Completeness for Parallel Access to NP and Counting Class Separations

Holger Spakowski

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Universitätsstraße 1
40225 Düsseldorf
spakowsk@cs.uni-duesseldorf.de

Abstract: Die Dissertation beschäftigt sich mit Problemen, die vollständig für die Komplexitätsklasse $\mathrm{P}_{\|}^{\mathrm{NP}}$ sind, sowie mit der Separation von Zählklassen. $\mathrm{P}_{\|}^{\mathrm{NP}}$ ist die Klasse der Probleme, die sich effizient mit parallelem Zugriff auf NP lösen lassen.

Wir untersuchen die Komplexität von mit Wahlsystemen assoziierten Problemen. Wahlsysteme sind Vorschriften, nach denen aus einer Kandidatenmenge die Gewinner einer Abstimmung bestimmt werden können. Wir beweisen, daß das Gewinner-Problem für die Wahlsysteme von Kemeny und Young beide vollständig für die Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$ sind. Weiterhin betrachten wir zwei prominente Heuristiken für die Approximation des NP-vollständigen Problems der minimalen Knotenüberdeckung. Wir weisen nach, daß gewisse Entscheidungsprobleme, die mit der Qualität der Approximation durch diese Heuristiken in Zusammenhang stehen, vollständig für $\mathrm{P}_{\|}^{\mathrm{NP}}$ sind.

Der letzte Teil der Dissertation beantwortet Fragen, die in der einflußreichen Arbeit von Fenner, Fortnow und Kurtz im Jahre 1994 aufgeworfen wurden: Wir zeigen, daß die Zählklassen LWPP und WPP nicht uniform gap-definierbar sind. Desweiteren konstruieren wir ein Orakel, relativ zu dem WPP nicht abgeschlossen unter polynomialzeitbeschränkter Turing-Reduzierbarkeit ist. Dies hat zur Folge, daß ein Beweis für die Gleichheit der ähnlich definierten Klassen LWPP und WPP nichtrelativierbar sein muß. Wir erhalten diese Resultate durch Anwendung einer bekannten Technik, bei der Orakel-Turingmaschinen in multilineare Polynome mit kleinem Grad kodiert werden. Wir beweisen dazu eine neue kombinatorische Eigenschaft solcher Polynome.

1 Einleitung: Die Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$

Eines der zentralen Ziele der Komplexitätstheorie ist es, den Ressourcenbedarf (üblicherweise Zeit oder Speicherplatz) zum Lösen von wichtigen Berechnungsproblemen zu bestimmen. Wir würden insbesondere gerne in der Lage sein, effizient lösbare von nicht effizient lösbaren Problemen zu unterscheiden. Ein Entscheidungsproblem wird gewöhnlich als effizient lösbar angesehen, wenn es eine *deterministische* Turingmaschine gibt, die dieses Problem in polynomialer Zeit, gemessen an der Eingabegröße, lösen kann. Die Klasse dieser Probleme wird mit P bezeichnet.

Eine andere fundamentale Komplexitätsklasse ist die Klasse NP. Die Klasse NP enthält alle Entscheidungsprobleme, die von einer *nichtdeterministischen* Turingmaschine in polynomialer Zeit akzeptiert werden können. Eine äquivalente Definition für NP ist: NP ist die Klasse aller Probleme, deren "ja"-Instanzen sich in polynomialer Zeit *verifizieren* lassen.

Wir betrachten als Beispiel das Entscheidungsproblem `Vertex Cover`. Sei G ein beliebiger ungerichteter Graph. Eine *Knotenüberdeckung* (*vertex cover*) von G ist eine Knotenmenge V' mit der Eigenschaft, daß jede Kante in G wenigstens einen Knoten aus V' enthält. Mit $\tau(G)$ bezeichnen wir die Anzahl der Knoten einer kleinsten Knotenüberdeckung von G. Das Entscheidungsproblem `Vertex Cover` ist nun folgender-

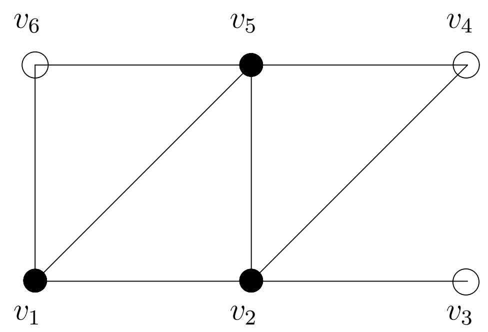

Abbildung 1: Graph G mit einer kleinsten Knotenüberdeckung $\{v_1, v_2, v_5\}$ der Größe 3

maßen definiert:
Instanz: Ein ungerichteter Graph G und eine Zahl $k \in \mathbb{N}$.
Frage: Gilt $\tau(G) \leq k$?

Dieses Problem hat die folgende entscheidende Eigenschaft: Falls die Antwort auf die Frage "ja" ist, dann existiert ein Beweis für die Korrektheit der "ja"-Antwort, der in polynomialer Zeit verifiziert werden kann. Ein solcher Beweis besteht hier einfach aus einer Knotenüberdeckung von G mit $\leq k$ Knoten. Damit ist gezeigt, daß `Vertex Cover` in NP liegt.

Offensichtlich gilt P $\subseteq$ NP. Aber es ist nicht bekannt, ob diese Inklusion echt ist, d.h. wir wissen nicht, ob es NP-Probleme gibt, die nicht in polynomialer Zeit mit einem (deterministischen) Algorithmus lösbar sind. Wir haben es hier mit einem der wichtigsten Probleme der theoretischen Informatik und der Mathematik zu tun. Die Bedeutung dieses P versus NP Problems rührt zum großen Teil daher, daß es hunderte praktisch und theoretisch interessante Probleme gibt (z.B. das Problem des Handelsreisenden, Scheduling-Probleme, das Erfüllbarkeitsproblem für boolesche Formeln, Graphfärbbarkeitsprobleme), die zwar in NP liegen, für die aber bisher noch kein Polynomialzeitalgorithmus gefunden worden ist. Man vermutet allgemein, daß P $\neq$ NP gilt. Ein Beweis dieser Vermutung scheint jedoch mit den aktuell verfügbaren mathematischen Methoden nicht möglich zu sein.

Diese Tatsache motivierte die Suche nach den schwierigsten Problemen in NP, die so genannten NP-vollständigen Probleme. Für die Definition der NP-Vollständigkeit benötigen wir das Konzept der komplexitätsbeschränkten Reduktion, welche eine wichtige Methode zum Vergleich der Schwierigkeit von Problemen darstellt. Eine Reduktion ist eine Umformung eines Problems in ein anderes Problem. Es gibt verschiedene Arten von Reduktionen. Die wichtigste ist die polynomialzeitbeschränkte many-one-Reduktion. [1] Ein Problem A ist auf ein Problem B polynomialzeit many-one reduzierbar (formal: $A \leq_{\mathrm{m}}^{\mathrm{p}} B$), wenn eine polynomialzeitberechenbare Funktion f existiert mit $x \in A$ genau dann wenn $f(x) \in B$. Falls $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ gilt, dann ist A nicht (viel) schwerer als Problem B, denn jeder Algorithmus für B kann (zusammen mit Funktion f) dazu verwendet werden, das Problem A zu lösen. Insbesondere folgt aus $B \in \mathrm{P}$ stets $A \in \mathrm{P}$. Oder äquivalent: $A \notin \mathrm{P}$ impliziert $B \notin \mathrm{P}$.

Ein Problem A ist NP-vollständig, falls folgendes gilt:

1. A ist NP-hart[2], d.h. jedes Problem in NP ist auf A polynomialzeit many-one reduzierbar, und
2. $A \in \mathrm{NP}$.

Der Nachweis der NP-Vollständigkeit eines NP-Problems A liefert ein starkes Indiz dafür, daß es für A keinen Polynomialzeitalgorithmus gibt: Die Existenz eines solchen Algorithmus würde nach sich ziehen, daß *jedes* NP-Problem in polynomialer Zeit lösbar ist, was $\mathrm{P} = \mathrm{NP}$ bedeuten würde.

Die Theorie der NP-Vollständigkeit wurde durch die Arbeiten von Cook, Karp und Levin initiiert. Cook [Coo71] bewies, daß das Erfüllbarkeitsproblem für boolesche Formeln (SAT) NP-vollständig ist. Durch Reduktion von diesem Erfüllbarkeitsproblem bewies er, daß das Problem *Subgraph Isomorphism* ebenfalls NP-vollständig ist. Aufbauend auf Cooks Resultat zeigte Karp [Kar72] die NP-Vollständigkeit für 20 weitere natürliche Probleme (darunter zum Beispiel das oben erwähnte Problem *vertex cover*). Hiermit demonstrierte er, daß NP-Vollständigkeit ein weit verbreitetes Phänomen ist. Wir bemerken, daß Levin [Lev73] nahezu gleichzeitig und unabhängig zu ähnlichen Resultaten gelangte. Seit den 1970er Jahren wurden hunderte weitere Probleme NP-vollständig nachgewiesen [GJ79].

Kommen wir zurück auf das oben angegebene Problem `Vertex Cover`. Wir erwähnten schon, daß dieses Problem NP-vollständig ist. Wir ändern nun die Fragestellung ein wenig:

Instanz: Ein ungerichteter Graph G.
Frage: Ist $\tau(G)$ *ungerade*?

Wir nennen dieses Entscheidungsproblem `Odd Minimum Vertex Cover`. Was ist die Komplexität dieses Problems? Man kann zeigen, daß `Odd Minimum Vertex Cover` NP-hart ist. Es ist jedoch nicht klar, ob dieses Problem auch NP-vollständig ist, d.h. ob es in NP enthalten ist. Es ist natürlich einfach, einen effizient verifizierbaren Beweis für die Behauptung $\tau(G) \leq k$ anzugeben. Dies hilft uns aber nicht unmittelbar weiter wenn wir die Parität von $\tau(G)$ wissen möchten. Alle Versuche, das Problem `Odd Minimum Vertex Cover` in NP zu plazieren scheiterten.

[1] Wenn nichts anderes gesagt ist, sind mit Reduktionen im folgenden immer polynomialzeitbeschränkte many-one-Reduktionen gemeint.

[2] Oft auch "NP-schwer" genannt.

Es gibt eine komplexitätstheoretische Erklärung für dieses Scheitern: Wagner [Wag87] zeigte, daß das Problem `Odd Minimum Vertex Cover` vollständig für $\mathrm{P}_{\|}^{\mathrm{NP}}$ ist, eine Komplexitätsklasse, die vermutlich eine echte Obermenge von NP darstellt. Die Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$ ist die Klasse aller Probleme, die sich von Polynomialzeitalgorithmen mit parallelem Zugriff auf ein NP-Orakel lösen lassen. Die Bezeichnung “paralleler Zugriff” rührt daher, daß die im Laufe der Berechnung gestellten Fragen nicht von den Antworten auf zuvor gestellte Fragen abhängen.

Der Begriff der $\mathrm{P}_{\|}^{\mathrm{NP}}$-Vollständigkeit ist analog dem der NP-Vollständigkeit. Das heißt, ein Problem A ist $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig, falls folgendes gilt:

1. A ist $\mathrm{P}_{\|}^{\mathrm{NP}}$-hart, d.h. jedes Problem in $\mathrm{P}_{\|}^{\mathrm{NP}}$ ist auf A polynomialzeit many-one reduzierbar, und
2. $A \in \mathrm{P}_{\|}^{\mathrm{NP}}$.

Die $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständigen Probleme gehören zu den schwierigsten Problemen in $\mathrm{P}_{\|}^{\mathrm{NP}}$. Ein $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständiges Problem kann zum Beispiel nicht in der kleineren Klasse NP liegen, außer es gilt $\mathrm{P}_{\|}^{\mathrm{NP}} = \mathrm{NP}$.

Wir zeigen nun mit folgendem Orakel-Algorithmus, daß das Problem `Odd Minimum Vertex Cover` tatsächlich in $\mathrm{P}_{\|}^{\mathrm{NP}}$ enthalten ist. Als Orakel nehmen wir das Problem `Vertex Cover`. Sei ein Graph G mit n Knoten gegeben. Natürlich gilt $\tau(G) < n$. Wir erstellen die folgende Liste von Orakelfragen: $\langle G, 0\rangle, \langle G, 1\rangle, \ldots, \langle G, n-1\rangle$. Wir übergeben diese Liste an das Orakel `Vertex Cover`, und das Orakel gibt die Antworten auf die Fragen zurück, also beispielsweise “nein”, “nein”, ..., “nein”, “ja”, ..., “ja”.[3] Mit dieser Antwortliste in der Hand ist es einfach, $\tau(G)$ zu bestimmen. Wir akzeptieren nun G gdw. $\tau(G)$ ungerade ist.

Bevor wir zur $\mathrm{P}_{\|}^{\mathrm{NP}}$-Härte des Problems `Odd Minimum Vertex Cover` kommen, erwähnen wir einige wichtige Eigenschaften der Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$. Die Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$ hat mehrere äquivalente Charakterisierungen (siehe [Wag90]). So stimmt $\mathrm{P}_{\|}^{\mathrm{NP}}$ zum Beispiel mit der Klasse $\mathrm{P}^{\mathrm{NP}[\log]}$ überein, die alle Probleme enthält, die mittels logarithmisch vieler (adaptiver) Turing-Fragen an ein NP-Orakel in polynomialer Zeit gelöst werden können. Aus der Definition von $\mathrm{P}_{\|}^{\mathrm{NP}}$ folgt unmittelbar, daß $\mathrm{P}_{\|}^{\mathrm{NP}}$ zwischen der ersten und zweiten Stufe der Polynomialzeithierarchie liegt, d.h. $\mathrm{NP} \cup \mathrm{coNP} \subseteq \mathrm{P}_{\|}^{\mathrm{NP}} \subseteq \mathrm{P}^{\mathrm{NP}}$.

Wie zeigt man nun, daß ein Problem $\mathrm{P}_{\|}^{\mathrm{NP}}$-hart ist? Wagner [Wag87] formulierte ein sehr nützliches Kriterium für die $\mathrm{P}_{\|}^{\mathrm{NP}}$-Härte. Dieses Kriterium wurde in einer Vielzahl von Arbeiten angewandt, siehe Überblicksartikel [HHR97b].

In Kapitel 3 der Dissertation geben wir einen alternativen Zugang zum Nachweis von $\mathrm{P}_{\|}^{\mathrm{NP}}$-Härte an. Ausgangspunkt ist eine Charakterisierung der Klasse P^{NP} mit Hilfe nichtdeterministischer Turingmaschinen mit einem speziellen Akzeptierungstyp, der auf eine Arbeit von Krentel [Kre88] zurückgeht. Das Akzeptierungsverhalten von Turing-

[3] Es ist klar, daß in diesem Fall niemals ein “nein” nach einem “ja” kommt.

maschinen läßt sich leicht durch boolesche Formeln beschreiben. Wir erhalten $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständige Erfüllbarkeitsprobleme, in Analogie zu dem klassischen NP-vollständigen Erfüllbarkeitsproblem 3SAT.

Wir beweisen, daß die Erfüllbarkeitsprobleme `MAX TRUE 3SAT COMPARE`, `MAX TRUE 3SAT EQUALITY` und `ODD MAX TRUE 3SAT` vollständig in $\mathrm{P}_{\|}^{\mathrm{NP}}$ sind. Für jede boolesche Formel F in 3-CNF-Form bezeichne $\max\text{-}\mathbb{1}(F)$ die maximale Anzahl von zu `true` gesetzten Variablen, die wir in einer erfüllenden Belegung von F finden können. Dann ist `MAX TRUE 3SAT COMPARE` folgendermaßen definiert.

`MAX TRUE 3SAT COMPARE`
Instanz: 3-CNF-Formeln F_1 und F_2.
Frage: Gilt $\max\text{-}\mathbb{1}(F_1) \leq \max\text{-}\mathbb{1}(F_2)$?

Das Problem `MAX TRUE 3SAT EQUALITY` ist analog definiert mit der Frage "$\max\text{-}\mathbb{1}(F_1) = \max\text{-}\mathbb{1}(F_2)$?". Beim Problem `ODD MAX TRUE 3SAT` haben wir als Eingabe eine 3-CNF-Formel F, und die Frage ist, ob $\max\text{-}\mathbb{1}(F)$ ungerade ist.

Wie in der Theorie der NP-Vollständigkeit eignen sich diese Probleme als Startpunkte für Reduktionen auf weitere Probleme. Wir gelangen z.B. zu einem alternativen Beweis für das Resultat von Wagner, daß das Problem `Minimum Vertex Cover Compare` $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig ist. Dieses Entscheidungsproblem nutzen wir in der Arbeit für den Nachweis der $\mathrm{P}_{\|}^{\mathrm{NP}}$-Vollständigkeit von weiteren Problemen.

Die verwendete Methode ist auch auf andere Komplexitätsklassen anwendbar. Wir zeigen von einigen verwandten Problemen, daß sie vollständig in der Komplexitätsklasse P^{NP} sind.

2 Komplexität von Wahlsystemen

In den Kapiteln 4 und 5 untersuchen wir die Komplexität von mit Wahlsystemen assoziierten Problemen. Wahlsysteme sind Vorschriften nach denen aus einer Kandidatenmenge die Gewinner einer Abstimmung bestimmt werden können. Wir untersuchen Wahlsysteme, bei denen die Wähler die Kandidaten (oder allgemeiner: Alternativen) nach Präferenz ordnen können. Welche Kandidaten als Gewinner einer solchen Abstimmung festgelegt werden, hängt im allgemeinen vom verwendeten Wahlsystem ab. In der Literatur wurde eine Vielzahl von Wahlsystemen vorgeschlagen. Ein berühmtes Resultat von Arrow besagt, daß es kein "ideales" Wahlsystem gibt, d.h. es gibt kein Wahlsystem, das gleichzeitig alle Fairness-Eigenschaften erfüllt, die man von einem zufriedenstellenden Wahlsystem erwarten würde.

In der Dissertation betrachten wir die folgenden drei Wahlsysteme: Das Young-Wahlsystem, das Dodgson-Wahlsystem und das Kemeny-Wahlsystem. Dies sind sogenannte Condorcet-Wahlsysteme. Ein Wahlsystem heißt *Condorcet-Wahlsystem*, falls es das Condorcet-Prinzip respektiert [Con85]. Ein Kandidat c ist ein *Condorcet-Gewinner*, falls er jeden anderen Kandidaten mit Mehrheitsentscheid schlägt, d.h., falls im paarweisen Vergleich mit jedem anderen Kandidaten eine echte Mehrheit der Wähler c den Vorzug gibt. Ein Wahlsystem *respektiert das Condorcet-Prinzip*, falls der Condorcet-Gewinner Wahlsieger bezüglich diesem Wahlsystem ist, falls er existiert.[4] Ein Condorcet-Gewinner existiert aber im allgemeinen nicht.

Wir geben eine informale Beschreibung der in der Dissertation behandelten Wahlsysteme. Das Dodgson-Wahlsystem wurde von Charles L. Dodgson (bekannter unter seinem Pseudonym Lewis Carroll) vorgeschlagen [Dod76]. Die Gewinner werden folgendermaßen aus den Präferenzordnungen der Wähler bestimmt. Jedem Kandidaten c wird eine Zahl zugeordnet (der $\mathrm{DodgsonScore}$ von c), welche die kleinste Zahl von aufeinanderfolgenden Vertauschungen benachbarter Kandidaten in den Wählerpräferenzordnungen ist, die erforderlich ist, um c zum Condorcet-Gewinner zu machen. Dodgson-Gewinner ist jeder Kandidat mit minimalem $\mathrm{DodgsonScore}$.

Das Young-Wahlsystem [You77] basiert ebenfalls auf Veränderungen in den Wählerpräferenzordnungen. Im Gegensatz zum Dodgson-Wahlsystem erweitert das Young-Wahlsystem das Condorcet-Prinzip dadurch, daß für jeden Kandidaten c die kleinste Zahl von Wählerpräferenzordnungen (der $\mathrm{YoungScore}$ von c) bestimmt wird, die entfernt werden müssen um c zu einem Condorcet-Gewinner zu machen. Young-Gewinner ist jeder Kandidat mit minimalem $\mathrm{YoungScore}$.

Kemeny [Kem59] schlug ein Wahlsystem vor, das auf dem Begriff des Konsensrankings beruht. Ein Konsensranking ist eine Präferenzordnung, die den Präferenzordnungen der Wähler insgesamt am nächsten liegt. Die Kemeny-Gewinner sind diejenigen Kandidaten, die an erster Stelle in einem Konsensranking stehen. Als Distanz zwischen zwei gegebenen Präferenzordnungen nimmt man hierbei die Anzahl der ungeordnenten Kandidatenpaare, deren Reihenfolge in den beiden Präferenzordnungen unterschiedlich ist.

Die Untersuchung von Wahlsystemen unter komplexitätstheoretischen Aspekten wurde von Bartholdi, Tovey und Trick [BTT89] initiiert. Sie zeigten insbesondere, daß die Gewinner-Probleme des Dodgson- und des Kemeny-Wahlsystems NP-hart sind. Sie ließen offen, ob diese Probleme auch NP-vollständig sind. Hemaspaandra, Hemaspaandra und Rothe [HHR97a] beantworteten diese Frage für das Dodgson-Wahlsystem indem sie das Resultat von Bartholdi, Tovey und Trick zu einem $\mathrm{P}_{\|}^{\mathrm{NP}}$-Vollständigkeitsresultat verbesserten, denn aus diesem Vollständigkeitsresultat folgt, daß das Gewinner-Problem für das Dodgson-Wahlsystem nicht in NP liegt, außer es gilt $\mathrm{P}_{\|}^{\mathrm{NP}} = \mathrm{NP}$. In der Dissertation werden diese Untersuchungen fortgeführt. In Kapitel 4 beweisen wir, daß das Gewinner-Problem für das Young-Wahlsystem $\mathrm{P}_{\|}^{\mathrm{NP}}$-hart ist. Dazu geben wir eine Reduktion von dem Problem `Maximum Set Packing Compare` an. Die $\mathrm{P}_{\|}^{\mathrm{NP}}$-Härte von `Maximum Set Packing Compare` wiederum läßt sich leicht durch Reduktion vom oben erwähnten Problem `Minimum Vertex Cover Compare` zeigen. Außerdem untersuchen wir eine homogene Variante des Dodgson-Wahlsystems. Basierend auf ein ganzzahliges lineares Programm von Bartholdi et al. [BTT89] geben wir ein lineares Programm an, mit dessen Hilfe sich das Gewinner-Problem für diese Variante des Dodgson-Wahlsystems effizient lösen läßt.

Bartholdi et al. [BTT89] bewiesen, daß das Gewinner-Problem für das Kemeny-Wahlsystem NP-hart ist. Sie gaben dazu eine Reduktion vom klassischen NP-

[4]Condorcet-Gewinner sind eindeutig bestimmt, falls sie existieren.

vollständigen Problem `Feedback Arc Set` an. Im Kapitel 5 verbessern wir dieses NP-Härteresultat zu einem $\mathrm{P}_{\|}^{\mathrm{NP}}$-Vollständigkeitsresultat. Dazu definieren wir die Probleme `Feedback Arc Set Member` und `Vertex Cover Member` und beweisen, daß sie $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig sind durch Reduktion vom Problem `Minimum Vertex Cover Compare`. Wir modifizieren die von Bartholdi et al. angegebene Reduktion, so daß sie eine Reduktion von `Feedback Arc Set Member` auf `Kemeny Winner` wird.

Es stellt sich nun die Frage, warum diese Wahlsysteme ausgerechnet vollständig für die Klasse $\mathrm{P}_{\|}^{\mathrm{NP}}$ sind. Die oben betrachteten Wahlsysteme haben gemein, daß sie jedem Kandidaten einen Score zuordnen und die Kandidaten mit kleinstem Score gewinnen. Die Berechnung der Scores stellt in allen drei Fällen ein NP-hartes kombinatorisches Optimierungsproblem dar. Wir wissen, daß "Vergleichsversionen" von NP-harten Optimierungsproblemen (wie zum Beispiel `Minimum Vertex Cover Compare`) oft $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig sind. Dies gibt uns einen ersten Hinweis darauf, daß diese Wahlsystemprobleme $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig sein *könnten*.[5] Ob sie es wirklich sind, muß allerdings jeweils individuell überprüft werden.

3 Zwei Heuristiken für das Knotenüberdeckungsproblem

In Kapitel 6 untersuchen wir Heuristiken zur Bestimmung einer kleinsten Knotenüberdeckung eines gegebenen Graphen. Die zwei bekanntesten Heuristiken für dieses Problem sind die die *Edge-Deletion-Heuristik* und die *Maximum-Degree-Greedy-Heuristic* (siehe zum Beispiel [PS82, Pap94]). Für beide Heuristiken untersuchen wir das Problem, für einen gegebenen Graphen festzustellen, ob die minimale von dieser Heuristik gefundene Knotenüberdeckung höchstens r-mal so groß ist wie die kleinste Knotenüberdeckung. Wir beweisen, daß dieses Problem jeweils vollständig ist in $\mathrm{P}_{\|}^{\mathrm{NP}}$, falls $r \in [1, 2)$ rational ist. Wie in den vorangegangenen Kapiteln beweisen wir die $\mathrm{P}_{\|}^{\mathrm{NP}}$-Härte durch Reduktion von `Minimum Vertex Cover Compare`.

Ein analoges Problem für `Independent Set` wurde früher von Bodlaender, Thilikos und Yamazaki [BTY97] untersucht und von Hemaspaandra und Rothe [HR98] als $\mathrm{P}_{\|}^{\mathrm{NP}}$-vollständig klassifiziert (siehe auch den Überblicksartikel von E. Hemaspaandra, L. Hemaspaandra und Rothe [HHR97b]).

4 Separation von Zählklassen

In Kapitel 7 untersuchen wir Komplexitätsklassen, die sich durch Zählen der akzeptierenden und ablehnenden Berechnungspfade von nichtdeterministischen Polynomialzeitturingmaschinen (NPTM) definieren. Valiant [Val79] führte die berühmte Klasse $\#\mathrm{P}$ ein. Dies ist die Klasse aller Funktionen, die sich durch die Anzahl der akzeptierenden Pfade einer NPTM definieren lassen. Er bewies für verschiedene natürliche Probleme, daß

[5] Es ist jeweils leicht zu sehen, daß die Probleme in $\mathrm{P}_{\|}^{\mathrm{NP}}$ liegen.

sie vollständig in der Klasse $\#\mathrm{P}$ sind, darunter zum Beispiel das Problem, die Permanente einer 0-1-Matrix zu bestimmen. Fenner, Fortnow und Kurtz [FFK94] verallgemeinerten $\#\mathrm{P}$ zu GapP und entwickelten eine Theorie der gap-definierbaren Zählklassen.[6] Die Klasse GapP ist die Menge aller Funktionen, die sich definieren lassen als Differenz ("*gap*") der Anzahl der akzeptierenden und ablehnenden Pfade einer NPTM. Viele prominente Zählklassen, darunter PP, $\oplus\mathrm{P}$ und $\mathrm{C_{=}P}$, können bequem durch GapP-Funktionen charakterisiert werden. Fenner et al. definierten die neuen Komplexitätsklassen SPP, LWPP und WPP.[7] Die Klasse SPP ist das "gap-Analogon" der bekannten Klasse UP: Während UP mit Hilfe von $\#\mathrm{P}$-Funktionen definiert wird, wird SPP mit Hilfe von GapP-Funktionen definiert. Da jede $\#\mathrm{P}$-Funktion auch eine GapP-Funktion ist, gilt trivialerweise $\mathrm{UP} \subseteq \mathrm{SPP}$. Die Klasse SPP entält als wichtiges natürliches Problem das Graphisomorphie-Problem [AK02]. Arvind und Vinodchandran [AV97] und Vinodchandran [Vin04] zeigten, daß viele gruppentheoretische Berechnungsprobleme in SPP oder LWPP liegen.

Fenner, Fortnow und Kurtz [FFK94] führten den Begriff der gap-Definierbarkeit ein.

Definition 1 ([FFK94]) *Eine Klasse $\mathcal{C}$ ist* gap-definierbar*, falls es disjunkte Mengen $A, R \subseteq \Sigma^* \times \mathbb{Z}$ gibt, so daß für jedes $L \subseteq \Sigma^*$ folgendes gilt:*
$L \in \mathcal{C}$ gdw. es gibt eine NPTM *N, so daß für alle $x \in \Sigma^*$*

$$\begin{aligned} x \in L &\implies (x, \mathrm{gap}_N(x)) \in A \quad \textit{und} \\ x \notin L &\implies (x, \mathrm{gap}_N(x)) \in R. \end{aligned}$$

Hier bezeichnet $\mathrm{gap}_N(x)$ die Differenz zwischen der Anzahl der akzeptierenden Pfade und der Anzahl der ablehnenden Pfade von NPTM N bei Eingabe x. Fenner et al. [FFK94] bemerkten, daß es zwei verschiedene plausible Möglichkeiten gibt, diese Definition zu relativieren. Entsprechend schlugen sie zwei verschiedene Varianten von gap-Definierbarkeit vor: *uniforme* und *nichtuniforme* gap-Definierbarkeit. Eine relativierbare Klasse ist *uniform* gap-definierbar, falls sie in jeder relativierten Welt gap-definierbar ist, wobei die Wahl von A und R *fixiert und unabhängig vom Orakel* ist. Eine relativierbare Klasse ist *nichtuniform* gap-definierbar, falls sie in jeder relativierten Welt gap-definierbar ist, wobei die Wahl von A und R *vom Orakel abhängig* sein kann. Jede uniform gap-definierbare Klasse ist offensichtlich auch nichtuniform gap-definierbar. Beispiele für uniform gap-definierbare Zählklassen sind PP, $\mathrm{C_{=}P}$, $\oplus\mathrm{P}$ und SPP. Fenner et al. [FFK94] zeigten, daß LWPP und WPP nichtuniform gap-definierbar sind, ließen aber explizit offen, ob diese Klassen außerdem uniform gap-definierbar sind.

Fenner et al. bewiesen, daß SPP low für GapP ist. (Eine Klasse $\mathcal{D}$ ist *low* für eine Klasse $\mathcal{C}$ falls $\mathcal{C}^{\mathcal{D}} \subseteq \mathcal{C}$ gilt.) Daraus folgt unmittelbar, daß SPP low für jede *uniform* gap-definierbare Zählklasse ist.

Wir zeigen, daß es eine relativierte Welt gibt, in der $\mathrm{UP} \cap \mathrm{coUP}$ (und demzufolge auch SPP) weder für LWPP noch für WPP low ist. Demzufolge sind LWPP und WPP nicht

[6] Gupta [Gup95] definierte unabhängig dieselbe Klasse unter dem Namen $\mathcal{Z}\#\mathrm{P}$.

[7] SPP wurde unabhängig eingeführt von Gupta [Gup95] als $\mathcal{Z}\mathrm{UP}$ und von Ogiwara und Hemachandra [OH93] als XP. Die erste SPP-Machine ist implizit in einer Arbeit von Köbler, Schöning, Toda und Torán [KSTT92].

uniform gap-definierbar. Dies beantwortet die oben erwähnte Frage von Fenner, Fortnow und Kurtz. Zur Konstruktion dieser relativierten Welt machen wir Gebrauch von der wohlbekannten Technik, bei der Orakel-Turingmaschinen in multilineare Polynome mit kleinem Grad kodiert werden. Wir beweisen dazu eine neue kombinatorische Eigenschaft solcher Polynome.

Mit einer ähnlichen Beweistechnik zeigen wir, daß es ein Orakel gibt, relativ zu dem WPP nicht abgeschlossen unter polynomialzeitbeschränkter Turingreduzierbarkeit ist. Dies beantwortet eine andere Frage von Fenner et al. [FFK94]: Die ähnlich definierten Klassen LWPP und WPP sind in geeigneter Relativierung verschieden.

Literatur

[AK02] V. Arvind und P. Kurur. Graph Isomorphism is in SPP. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, Seiten 743–750, Los Alamitos, November 2002. IEEE Computer Society.

[AV97] V. Arvind und N. Vinodchandran. Solvable black-box group problems are low for PP. *Theoretical Computer Science*, 180(1–2):17–45, Juni 1997.

[Bla58] D. Black. *Theory of Committees and Elections*. Cambridge University Press, 1958.

[BTT89] J. Bartholdi III, C. Tovey und M. Trick. Voting Schemes for Which it Can Be Difficult to Tell Who Won The Election. *Social Choice and Welfare*, 6:157–165, 1989.

[BTY97] H. Bodlaender, D. Thilikos und K. Yamazaki. It is Hard to Know when Greedy is Good for Finding Independent Sets. *Information Processing Letters*, 61:101–106, 1997.

[Con85] M. J. A. N. de Caritat, Marquis de Condorcet. *Essai sur l'Application de L'Analyse à la Probabilité des Décisions Rendues à la Pluraliste des Voix*. 1785. Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, Seiten 151–158. ACM Press, 1971.

[Dod76] C. Dodgson. A Method of Taking Votes on More than Two Issues, 1876. Pamphlet printed by the Clarendon Press, Oxford, and headed "not yet published" (see the discussions in [MU95, Bla58], both of which reprint this paper).

[FFK94] S. Fenner, L. Fortnow und S. Kurtz. Gap-Definable Counting Classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[GJ79] M. Garey und D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[Gup95] S. Gupta. Closure Properties and Witness Reduction. *Journal of Computer and System Sciences*, 50(3):412–432, 1995.

[HHR97a] E. Hemaspaandra, L. Hemaspaandra und J. Rothe. Exact Analysis of Dodgson Elections: Lewis Carroll's 1876 Voting System is Complete for Parallel Access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[HHR97b] E. Hemaspaandra, L. Hemaspaandra und J. Rothe. Raising NP Lower Bounds to Parallel NP Lower Bounds. *SIGACT News*, 28(2):2–13, 1997.

[HR98] E. Hemaspaandra und J. Rothe. Recognizing When Greed Can Approximate Maximum Independent Sets is Complete for Parallel Access to NP. *Information Processing Letters*, 65(3):151–156, 1998.

[Kar72] R. Karp. Reducibilities among combinatorial problems. In R. Miller und J. Thatcher, Hrsg., *Complexity of Computer Computations*, Seiten 85–103, 1972.

[Kem59] J. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.

[Kre88] M. Krentel. The Complexity of Optimization Problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[KSTT92] J. Köbler, U. Schöning, S. Toda und J. Torán. Turing machines with Few Accepting Computations and Low Sets for PP. *Journal of Computer and System Sciences*, 44(2):272–286, 1992.

[Lev73] L. Levin. Universal search Problems. *Problems of Information Transmission*, 9:265–266, 1973.

[MU95] I. McLean und A. Urken. *Classics of Social Choice*. University of Michigan Press, 1995.

[OH93] M. Ogiwara und L. Hemachandra. A Complexity Theory for Feasible Closure Properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[PS82] C. Papadimitriou und K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[Val79] L. Valiant. The Complexity of Computing the Permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Vin04] N. Vinodchandran. Counting Complexity of Solvable Black-box Group Problems. *SIAM Journal on Computing*, 33(4):852–869, 2004.

[Wag87] K. Wagner. More Complicated Questions About Maxima and Minima, and some Closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.

[Wag90] K. Wagner. Bounded Query Classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[You77] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16:335–353, 1977.

Holger Spakowski studierte von 1994 bis 1999 Informatik an der Friedrich-Schiller-Universität Jena. Von 1999 bis 2000 war er am Institut für Mathematik und Informatik der Ernst-Moritz-Arndt-Universität Greifswald als wissenschaftlicher Mitarbeiter beschäftigt. Seit 2000 ist er wissenschaftlicher Mitarbeiter am Institut für Informatik der Heinrich-Heine-Universität Düsseldorf, wo er im Juli 2005 promovierte. Von 2002 bis 2003 war er mit einem Stipendium des DAAD "Visiting Scholar" am Department of Computer Science der University of Rochester, USA.