \left| \frac{\|p_{\text{verfolger}}, p_i\|_1}{v_K - v_i} \right| \cdot w_3$$

■ Priorität

$$\alpha_i(z_i, \omega) := \varphi_1(z_i, w_1) + \varphi_2(z_i, w_2) - \varphi_3(z_i, w_3)$$

Prioritätsansatz

■ Gewichte

$$\omega = (w_1, w_2, w_3)$$

■ Geschwindigkeitsfaktor

$$\varphi_1(z_i, w_1) = \frac{|v_i|}{v_K} \cdot w_1$$

■ Positionsfaktor

$$\varphi_2(z_i, w_2) = \frac{|p_i|}{v_K} \cdot a \cdot w_2$$

■ Distanzfaktor

$$\varphi_3(z_i, w_3) = \left| \frac{\|p_{\text{verfolger}}, p_i\|_1}{v_K - v_i} \right| \cdot w_3$$

■ Priorität

$$\alpha_i(z_i, \omega) := \varphi_1(z_i, w_1) + \varphi_2(z_i, w_2) - \varphi_3(z_i, w_3)$$

Prioritätsansatz

■ Gewichte

$$\omega = (w_1, w_2, w_3)$$

■ Geschwindigkeitsfaktor

$$\varphi_1(z_i, w_1) = \frac{|v_i|}{v_K} \cdot w_1$$

■ Positionsfaktor

$$\varphi_2(z_i, w_2) = \frac{|p_i|}{v_K} \cdot a \cdot w_2$$

■ Distanzfaktor

$$\varphi_3(z_i, w_3) = \left| \frac{\|p_{\text{verfolger}}, p_i\|_1}{v_K - v_i} \right| \cdot w_3$$

■ Priorität

$$\alpha_i(z_i, \omega) := \varphi_1(z_i, w_1) + \varphi_2(z_i, w_2) - \varphi_3(z_i, w_3)$$

- In jeder Iteration des Algorithmus wird
 - die Priorität jedes Ziels berechnet
 - die Position von jedem Ziel aktualisiert
 - geprüft, ob Ziele zwischen dem betrachteten und dem vorherigen Ziel abgefangen wurden
- garantiert keine optimalen Ergebnisse
- $\mathcal{O}(n^2)$

- In jeder Iteration des Algorithmus wird
 - die Priorität jedes Ziels berechnet
 - die Position von jedem Ziel aktualisiert
 - geprüft, ob Ziele zwischen dem betrachteten und dem vorherigen Ziel abgefangen wurden
- garantiert keine optimalen Ergebnisse
- $\mathcal{O}(n^2)$

- In jeder Iteration des Algorithmus wird
 - die Priorität jedes Ziels berechnet
 - die Position von jedem Ziel aktualisiert
 - geprüft, ob Ziele zwischen dem betrachteten und dem vorherigen Ziel abgefangen wurden
- garantiert keine optimalen Ergebnisse
- $\mathcal{O}(n^2)$

Brute-Force-Ansatz

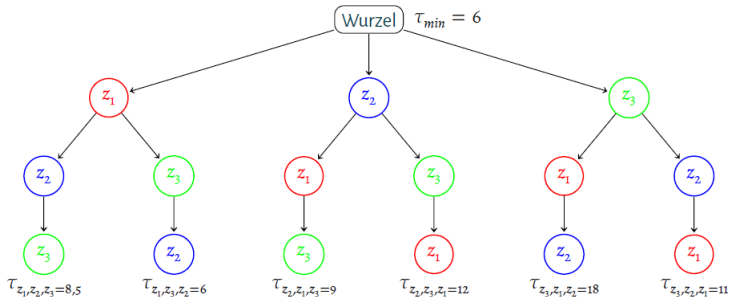


Abbildung: Brute-Force-Ansatz bei $n = 3$ Zielen.

Brute-Force-Ansatz

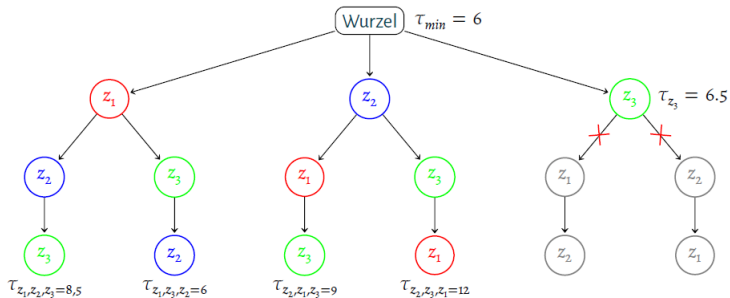


Abbildung: Brute-Force-Ansatz bei $n = 3$ Zielen.

Ergebnisse

Durchschnittliche Laufzeit des 1D-Algorithmus

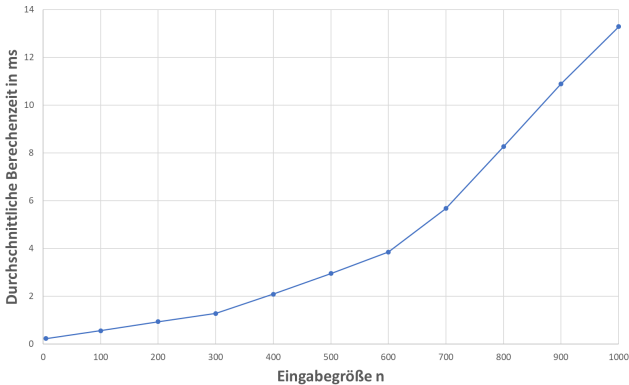


Abbildung: Berechenzeit der optimalen Tour bei bis zu 1.000 Zielen und $v_k = 40$.

Brute-Force-Algorithmus mit unterschiedlichen Eingabegrößen

n	Instanzen	Ø betr. Knoten	Ø Anteil Knoten	Ø Schnitte	Ø ber. Blätter	Fails
1	10.000	1,00	1,000000	0,00	1,00	0
2	10.000	3,75	0,939151	0,24	1,57	0
3	10.000	11,75	0,783513	0,95	3,14	0
4	10.000	37,81	0,590758	3,22	6,71	0
5	10.000	130,44	0,401344	11,87	15,02	0
6	10.000	471,76	0,241184	47,23	32,46	0
7	10.000	1.819,94	0,132852	194,96	70,33	0
8	10.000	7.353,01	0,067090	852,13	150,30	0
9	10.000	31.426,98	0,031860	3.833,61	313,76	0
10	1.000	140.342,54	0,014228	18.169,79	620,86	0
11	1.000	650.570,40	0,005996	88.862,13	1.304,56	0
12	100	3.562.601,51	0,002736	457.589,77	3.246,11	0
13	100	8.184.931,87	0,000484	1.523.396,29	4.303,92	22
14	10	5.033.246,67	0,000021	1.147.070,33	2.025,67	7

Tabelle: Beschneidungen im Suchbaum des Brute-Force-Algorithmus bei einer Eingabe von $1 \leq n \leq 14$ Zielen und $v_K = 40$.

Güte des Prioritäts-Algorithmus

n	Instanzen	$\max\{v_i\}$	\varnothing Güte	max Güte	opt. Touren	Anteil opt.
8	10.000	20	1,10	2,59	6313	0,63
8	10.000	40	1,41	11,93	3341	0,33
8	10.000	60	3,53	328,88	1390	0,14
10	1.000	20	1,14	2,15	518	0,52
10	1.000	40	1,54	9,22	220	0,22
10	1.000	60	5,32	282,26	83	0,08
12	100	20	1,16	2,31	43	0,43
12	100	40	1,56	4,79	12	0,12
12	100	60	5,30	82,29	5	0,05

Tabelle: Güte des Prioritäts-Algorithmus bei $\omega = (87, 34, 31)$ und $v_K = 61$.

Zusammenfassung und Ausblick

- 1D-Algorithmus auch bei großen Instanzen effizient
- Prioritätsalgorithmus abhängig von den gewählten Gewichten
- Brute-Force-Algorithmus effizient, sofern schnell ein kleines τ_{min} gefunden wird.

- 1D-Algorithmus auch bei großen Instanzen effizient
- Prioritätsalgorithmus abhängig von den gewählten Gewichten
- Brute-Force-Algorithmus effizient, sofern schnell ein kleines τ_{min} gefunden wird.

- 1D-Algorithmus auch bei großen Instanzen effizient
- Prioritätsalgorithmus abhängig von den gewählten Gewichten
- Brute-Force-Algorithmus effizient, sofern schnell ein kleines τ_{min} gefunden wird.

- Weiterhin optimale Heuristik mit polynomieller Laufzeit gesucht
- k -Achsen im MT-TSP

- Weiterhin optimale Heuristik mit polynomieller Laufzeit gesucht
- k -Achsen im MT-TSP

Danke für Ihre Aufmerksamkeit!

References

-  Christian Grimme and Jakob Bossek, *Grundbegriffe und komplexität*, pp. 1–25, Springer Fachmedien Wiesbaden, Wiesbaden, 2018.
-  Christopher S Helvig, Gabriel Robins, and Alex Zelikovsky, *The moving-target traveling salesman problem*, *Journal of Algorithms* **49** (2003), no. 1, 153–174.
-  Wikipedia, *Versorgungsschiff*—*Wikipedia, the free encyclopedia*, <http://de.wikipedia.org/w/index.php?title=Versorgungsschiff&oldid=180421247>, 2020, [Online; accessed 19-January-2020].

Lemmata I

Lemma

In jeder optimalen Tour bei zwei orthogonalen Achsen im MT-TSP muss sich der Verfolger mit seiner maximalen Geschwindigkeit bewegen.

Beweis.

Der Beweis basiert darauf, dass in jedem Fall eine Reduzierung auf den Beweis von [HRZo3] vorgenommen wird. Nehmen dafür eine Fallunterscheidung vor:

1. Das nächste Ziel des Verfolgers liegt auf der selben Achse:
Mit dem Beweis für 1D-Fälle in [HRZo3] gilt dies auch für diesen Fall.
2. Das nächste Ziel des Verfolgers bewegt sich auf der anderen Achse:
Wir nehmen für einen Widerspruch an, der Verfolger bewegt sich mit $v < v_K$. Dies ist äquivalent dazu, dass der Verfolger an seiner aktuellen Position eine Zeit τ wartet und sich dann mit v_K weiterbewegt, um dann das nächste Ziel z einzuholen. Dabei befindet sich z auf der anderen Achse. Nach der Wartezeit erreicht der Verfolger an Zeitpunkt t_1 den Ursprung und holt das Ziel z an der Position p zum Zeitpunkt t_2 ein. Nehmen nun an, dass der Verfolger sich direkt zum Mittelpunkt bewegt. Bis zum Eintreffen des Zeitpunktes t_1 wartet der Verfolger nun wieder die Zeit τ . Das Ziel z wird nun zum selben Zeitpunkt t_2 bei p erreicht, wie im vorherigen Szenario. Dies wird nun fortgeführt, indem der Verfolger nicht im Ursprung wartet, sondern von diesem aus p direkt erreicht. Bis zum Zeitpunkt t_2 wird nun wieder für die Dauer von τ gewartet. Außerdem kann der Verfolger schon zu einem Zeitpunkt $t_1 \leq t_s \leq t_2$ abfangen, sofern sich z vom Verfolger wegbewegt. Werden die Wartezeiten nun jeweils auch für alle restlichen Ziele der Tour hinten angehängt, resultiert dies letztendlich in Wartezeit am Ende der Tour, was offensichtlich nicht optimal ist. Dieser Fall ist demnach nur eine Erweiterung des 1D-Fall-Beweises um den Ursprung zwischen Zielen, die auf unterschiedlichen Achsen liegen.

In jedem der Fälle wird eine Wartezeit erzeugt, welche an das Ende der Tour verschoben werden kann. Somit ist die Tour offensichtlich nicht mehr optimal. Der Verfolger bewegt sich also zu jeder Zeit mit v_K . □

Lemmata III

Lemma

In jeder optimalen Tour bei zwei orthogonalen Achsen im MT-TSP gelten für den Verfolger folgende Eigenschaften:

- *Bewegt sich der Verfolger wegführend vom Ursprung, ändert dieser erst seine Richtung, sofern er das schnellste Ziel in seiner Richtung abgefangen hat.*
- *Bewegt sich der Verfolger in Richtung des Ursprungs, ändert dieser solange nicht seine Richtung, bis er den Ursprung erreicht hat.*
- *Der Verfolger befindet sich im Ursprung, kann sich dieser nicht in die Richtung bewegen, aus die er gerade kommt.*

Beweis.

Für den Beweis des Lemmas müssen beide Eigenschaften bewiesen werden. Fallunterscheidung:

1. Der Verfolger bewegt sich wegführend vom Ursprung in Richtung des schnellsten Ziels z_1 :
Der Verfolger kann sich dabei auf einer beliebigen Position der vier Seiten des Ursprungs¹ zum Zeitpunkt t_1 befinden. Der Beweis für diesen Fall ist ähnlich zu dem Beweis für Wendepunkte aus [HRZ03].
Wir nehmen für einen Widerspruch an, dass der Verfolger in einer optimalen Tour seine Richtung zum Zeitpunkt t_2 ändert, bevor z_1 abgefangen wurde. Damit gibt es ein kleines $\delta > 0$, sodass in dem Zeitraum zwischen $t_2 - \delta$ und $t_2 + \delta$ der Verfolger nur zum Zeitpunkt t_2 seine Richtung ändert. Wir nehmen als alternative Tour an, dass der Verfolger zum Zeitpunkt $t_2 - \delta$ anhält und bis $t_2 + \delta$ im selben Punkt p wartet. Anschließend setzt er die ursprüngliche Tour fort. Jegliche Ziele² befinden sich in dem Zeitraum von $t_2 - \delta$ bis $t_2 + \delta$ zwischen p und z_1 und sind nach Definition langsamer, als z_1 , sodass diese z_1 nicht überholen können. Mit der Wartezeit wird also die Tourzeit nicht reduziert.
Betrachten als Ausnahme das Ziel z_2 von der anderen Seite des Ursprungs. Im Gegensatz zum eindimensionalen Fall kann ein solches Ziel alle Ziele auf der anderen Seite des Ursprungs überholen. Dabei befindet sich z_2 zum Zeitpunkt t_1 zwischen dem Ursprung und p . Sei z_2 schneller als z_1 . Somit erreicht der Verfolger z_2 zwischen p und z_1 . Allerdings wird damit die Tourzeit ebenfalls nicht verbessert, da z_2 sowieso auf dem Rückweg zum Ursprung abgefangen wird.
Selbst wenn der Verfolger vor z_1 die Richtung ändert, um ein Ziel auf einer der anderen drei Seiten abzufangen, hat der Verfolger in der vorherigen Zeit nicht das Ziel z_1 verfolgt und nur Ziele abgefangen, die sich auf dem Weg dorthin befanden. Demnach führt die Zeit, die nicht für das Verfolgen des schnellsten Ziels aufgewendet wurde, zu einer Wartezeit an einer Position. Somit wäre die Tour nach Lemma 1 nicht optimal.

2. Der Verfolger hat soeben das schnellste Ziel auf einer Achse abgefangen und bewegt sich nun zum Ursprung: Dieses Szenario gilt ebenfalls für alle vier Seiten des Ursprungs. Dabei gelten dieselben Bedingungen, weshalb es reicht, einen generellen Fall zu zeigen.
Nach Definition kann sich nun der Verfolger nicht umdrehen und ein anderes Ziel auf der Achse einholen. Damit hätte er eine Zeit lang nicht das schnellste Ziel einer Richtung eingeholt, was nach [HRZo3] äquivalent zum Warten in einem Punkt ist. Dies resultiert in eine nicht optimale Tour.
Somit muss der Verfolger zunächst den Ursprung erreichen. Nach dem Erreichen des Ursprungs bewegt sich der Verfolger in eine der anderen drei Richtungen. Ab diesem Zeitpunkt gilt wieder der erste Fall, welcher bereits bewiesen wurde.
3. **Der Verfolger befindet sich im Ursprung:**
Befindet sich der Verfolger im Ursprung kann sich dieser nicht in die Richtung bewegen, aus welcher der Verfolger gerade kommt. Damit würde er eine Zeit lang nicht das schnellste Ziel verfolgen. Dies ist wieder äquivalent zu einer Wartezeit an einem Punkt und resultiert in eine nicht optimale Tour. Befindet sich der Verfolger zu Beginn der Tour im Ursprung, kann der Verfolger zwischen jeder der vier Richtungen wählen. □

¹Die Position kann sich ebenfalls auf dem Ursprung selbst befinden.

²Ziele, die auf der Seite des Ursprungs gestartet sind.