Bachelorarbeit

Approximationsalgorithmen für Set-Cover-Probleme mit Rechtecken

Maike Rösch

Abgabedatum: 6. Juli 2018
Betreuer: Dr. habil. Joachim Spoerhase

Julius-Maximilians-Universität Würzburg
Lehrstuhl für Informatik I
Algorithmen, Komplexität und wissensbasierte Systeme

Zusammenfassung

Das Ziel der vorliegenden Bachelorarbeit war es, zwei geometrische Fälle des Set Cover Problems zu untersuchen und gegebenfalls eine Annäherung mithilfe von linearer Programmierung zu bestimmen. Hierbei dienten einige Beweisideen ähnlicher Fragestellungen als Grundlage für eigene Versuche. Für das Slabs-Cover Problem ergab sich hierdurch ein 2-LP-Approximationsalgorithmus. Für das Rechtecke-Cover Problem wurden basierend auf existierenden Arbeiten eigene Ansätze entwickelt. Diese potentiellen Herangehensweisen wurden vorgeschlagen und diskutiert.

Inhaltsverzeichnis

1 Einleitung

Eines der grundlegenden Optimierungsprobleme ist das sogenannte Set-Cover-Problem. Dabei handelt es sich um zwei gegebene Mengen, bei der eine die andere überdeckt. Bei diesem Problem geht es darum, eine gegebene Grundmenge mit möglichst wenig Mengen aus einer gegebenen Mengenfamilie zu überdecken. Genauer gesagt ist eine Grundmenge U sowie eine Familie von Teilmengen mit $\bigcup\limits_{V \in F} V = U$ gegeben. Das Ziel besteht darin, eine kleinste Teilfamilie $F' \subseteq F$ zu finden, bei der noch gilt $\bigcup\limits_{V \in F'} V = U$.

Im allgemeinen Fall ist dieses Problem NP-schwer [HMRR14]. Das heißt unter der Annahme, dass $P \neq NP$ gibt es für ein allgemeines Set Cover Problem keinen exakten Polynomialzeit Algorithmus. Bei einem Approximationsalgorithmus des Problems ergibt sich bestenfalls ein logarithmischer Faktor zur optimalen Lösung in einer polynomiellen Laufzeit, das heißt es existiert ein Approximationsalgorithmus mit logarithmischem Faktor zur optimalen Lösung und es gibt einen Beweis, dass das die bestmögliche Approximationsgüte ist. [Fei98].
Damit ist die Frage nach einem Polynomialzeitalgorithmus für den allgemeinen Fall bereits geklärt.
Daher ist die Frage nach Polynomialzeit-Approximationsalgorithmen für Spezialfälle interessant. Zum Beispiel gibt es für Halbräume im $\mathbb{R}^2$ sowohl im gewichteten, als auch im ungewichteten Fall einen exakten Algorithmus [HMRR14].

Obwohl Set-Cover selbst in vielen natürlichen Spezialfällen noch NP-schwer ist, lassen sich in einigen Fällen sublogarithmische Approximationsfaktoren erzielen. Hierbei gibt es ganze Klassen an Problemen, die einen PTAS haben und weitere, die einen konstanten Approximationsalgorithmus erreichen. Zum Beispiel gibt es für Probleme mit so genannter Shallow-Cell-Complexity und Netzwerkstrukturen konstante Approximationsalgorithmen [CGKS12]. Shallow-Cell-Complexity bedeutet hier, dass durch starke Strukturen die Elemente im Universum U nur in wenigen Teilmengen der Familie enthalten sind [CGKS12].

Gerade das Set Cover Problem und seine Anwendungsfälle sind auch über ein theoretisches Konzept hinaus sehr interessant. So kann zum Beispiel das Scheduling-Problem auf Set Cover reduziert werden. Dabei ist das folgende Scheduling-Problem hier definiert über die Eingabe aus n Aufträgen, die alle willkürliche Anfangszeit, Größe und monotone Kostenfunktionen, abhängig von dem Abschlusszeitpunkt, haben. Die Aufgabe ist einen Ablaufplan oder auch Schedule zu finden mit minimalen Kosten.
Das dazu verwendete Set Cover Problem ist das achsenparallele Rechteck-Set Cover Pro-

blem [BP14].

Die Approximationsalgorithmen funktionieren zum Beispiel über Greedy-Ansätze, Strukturanalysen oder Lineare Programmierung, wobei gerade letzteres in dieser Arbeit relevant wird. Ziel ist, anhand von zwei speziellen geometrischen Problemen mögliche Ansätze für einen Beweis solcher Probleme darzustellen.

In dieser Arbeit werden zwei Probleme betrachtet. In dem einen überdecken achselparallele Slabs im $\mathbb{R}^2$ Punkte. Bei dem Zweiten betrachten wir als Überdeckende Menge Rechtecke, die achsenparallel sind und sich nur so schneiden dürfen, dass kein Eckpunkt des einen im anderen Rechteck liegt. Als Grundmenge betrachten wir wieder Punkte im $\mathbb{R}^2$. Für beide Probleme gilt, man kann sie über Spezialfälle des allgemeinen Rechteck-Punkte Set Covers mit achsenparallelen Rechtecken lösen. Für die beiden betrachteten Probleme gilt, dass kein beliebig genaues Approximationsschema gefunden werden kann, weswegen ein konstanter Approximationsalgorithmus erstrebenswert ist [CG14].

Es gibt für dieses allgemeine achsenparallele Rechtecke Problem einen Approximationsalgorithmus mit der Güte $O(\log \log OPT)$, wobei OPT eine optimale Lösung ist [AES10]. Es wurde ebenfalls gezeigt, dass diese Schranke auch scharf ist, dass heißt es existiert kein besserer Approximationsalgorithmus für das allgemeine Rechteckproblem [PT10]. Interessant ist, dass die Arbeit, die beweist, dass für das LP-duale Rechteck-Cover-Problem ein Spezialfall, der die Schneidebedingung, die in dieser Arbeit gefordert wird ebenfalls erfüllt, über die Methode der Epsilon-Netze eine untere Schranke gefunden ist.

Umso überraschender ist, dass das Slab-Cover-Problem eine 2-Approximation hat.
Für das Rechteck-Cover-Problem mit spezieller Schneidebedingung wird ein Ansatz vorgestellt, der potenziell eine gute Approximationsgüte liefern könnte. Im Rahmen der Arbeit konnte kein Beispiel gefunden werden, in dem der Algorithmus schlechter als ein 3-LP-Approximationsalgorithmus ist.

2 Grundlagen

Um sich mit den speziellen geometrischen Problemen beschäftigen zu können, bedarf es zuerst der Definition und Wiederholung von Grundlagen. Zu Beginn soll ein lineares Programm formal definiert werden.

Definition 2.1. Ein *lineares Programm* ist in Matrixnotation wie folgt notiert:

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{subject to} & Ax \geq b \\ & x \geq 0 \end{array}$$

Hierbei c ein Kostenvektor $\in \mathbb{R}^m$, A eine, die zu erfüllenden linearen Ungleichungen beschreibende Matrix $\in \mathbb{R}^{n \times m}$, b ein Vektor $\in \mathbb{R}^n$, indem die Bedingungen festgehalten werden und x den Lösungsvektor $\in \mathbb{R}^m$ darstellt [LRS11][S.12].

Da nun das lineare Programm definiert ist, stellt sich nun die Frage, wann ein x eine zulässige Lösung dazu ergibt.

Definition 2.2. Zu einem linearen Programm wie in 2.1 ist eine *zulässige Lösung* ein Vektor $x \in \mathbb{R}^m$, der $Ax \geq b$ erfüllt, sowie die linearen Ungleichungen $x \geq 0$.
Die Menge aller zulässigen Lösungen wird mit P beschrieben.

In einem Optimierungsproblem gilt es weniger die Existenz einer Lösung zu beweisen, als eine optimale Lösung zu finden.

Definition 2.3. Eine *optimale Lösung* ist eine zulässige Lösung wie in 2.2, bei der das Minimum der Funktion $c^\top x$ aus 2.1 angenommen wird.

Ein Spezialfall der optimalen Lösungen bilden Extrempunktlösungen, die besonders interessieren, da einige Algorithmen genau diese ausgeben und sie gut charakterisiert sind.

Definition 2.4. Eine zulässige Lösung x ist eine *Extrempunktlösung*, wenn es keinen Vektor $y \neq 0 \in \mathbb{R}^m$ gibt, mit $x + y$, $x - y \in P$.
Gibt es eine optimale Lösung, so ist gibt es eine Extrempunktlösung, die eine optimale Lösung ist.
Es ist bekannt, dass eine Extrempunktlösung durch das Rank Lemma charakterisiert wird.
Dieses besagt, dass genau dann, wenn die Anzahl s, der scharf erfüllten linear unabhängigen Ungleichungen aus A, $s \geq m$ erfüllt, handelt es sich um eine Extrempunktlösung [LRS11][S.3-4].

Da diese Arbeit sich mit ganzzahligen Lösungen beschäftigt, ist die Frage nach ganzzahligen Extrempunkten bei Problemen eine relevante.

Definition 2.5. Eine Extrempunktlösung wie in 2.4 wird *integral* genannt, wenn jede Koordinate ganzzahlig ist. Ein Problem heißt *integral*, wenn jede Extrempunktlösung integral ist [LRS11][S.13].

Um Beweise über Set Cover zu führen, muss das Problem formal eingeführt werden und auch sein Lineares Programm definiert werden.

Definition 2.6. Ein *Set Cover Problem* wird auch über ein lineares Programm wie in 2.1 definiert.
Die überdeckende Menge O habe die Größe m, die Grundmenge U die Größe n. Dann beschreibt die Matrix A, welche Elemente aus O welche aus U überdecken.
Also $\forall i \in \{1, ..., n\}, j \in \{1, .., m\}$:

$$a_{ij} = \begin{cases} 1 & \text{wenn } j\text{-tes Element aus } O \text{ das } i\text{-te aus } U \text{ überdeckt} \\ 0 & \text{sonst} \end{cases}$$

b ist ein Vektor $\in \mathbb{R}^n$, bei dem jeder Eintrag 1 ist, da jedes Element aus U zu überdecken ist.
c kann Kosten für jedes der Elemente aus O enthalten und hat im ungewichteten Fall überall eine 1 als Eintrag.
Man unterscheidet zwischen ganzzahligem Set-Cover für die gilt, dass die Lösung $x \in \mathbb{R}^m$ für jede Koordinate $x_i \in 0, 1$ erfüllt für $i \in 1, ..., m$ und LP-Relaxierung, bei dem für jede Koordinate $x_i \geq 0$ mit $i \in 1, ..., m$ gilt.

Im weiteren Verlauf wird nicht nur Set Cover, sondern ein im ersten Moment ähnlich klingendes Problem verwendet, welches an dieser Stelle bereits formal eingeführt werden soll.

Definition 2.7. Ein *Stabbing Problem* kann ebenfalls über ein lineares Programm definiert werden. Der Begriff stabben bedeutet, dass ein Objekt, das Stabbendem, mindestens zum Teil in einem Anderen, dem Gestabbten, liegt.
Die stabbende Menge O habe die Größe m, die zu Stabbende U die Größe n. Dann beschreibt die Matrix $A^\top$ welche Elemente aus U welche aus O stabben.
Also $\forall i \in \{1, ..., n\}, j \in \{1, .., m\}$:
$a_{ji} = \begin{cases} 1 & \text{wenn } j\text{-tes Element aus } O \text{ von dem } i\text{-ten aus } U \text{ gestabbt wird} \\ 0 & \text{sonst} \end{cases}$

b ist ein Vektor $\in \mathbb{R}^m$, bei dem jeder Eintrag 1 ist, da jedes Element aus U zu stabben ist.
c kann Kosten für jedes der Elemente aus O enthalten und hat im ungewichteten Fall überall eine 1 als Eintrag [BDGMS12].
Der Unterschied zwischen ganzzahligem Problem und LP-Relaxierung ist analog zu Definition 2.6.

Um verschiedene Approximationsalgorithmen zu vergleichen sollte man eine Approximationsgüte einführen, die beschreibt, wie akkurat ein Algorithmus funktioniert.

Definition 2.8. Die *Approximationsgüte* eines Algorithmus zu einem Problem berechnet sich über das Verhältnis zwischen einer optimalen Lösung x^* und der Ausgabe x des Algorithmus.
Die Güte ist bei Minimierungsproblemen definiert als:

$$\max_{y\, ist\, Eingabe} \frac{Kosten\, von\, Ausgabe\, des\, Algorithmus\, bei\, Eingabe\, y}{Optimale\, Kosten\, zu\, Eingabe\, y}.$$

Häufig wird auch eine obere Schranke für diesen Wert als Approximationsgüte definiert.

Bei der Approximationsgüte gibt es verschiedene Klassen. Dabei ist der gewählte Algorithmus umso besser, je näher seine Güte an 1 liegt. Nun werden zwei in der Arbeit relevanten Klassen eingeführt werden.

Definition 2.9. Ein *PTAS* (polynomial-time apprioximation scheme) ist ein Approximationsalgorithmus, der für eine Instanz eines Optimierungsproblems und einen Parameter $\varepsilon > 0$ in Polynomialzeit eine Lösung mit Approximationsgüte $1 + \varepsilon$ zur optimalen Lösung bei Minimierungsproblemen (bei Maximierungsproblemen ist die Approximationsgüte $1 - \varepsilon$) ausgibt.
Wichtig ist, dass der Algorithmus polynomiell in Problemgröße sein muss, aber von ε auch nicht-polynomiell abhängen darf.

Definition 2.10. APX ist eine Menge von Optimierungsproblemen, für die es einen Polynomialzeit-Approximationsalgorithmus gibt mit konstanter Approximationsgüte.

LP-Duale Probleme zu Linearen Programmen geben mögliche Schranken mit jeder zulässigen Lösung und haben bei starker LP-Dualität sogar diesselben Extrempunkte [LRS11][S.19].

Definition 2.11. Zu einem Lineare Programm wie in 2.1 gibt es das korrespondierende *LP-duale Programm* der folgenden Form:

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & A^\top y \leq c \\ & y \geq 0 \end{array}$$

wobei c ein Kostenvektor $\in \mathbb{R}^m$, A eine die zu erfüllenden linearen Ungleichungen beschreibende Matrix $\in \mathbb{R}^{n \times m}$ ist, b ein Vektor $\in \mathbb{R}^n$, indem die Beschränkung festgehalten werden und x den Lösungsvektor $\in \mathbb{R}^m$ darstellt.
Wie zuvor sind das ganzzahlige Problem und die LP-Relaxierung hier analog wie in 2.6 definiert.
Das ist interessant zu betrachten, da zum Beispiel jede zulässige Lösung des dualen Programms eine untere Schranke für die optimale Lösung des primären linearen Programms ergibt [LRS11][S.18].

Definition 2.12. Wenn man nun das LP-duale Programm zum Stabbing-Problem 2.7 betrachtet, so erhält man das so genannte *Packing-Problem*. Dieses kann formal wie folgt dargestellt werden:

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & Ay \leq c \\ & y \geq 0 \end{array}$$

b ist ein Vektor $\in \mathbb{R}^m$, bei dem jeder Eintrag 1 ist, da jedes Element aus U zu stabben ist.
c kann Kosten für jedes der Elemente aus O enthalten und hat im ungewichteten Fall überall eine 1 als Eintrag. $y \in \mathbb{R}^m$ ist die gesuchte Variable. Die Matrix A ist die aus 2.7.

Es existiert eine Methode, mit der stabil in Polynomialzeit eine optimale Extrempunktlösung bestimmt werden kann für relaxierte lineare Programme.

Lemma 2.13. *Über die Innerer-Punkt-Methode lässt sich ein relaxiertes Lineares Problem in polynomieller Zeit lösen, allerdings muss diese Lösung nicht ganzzahlig sein.*
Die erhaltene Lösung ist eine Extrempunktlösung und erfüllt damit auch das Rank Lemma aus Definition 2.4 [Ste10].

Da die Extrempunktlösungen von relaxierten linearen Programmen aber nicht zwingend ganzzahlig sein müssen und die gesuchte Lösung meist ganzzahlig sein soll, ist die Frage zwischen dem Verhältnis einer ganzzahligen und einer relaxierten-LP Lösung interessant.

3 Überdecken von Punkten durch Streifen

3.1 Eigenschaften von COVER-SLABS

Definition 3.1. Das erste in dieser Arbeit betrachtete Problem wird als *COVER-SLABS* bezeichnet. Die überdeckende Menge O besteht aus achsenparallelen Slabs im $\mathbb{R}^2$ der Form $[a_i, b_i] \times [-\infty, \infty]$ oder $[-\infty, \infty] \times [a_i, b_i]$, wobei a_i und b_i aus $\mathbb{R}$ sind.
Die zu überdeckende Menge U besteht aus Punkten im $\mathbb{R}^2$.
Es wird das ungewichtete Problem betrachtet, das heißt alle Slabs haben dasselbe Gewicht [CG14].

Für dieses Problem ist bereits bekannt, dass es nicht beliebig angenähert werden kann.

Lemma 3.2. *Es gibt keinen PTAS für COVER-SLABS [CG14].*

Es gibt jedoch die Möglichkeit eines konstanten Approximationsalgorithmus. Die Idee für ein solches Verfahren liefert folgendes Lemma:

Lemma 3.3. *Seien P, P_1, P_2 Set Cover Probleme, wobei P_1 und P_2 jeweils einen α_1- und α_2-LP-Approximationsalgorithmus haben.*
Sei P definiert wie in Definition 2.6.
Dann hat das Problem P einen $(\alpha_1 + \alpha_2)$-LP-Approximationsalgorithmus, wenn für jede Instanz die Menge O aufgeteilt werden kann in O_1 und O_2, so dass für jede Partition von U in U_1 und U_2, bei der U_1 von O_1 und U_2 von O_2 überdeckt wird, gilt, dass sie jeweils Instanzen von P_1 und P_2 bilden[TMC18].

Nun bleibt die Frage, in welcher Weise man die Slabs partionieren könnte und welche Approximationsalgorithmen die so entstehenden Probleme P_1 und P_2 haben.

Lemma 3.4. *Ein d-Intervall-Stabbing Problem ist eine Stabbing-Problem , bei dem die stabbende Menge $\subset \mathbb{R}$, und ein Element in der zu stabbenden Menge aus genau d kompakten nichtleeren Intervallen $\subset \mathbb{R}$ besteht.*
Es gibt zu dem d-Intervall Stabbing Problem einen d-LP-Approximationsalgorithmus. [BDGMS12]

Für den in der Arbeit betrachteten Fall reicht eine schwächere Aussage aus.

Lemma 3.5. *Das Stabbing-Problem, bei dem die zu stabbende Menge aus kompakten, nichtleeren Intervallen besteht, und die stabbende Menge aus Punkten in $\mathbb{R}$, hat einen 1-LP-Approximationsalgorithmus.*

Der Beweis der 1-Approximationsgüte von 1-Intervall-Stabbing kann abgewandelt auf Intervall-Covering angewandt werden und ist deswegen interessant.

Definition 3.6. Man definiere den *Hypergraph* $G_H = (V, E)$ zu einer Probleminstanz an Intervallen, und diese Intervalle stabbende Punkte. Wobei H die Menge zu stabbender Intervalle ist, $V = H$ gilt und E aus all den Teilmengen $I \subseteq H$, so dass es einen Punkt $p \in \mathbb{R}$ gibt, dass genau alle Intervalle in I stabbed. Solch ein Punkt p ist ein Repräsentant der Hyperkante $I \in E$ [BDGMS12].

Neben der Hypergraphen werden auch die folgenden Eigenschaft in diesem Beweis verwendet.

Definition 3.7. Für eine Matrix $\mathrm{A} \in \mathbb{R}^{m \times n}, n, m \in \mathbb{N}$ gilt, sie ist *total unimodular*, wenn alle quadratischen Untermatizen von ihr Determinante $= 0$ oder $= \pm 1$ erfüllen [Tru78].

Definition 3.8. Die sogenannte *consecutive ones* Eigenschaft (CIP), also weiterführende Einsen Eigenschaft ist definiert für binäre Matrizen. Eine solche Matrix erfüllt CIP für Zeilen, wenn es eine Permutation der Spalten gibt, so dass die Einsen in jeder Zeile aufeinanderfolgend sind. Analog lässt sich auch CIP für Spalten definieren. [MPT+98]
Man weiß, wenn eine Matrix die CIP Eigenschaft erfüllt, so ist sie total unimodular [BDGMS12][S.3].

Mit diesen Definitionen ergibt sich das folgende Lemma:

Lemma 3.9. *Wenn eine Matrix total unimodular ist, so ist die Menge an zulässigen Lösungen, die sie repräsentiert integral [HK10].*

Der Beweis in [BDGMS12][S.3] zu 3.5 sieht nun wie folgt aus:

Beweis. Das Lineare Programm ist Integral, da die Adjazenzmatrix des Hypergraph die CIP besitzt und dementsprechen total unimodular ist.

□

Analog behandelt die Arbeit [BDGMS12] auch die LP-duale Problemstellung zu Intervall-Stabbing und begründet auch hier Integralität:

Lemma 3.10. *Das zu dem Intervall-Stabbing Problem duale Intervall-Packing Problem ist ebenfalls integral [BDGMS12][S.3].*

Über den Beweis dieser LP-dualen Fragestellung kann man nun beweisen, dass Intervall-Punkt-Covering integral ist.

Satz 3.11. *Set Cover, bei dem die Menge O aus kompakten, nichtleeren Intervallen und U aus Punkten in $\mathbb{R}$ besteht, hat einen 1-LP-Approximationsalgorithmus.*

Beweis. Betrachte den Hypergraph zu dem LP-dualem Linearen Programm zu 1-Intervall-Stabbing. Da seine Adjazenzmatrix total unimodular ist und diese auch der Adjazenzmatrix des Hypergraphen von Intervall-Covering entspricht, ist auch Intervall-Covering nach 3.9 integral.

□

Damit sind die Grundlagen gelegt, um das folgende Verfahren zu motivieren.

3.2 Das Vorgehen

Um 3.3 anwenden zu können, gilt es nun, das Slabs-Cover-Problem gut aufzuteilen und die Teilprobleme zu lösen.

Dafür scheinen sich die im Problem definierten horizontalen oder vertikalen Slabs anzubieten. Sei nun also O_1 die Teilmenge der vertikalen und O_2 die Teilmenge der horizontalen Slabs von O. (Abb. 3.1)

Wird ein Punkt von keinem Slab überdeckt, gibt es keine zulässige Lösung und das Verfahren wird abgebrochen.

Nun betrachet man das Verfahren beispielhaft auf O_1. O_2 funktioniert analog.
Für ein Element o aus O_1 gilt nach 3.1, dass es der Form $[a_i, b_i] \times [-\infty, \infty]$ ist, wobei a_i und b_i aus $\mathbb{R}$ sind.
Nun weiß man, dass ein Punkt (x, y) genau dann von o überdeckt wird, wenn $x \in [a_i, b_i]$ gilt, also x als Punkt auf der x-Achse durch das Intervall $[a_i, b_i]$ überdeckt wird.

Wird ein Punkt von keinem der vertikalen Slabs überdeckt, wird er beim Teilproblem der Horizontalen betrachtet und an dieser Stelle ignoriert. (Abb.3.2)

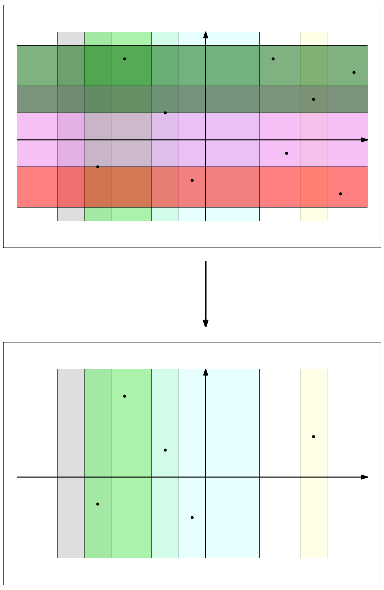

Abb. 3.1: Problem mit vertikalen und horizontalen Slabs zu einem Problem mit nur Vertikalen vereinfachen.

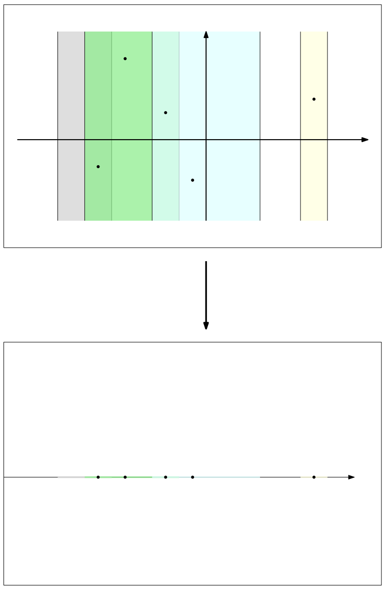

Abb. 3.2: Teilproblem auf Intervall-Punkt-Set Cover reduzieren.

Es reicht also an dieser Stelle, das Teilproblem über das Intervall-Punkt-Cover-Problem P_1 zu lösen, und danach die Slabs in die Teillösung zu nehmen, die den Intervallen entsprechen, die in der Lösung des Intervall-Punkt-Cover-Problem sind. 3.3

Formuliert man nun analog das Teilproblem P_2 und löst dieses ebenfalls, müssen nur noch die Teillösungen zu einer zusammengesetzt werden.

In diese Gesamtlösung werden nun alle Slabs genommen, die in einer der Teillösungen verwendet werden.

3.3 Korrektheit und Schranke

Das dieses Verfahren aber eine zulässige Lösung liefert, ist noch zu zeigen.
Interessant ist auch die Frage danach, wie gut die dabei erreichte Lösung ist.

Tatsächlich handelt es sich um einen konstante 2-LP-Approximationsgüte.

Satz 3.12. *Das Cover-Slabs Problem 3.1 hat einen 2-LP-Approximationsalgorithmus.*

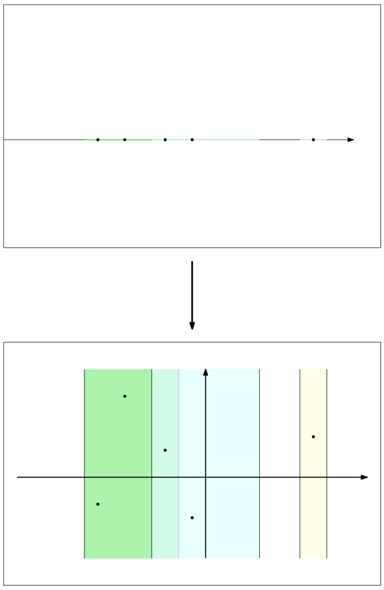

Abb. 3.3: Löse das Teilproblem und übertrage die Lösung aus Slabs.

Beweis. Zuerst gilt es zu zeigen, dass das Ergebnis eine zulässige Lösung ist, wenn das Problem eine zulässige Lösung besitzt:
Wenn es eine zulässige Lösung gibt, so wird jeder Punkt von einem vertikalen oder horizontalen Slab überdeckt.

Entsprechend wird der Punkt in einem der Teilprobleme beachtet und überdeckt. Wenn in der Gesamtlösung dann alle Slabs verwendet werden, die in einem der Teillösungen vorkommen, so wird der Punkt auf jeden Fall überdeckt.
Es handelt sich also um eine zulässige Lösung.

Um zu zeigen, dass es eine 2-LP-Approximationsgüte ergibt, wird das Lemma 3.3 genutzt.
Die Unterteilung in O_1 und O_2 erfüllt für jede Partition der Punkte in U_1 und U_2, dass wenn U_1 von O_1 und U_2 von O_2 überdeckt wird, sie jeweils eine Instanz von P_1 und P_2 bilden, wobei P_1 und P_2 Intervall-Punkt-Set Cover Probleme sind.

Da man über P_1 und P_2 nun nach 3.11 weiß, dass sie jeweils einen 1-LP-Approximationsalgorithmus haben, gilt nach 3.3 eine (1+1)-LP-Approximationsgüte, also eine 2-LP-Approximationsgüte.

□

4 Überdecken von Punkten durch Rechtecke

4.1 Eigenschaften von COVER-RECHTECK

Das zweite in dieser Arbeit betrachtete Problem wird als *COVER-RECHTECK* bezeichnet.

Definition 4.1. Die überdeckende Menge O besteht aus achsenparallelen Rechtecken im $\mathbb{R}^2$, deren Ränder sich genau viermal oder gar nicht schneiden dürfen.
Die Grundmenge U besteht aus Punkten im $\mathbb{R}^2$.
Das Problem wird hier als ungewichtet betrachtet, das heißt alle Rechtecke haben dasselbe Gewicht.
Da man den ungewichteten Fall betrachtet, kann man ohne Beschränkung der Allgemeinheit davon ausgehen, dass kein Rechteck Teilmenge eines anderen Rechtecks ist. Die Scheidungsbedingung ist folglich alternativ formulierbar, als kein Eckpunkt eines Rechtecks darf in einem anderen liegen[CG14].

Für dieses Problem ist ebenfalls bekannt, dass es nicht beliebig genau angenähert werden kann.

Lemma 4.2. *Es gibt keinen PTAS für COVER-RECHTECK [CG14].*

Ein mögliches Ergebnis wäre aber noch ein konstanter Approximationsalgorithmus.

Definition 4.3. Eine *Halbordnung* ist eine Relation $\leq$ auf einer Menge M, die reflexiv, antisymmetrisch und transitiv ist. Das bedeutet, dass folgende Eigenschaften gelten:

$$\forall x \in M : x \leq x$$

$$\forall x, y \in M : (x \leq y) \wedge (y \leq x) \Rightarrow x = y$$

$$\forall x, y, z \in M : (x \leq y) \wedge (y \leq z) \Rightarrow x \leq z$$

Bei dieser Problemstellung bieten sich zwei analog funktionierende Möglichkeiten an, eine Halbordnung zu definieren. Hierzu sagt man, dass für zwei Rechtecke a und b gilt:

$a \leq b$, falls die Projektion von a auf die x-Achse eine Teilmenge der Projektion von b ist. Analog funktioniert das auch mit einer Projektion auf die y-Achse. So arbeitet auch ein Beweis, indem er eine solche Halbordnung definiert, zeigt das bestimmte Rechtecke damit geordnet sind und damit dann ähnliche Beweisideen verwendet, wie im Folgenden[PT10].

Lemma 4.4. *Für eine Probleminstanz von COVER-RECHTECK mit vier Rechtecken, kann eine nicht-ganzzahlige Extrempunktlösung nicht für alle Koordinaten $\neq 0$ erfüllen, dass sie $= \frac{1}{3}$ sind.*

Beweis. Um zu zeigen, dass so ein Extrempunkt nicht existieren kann, wird angenommen, dass es eine solche Lösung x gibt, dass für alle sein Koordinaten $\neq 0$ erfüllt, dass sie $= \frac{1}{3}$ sind.
Es ist vom Rank Lemma 2.4 bekannt, dass es eine Teilmatrix $A' \in \mathbb{R}^{4\times 4}$ von A gibt, die genau an drei Stellen pro Zeile eine eins hat und deren Determinante $\neq 0$ ist.
Sie hat dann ohne Beschränkung der Allgemeinheit nach eventuellem Vertauschen von Zeilen und Spalten die Form:

$$A' = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Dann wird der Punkt, der durch die 1. Zeile repräsentiert wird, überdeckt von den Vierecken A, B und C, die per x-Achsen-Projektion-Halbordnung ohne Beschränkung der Allgemeinheit geordnet sind: $A \leq B \leq C$, da die Rechtecke alle denselben Punkt überdecken und sich entsprechend schneiden müssen.

1. Fall: Sei A das Viereck, dass den 2. Punkt nicht überdeckt. Dann gilt für das letzte Viereck D, dass es entweder links oder rechts von A liegt, wie in Abbildung 4.1. Will man nun den 3. Punkt einfügen, so kann dieser Punkt nicht in A und D gleichzeitig liegen.

2. Fall: Sei B das Viereck, dass den 2. Punkt nicht überdeckt. Nun wird aber der gesamte Schnittbereich von A und C von B überdeckt. Es kann also garnicht ein Punkt darin liegen, der nicht auch von B überdeckt wird.

3. Fall: Sei C das Viereck, dass den 2. Punkt nicht überdeckt. Nun kann man analog zum 1. Fall das letzte Viereck D entweder oberhalb oder unterhalb von C einfügen. Will man nun den 3. Punkt einfügen, kann der nicht gleichzeitig in C und D liegen.

$\square$

Dieses Ergebnis lässt sich erst einmal innerhalb des kleinen Beispiels verallgemeinern.

Lemma 4.5. *Für eine Probleminstanz von COVER-RECHTECK mit vier Rechtecken, kann eine nicht-ganzzahlige Extrempunktlösung nicht für alle Koordinaten $\neq 0$ erfüllen, dass sie $\leq \frac{1}{3}$ sind.*

Beweis. Um zu zeigen, dass so ein Extrempunkt nicht existieren kann, wird angenommen, dass es eine solche Lösung x gibt, dass für alle sein Koordinaten $\neq 0$ erfüllt, dass sie $\leq \frac{1}{3}$ sind.

Für den Fall, dass alle $= \frac{1}{3}$ sind, betrachte den Widerspruch über 4.4.
Sei nun also der erste Punkt von allen vier Vierecken überdeckt, wie in Abbildung 4.2, damit die Koordinaten $\leq \frac{1}{3}$ erfüllen können.
So wäre ohne Einschränkung der Allgemeinheit die Teilmatrix A' der Form:

$$A' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Bei zwei Punkten, die von vier Vierecken überdeckt würden, würde keine lineare Unabhängigkeit mehr gelten.

Nun wäre aber die einzige Lösung, da die Determinante ungleich null ist und somit die Abbildung bijektiv ist, der Vektor

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Das widerspricht der Annahme.

□

Ein alternativer Beweis, der strukturell mehr über einen möglichen Beweis ahnen lässt, ist der folgende:

Beweis. Um zu zeigen, dass so ein Extrempunkt nicht existieren kann, wird angenommen, dass es eine solche Lösung x gibt, dass für alle sein Koordinaten $\neq 0$ erfüllt, dass sie $= \frac{1}{3}$ sind.

Wie zuvor findet man über das Rank Lemma folgende Matrix:

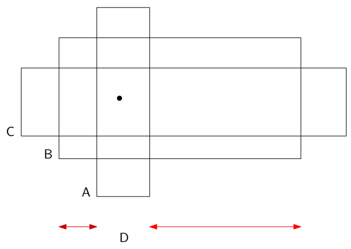

Abb. 4.1: Der 1. Fall graphisch dargestellt.

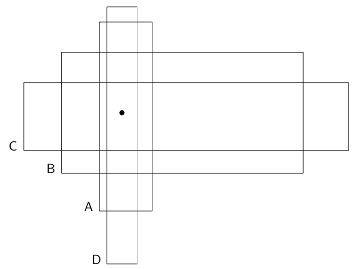

Abb. 4.2: Die vier Vierecke überdecken alle einen Punkt.

$$A' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

Sei nun also der erste Punkt von allen vier Vierecken überdeckt, wie in Abbildung 4.2. Nun gilt für die restlichen drei Punkte, dass sie von jeweils drei Vierecken überdeckt werden.

Hierbei gibt es die Möglichkeit von A ,B ,C von B, C, D von A, C, D oder von A, B, D jeweils überdeckt zu werden. Da aber A, B, C, D alle den gleichen Punkt überdecken, gilt ohne Beschränkung der Allgemeinheit $D \leq A \leq B \leq C$.

Dann kann aber ein Punkt nicht genau von B, C, D oder von genau A, C, D überdeckt werden, da der Schnittbereich immer in einem anderen Viereck enthalten ist.
Es gibt also nur zwei Möglichkeiten statt der benötigten drei um Punkte zu überdecken.

Eine solche Extrempunktlösung ist folglich nicht möglich.

□

Eine Verallgemeinerung dieser Aussagen lautete wie folgt:

Vermutung 1. Für eine beliebige Probleminstanz von COVER-RECHTECK kann eine nicht-ganzzahlige Extrempunktlösung nicht für alle Koordinaten $\neq 0$ erfüllen, dass sie $= \frac{1}{3}$ sind.

Vermutung 2. Für eine beliebige Probleminstanz von COVER-RECHTECK kann eine nicht-ganzzahlige Extrempunktlösung nicht für alle Koordinaten $\neq 0$ erfüllen, dass sie $\leq \frac{1}{3}$ sind.

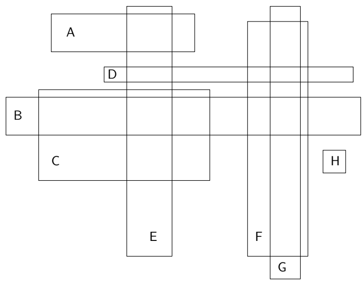

Abb. 4.3: Eine Menge an Vierecken, die die Überschneidungseigenschaft erfüllen.

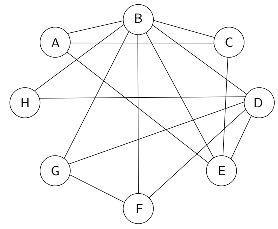

Abb. 4.4: Aus Abbildung 4.3 mit Hilfe der Halbordnung aus 4.1 ein Vergleichbarkeitsgraph.

Im Nachfolgenden sollen einige Beweisideen und Vorgehensweisen für diese Vermutungen aufgeführt werden.

Der Unterschied zwischen dem Problem 4.1 und dem allgemeinen Fall von achsenparallelen Vierecken, der nur eine $O(\log \log OPT)$ Approximationsgüte hat, ist die Überschneidungsart der Vierecke. Sie wird in [CC09] auch als no-corner Überschneidung bezeichnet und bietet Möglichkeiten für Beweise von 1. Um formal mit dieser Eigenschaft einen Beweis zu führen, muss diese strukturiert ausgenutzt werden.

Eine der strukturellen Darstellungsmethoden sind die über Halbordnung definierten Vergleichbarkeitsgraphen.

Definition 4.6. Ein *Vergleichbarkeitsgraph* ist über eine Menge M und eine Halbordnung $\leq$ auf M definiert.
Der Graph hat die Knotenmenge $V = M$ und für die Kanten E gilt:

$$\forall a \neq b \in M : \\ \text{Kante zwischen } a \text{ und } b \text{ ist in } E \Leftrightarrow (a \leq b) \vee (b \leq a)$$

Beispiel 4.1. Nimmt man die Instanz an Rechtecken 4.3, die sich entweder an genau vier Punkten oder gar nicht schneiden, und verwendet die Halbordnung der Projektion auf die x-Achse, erhält man den Vergleichbarkeitsgraphen 4.4.

Vergleichbarkeitsgraphen gehören einer Graphenklasse mit starken Eigenschaften an, die gegebenfalls eine Idee für einen Beweis liefern könnte.

Definition 4.7. Ein Graph ist *perfekt*, wenn für induzierte Subgraphen, dass heißt Teilmengen der Knoten mit allen inzidenten Kanten, gilt, dass die Cliquenzahl gleich der chromatischen Zahl ist.

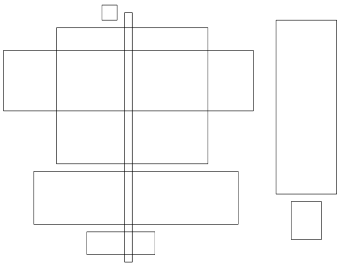

Abb. 4.5: Ein Beispiel für Rechtecke, die Fallunterscheidungen wie in 4.4 erschweren.

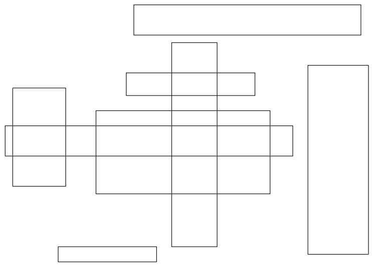

Abb. 4.6: Ein Beispiel, das verdeutlicht, inwiefern die Entscheidung, ob ein Rechteck im Vergleich zu einem Anderen oben oder unten beziehungsweise links oder rechts liegt, wie in 4.4 im Allgemeinen nicht trivial ist.

Vergleichbarkeitsgraphen sind perfekt [Gol04][S.52].

Es gilt der starke Satz über perfekte Graphen:

Satz 4.8. *Ein Graph ist genau dann perfekt, wenn kein induzierter Teilgraph ein ungerader Kreis mindestens der Länge fünf ist, oder das Komplement eines solchen Kreises [CRST06].*

Eine Beweisskizze könnte also wie folgt aussehen:
Es ist möglich, einen Vergleichbarkeitsgraphen zu einer Probleminstanz zu erstellen. Damit kann man ähnlich wie in 4.4 Fallunterscheidungen darüber treffen, welche Rechtecke geordnet sind.
Die Hoffnung wäre dann, die Existenz eines ungeraden Kreises mindestens der Länge fünf oder das Komplement eines solchen Kreises zu beweisen.
Aufgrund der zum Beispiel sich nicht schneidenden, aber immer noch geordneten Rechtecken, wie in Abbildung 4.5 ist das allerdings nicht trivial.

Eine weitere Beweisidee ist, ähnlich wie in 4.4, 4.1 und auch in Teilen des Beweises zur Aussage, dass die bestmögiche Approximationsgüte auch schon $O(\log\log OPT)$ für Rechtecke im Allgemeinen ist wie in [PT10][S.6], dass man über die Halbordnung und eine Sortierung nach x-Koordinaten vielleicht argumentieren kann, dass eines der Rechtecke bei einer Verteilung wie in 1 oder 2 gleichzeitig rechts und links, beziehungsweise oben und unten von einem anderen Rechteck liegen müsste.
Es gilt also zu zeigen, dass solche Strukturen, wie im 4×4-Fall dargestellt, immer als

Unterstrukturen existieren.
Das ist im Allgemeinen nicht einfach, da zum Beispiel teilweise Überlagerungen und Ähnliches wie in Abbildung 4.6 ein klares Entscheiden, was links und rechts beziehungsweise oben und unten von einem Rechteck genau bedeuten, erschweren.

Allerdings sei an dieser Stelle nochmal betont, wie komplex verwandte Probleme sind. So wird die Approximationsgüte für das Rechteckproblem in [PT10] dadurch bewiesen, dass im LP-Dualen Problem ein Spezialfall dieser Überschneidungsart zu finden ist, die scharf die Schranke erfüllen.
Daraus wird dann ebenfalls gefolgert, dass ein ähnlicher Überschneidungsfall im vierdimensionalen Set Cover auch keinen konstanten Approximationsalgorithmus hat [PT10]. Es ist also durchaus nicht trivial, ob es ein solches Ergebnis wie in 2 und damit ein konstanter Approximationsalgorithmus existiert.

4.2 Ein iterativer Ansatz

Für einen interativen Ansatz wird nun eine Standardstrategie des Aufrundens gewählt. Ein möglicher Algorithmus könnte wie folgt aussehen:

Im ersten Schritt wird eine relaxierte LP-Lösung für das Problem berechnet, zum Beispiel mit der Innerer-Punkt-Methode 2.13. Nun hat man eine eventuell nicht-ganzzahlige optimale Extrempunktlösung.

Im zweiten Schritt werden alle Rechtecke entfernt, deren korrespondierende Koordinate $= 0$ ist.

Im dritten Schritt wählt man eine der größten Koordinaten, die $\geq \frac{1}{3}$ ist und rundet diese auf.

Im vierten Schritt entfernt man alle Punkte, die von dem Rechteck, das diese Koordinate repräsentiert, überdeckt werden, sowie das Rechteck selbst aus der Problemstellung.

Jetzt beginnt man wieder beim ersten Schritt, bis alle Punkten überdeckt sind und alle Koordinaten 0 oder 1 sind.

4.3 Ausblick

Dieser iterative Ansatz liefert eine konstante Approximation in polynomialer Laufzeit, wenn die vorherigen Vermutungen gelten, denn:

Vermutung 3. Wenn die vorherigen Vermutung gelten würden, gäbe es eine 3-LP-Approximation für COVER-RECHTECKE.

Beweis. Es gelte 2.
Sei OPT die erste relaxierte optimale Extrempunktlösung, die im Ansatz bestimmt wird.

In jedem Durchlauf wird für eine Koordinate der Wert ≤ 3 mal genommen im Vergleich zu einer optimalen Lösung.
Danach wird diese Koordinate nicht mehr betrachtet.

Die Restlösung ist eine zulässige Lösung für das Restproblem, ohne die Rechtecke mit Koordinate $= 0$, das Rechteck, für das gerade aufgerundet wurde und die Punkte die bereits von diesem Rechteck überdeckt werden.

Für dieses Restproblem wird dann eine relaxierte optimale Extrempunktlösung bestimmt. Hierbei lässt sich feststellen, dass die Werte in den Koordinaten höchstens kleiner werden, da nur Rechtecke entnommen wurden, die für eine Restlösung nicht mehr in Frage kommen und Punkte aus dem Problem genommen wurden, die mehr Bedingungen liefern würde und damit Koordinaten höchstens vergrößert hätten.

Wird im nächsten Schritt wieder eine Koordinate aufgerundet, so wird ihr Wert ≤ 3 mal genommen, im Vergleich zur optimalen Teillösung und damit, da diese höchstens kleiner ist als die Zuvorige induktiv auch im Vergleich zur Koordinate in OPT.

Damit kann man im Ergebnis des Approximationsalgorithmus zeigen, dass die Koordinate ≤ 3 mal der Koordinate in OPT gilt.

Somit ist die Approximationsgüte gleich drei und es gibt eine 3-LP-Approximation für COVER-RECHTECKE.

□

Die Polynomialzeit ergibt sich über die polynomielle Laufzeit der Inneren-Punkt-Methode 2.13 und dadurch, dass in jedem Schritt mindestens eine Koordinate aufgerundet werden würde, wenn 2 gilt. Dann käme nur ein linearer Faktor dazu.

Das ist im Vergleich zu sonstigen Ergebnissen eine sehr genauer Approximationsalgorithmus. Es wäre von Interesse die Vermutungen 1 und 2 zu beweisen, zum Beispiel mit den bereits genannten Beweisideen, wie dem Vergleichbarkeitsgraphen oder dem Suchen nach widersprüchlichen Unterstrukturen.

5 Zusammenfassung

Die Arbeit hatte sich mit der Frage auseinandergesetzt, ob man mit Hilfe von Linearer Programmierung Approximationalgorithmen für 4.1 und 3.1 bekommen kann.
Zudem wurden verschiedene Beweisstrategien versucht und recherchiert.

Das Ergebnis führt für 3.1 zu einem 2-LP-Approximationsalgorithmus und für 4.1 zu zumindest der Vermutung einers 3-LP-Approximationsalgorithmus.
Interessant sind vorallem weiterführende Betrachtungen und Ideen, die in dieser Arbeit angesprochen wurden. Das Verfahren, das zur Lösung von COVER-SLABS genutzt wurde, kann abgewandelt auch auf ähnliche Probleme angewendet werden.
Die Eigenschaften hinter COVER-RECHTECKE zeigt Möglichkeiten auf, einen Approximationsalgorithmus aus Strukturen abzuleiten und nutzt gleichzeitig ein grundlegendes Prinzip zur Folgerung der Schranke.

Aus Anwendungssicht sind die hier erzielten Ergebnisse nicht direkt anwendbar, allerdings besitzen ähnliche Probleme über lineare Programmierung eventuell ähnliche Approximationsalgorithmen und dafür zeigt diese Arbeit Möglichkeiten auf.

Im Ausblick bleibt nun die Frage, nach COVER-RECHTECKE. Aus wissenschaftlicher Sicht ist die Frage nach einem konstanten Approximationsalgorithmus auch über diese Arbeit hinaus interessant, da die duale Version des Problems keinen konstanten Approximationsalgorithmus hat.
Es gibt die Möglichkeit, hier einen ganz anderes Verfahren, das auf Linearer Programmierung oder etwas anderem basiert, zu nutzen und damit gegebenfalls bessere Ergebnisse zu erzielen.

Ob es noch ein bessere Approximationsalgorithmen zu COVER-SLABS gibt, ist eine offene Frage. Der Algorithmus liefert, wie gezeigt, einen scharfen 2-LP-Approximationsalgorithmus, für das gesamte Problem muss diese Schranke allerdings nicht scharf sein.

Zusammenfassend kann man sagen, es wurde ein guter Approximationsalgorithmus für COVER-SLABS gefunden und einige versprechende Ansätze zu COVER-RECHTECKE, die es weiter zu verfolgen gilt.

Literaturverzeichnis

[AES10] B. Aronov, E. Ezra und M. Sharir: Small-Size eps-Nets for Axis-Parallel Rectangles and Boxes. *SIAM Journal on Computing*, 39(7):3248–3282, 2010. https://doi.org/10.1137/090762968.

[BDGMS12] Shalev Ben-David, Elyot Grant, Will Ma und Malcolm Sharpe: The approximability and integrality gap of interval stabbing and independence problems. *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012*, Januar 2012.

[BP14] N. Bansal und K. Pruhs: The Geometry of Scheduling. *SIAM Journal on Computing*, 43(5):1684–1698, 2014. https://doi.org/10.1137/130911317.

[CC09] Parinya Chalermsook und Julia Chuzhoy: Maximum independent set of rectangles. *SODA*, 2009.

[CG14] Timothy M. Chan und Elyot Grant: Exact algorithms and APX-hardness results for geometric packing and covering problems. *Comput. Geom.*, 47:112–124, 2014.

[CGKS12] Timothy M. Chan, Elyot Grant, Jochen Könemann und Malcolm Sharpe: Weighted Capacitated, Priority, and Geometric Set Cover via Improved Quasi-uniform Sampling. SODA '12, Seiten 1576–1585, Philadelphia, PA, USA, 2012. Society for Industrial and Applied Mathematics. http://dl.acm.org/citation.cfm?id=2095116.2095241.

[CRST06] Maria Chudnovsky, Neil Robertson, Paul Seymour und Robin Thomas: The strong perfect graph theorem. *Annals of mathematics*, Seiten 51–229, 2006.

[Fei98] Uriel Feige: A Threshold of Ln N for Approximating Set Cover. *J. ACM*, 45(4):634–652, Juli 1998, ISSN 0004-5411. http://doi.acm.org/10.1145/285055.285059.

[Gol04] Martin Charles Golumbic: *Algorithmic graph theory and perfect graphs*, Band 57. Elsevier, 2004.

[HK10] Alan J Hoffman und Joseph B Kruskal: Integral boundary points of convex polyhedra. In: *50 Years of Integer Programming 1958-2008*, Seiten 49–76. Springer, 2010.

[HMRR14] Nabil H. Mustafa, Rajiv Raman und Saurabh Ray: Settling the APX-Hardness Status for Geometric Set Cover. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, Dezember 2014.

[LRS11] Lap Chi Lau, Ramamoorthi Ravi und Mohit Singh: *Iterative methods in combinatorial optimization*, Band 46. Cambridge University Press, 2011.

[MPT+98] João Meidanis, Oscar Porto, Guilherme P Telles *et al.*: On the consecutive ones property. *Discrete Applied Mathematics*, 88(1-3):325–354, 1998.

[PT10] János Pach und Gábor Tardos: Tight lower bounds for the size of epsilon-nets. *CoRR*, abs/1012.1240, 2010. `http://arxiv.org/abs/1012.1240`.

[Ste10] Johannes Martin Stemick: Seminar zur Approximationstheorie im Wintersemester 2009/2010 Monge-Ampére-Gleichung Innere-Punkt-Methoden. 2010.

[TMC18] Krzysztof Fleszar Joachim Spoerhase Alexander Wolf Timothy M. Chan, Thomas C. van Dijk: Stabbing Rectangles by Line Segments – How Decomposition Reduces the Shallow-Cell Complexity. *26th Annual European Symposium on Algorithms (ESA 2018)*, Seiten 1–22, 2018.

[Tru78] Klaus Truemper: Algebraic characterizations of unimodular matrices. *SIAM Journal on Applied Mathematics*, 35(2):328–332, 1978.

Erklärung

Hiermit versichere ich die vorliegende Abschlussarbeit selbstständig verfasst zu haben, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, und die Arbeit bisher oder gleichzeitig keiner anderen Prüfungsbehörde unter Erlangung eines akademischen Grades vorgelegt zu haben.

Würzburg, den 6. Juli 2018

....
Maike Rösch