Widerlegung der Existenz eines Approximationsschemas

Zusammenfassung

Es gibt nicht für jedes Problem einen passenden Approximationsalgorithmus. Die Widerlegung der Existenz eines Approximationsschemas für ein NP-schweres Problem ist ein wichtiges Kapitel in der Komplexitätstheorie. In diesem Rahmen wurden einige Techniken zur Widerlegung der Existenz eines FPTAS und eines PTAS entwickelt.

Die Klasse der NP-schweren Probleme lässt sich ferner in zwei Unterklassen unterteilen. Ist das Problem NP-schwer bei binärer Kodierung, jedoch polynomiell lösbar bei unärer Kodierung der Eingabe, so heißt das Problem *gewöhnlich NP-schwer*, oder auch *pseudopolynomiell lösbar*. Ist das Problem bei beiden Kodierungen NP-schwer, so heißt es *streng NP-schwer*.

Allgemein ist das Vorgehen bei der Widerlegung der Existenz eines passenden Approximationsalgorithmus folgendes: man nimmt an, dass es einen solchen gibt und leitet daraus die Existenz eines polynomiellen Algorithmus für ein nachgewiesen NP-schweres Problem ab.

Optimierungsprobleme, deren Lösungen ganzzahlige Kosten besitzen und die Kosten der optimalen Lösung bei unärer Kodierung durch ein ganzzahliges Polynom in der Kodierungslänge beschränkt sind, heißen *gutartig*. Man kann beweisen, dass gutartige, streng NP-schwere Probleme kein FPTAS besitzen, falls P $\neq$ NP. Die dazu äquivalente Formulierung „gutartige, NP-schwere Probleme die ein FPTAS besitzen, sind polynomiell lösbar“ liefert ein Kriterium für die Existenz eines pseudopolynomiellen Algorithmus für ein NP-schweres Problem. Optimierungsprobleme, deren Lösungen ganzzahlige Kosten besitzen und die Kosten der optimalen Lösung bei *jeder* Kodierung durch ein ganzzahliges Polynom in der Kodierungslänge beschränkt ist, heißen *sehr gutartig*. Auch für sehr gutartige, NP-schwere Probleme kann die Existenz eines FPTAS widerlegt werden.

Zur Widerlegung der Existenz eines PTAS wurde in der Mitte der 1970er die „Lückentechnik“ entwickelt. Bei der Untersuchung eines NP-schweren Problems braucht man dabei eine polynomiell berechenbare Transformation seiner Instanzen auf die Instanzen eines NP-schweren Entscheidungsproblems. Die Idee dabei ist, dass sich die JA- und NEIN-Instanzen des Entscheidungsproblems in den Kosten der optimalen Lösung des zugehörigen Optimierungsproblems wiederspiegeln können. Wenn die JA-Instanzen nur Abbilder haben, deren optimale Lösung unterhalb einer festen Grenze liegt, die NEIN-Instanzen dagegen nur Abbilder haben, deren optimale Lösung oberhalb einer (höheren) Grenze liegt, so kann man mit einem genügend präzisen Approximationsalgorithmus die JA- und NEIN-Instanzen voneinander in polynomieller Zeit unterscheiden. So einen Algorithmus gibt es in jedem Approximationsschema (man muss nur die erlaubte Abweichung klein genug wählen, womit man eine lange – aber immernoch polynomielle! – Laufzeit in Kauf nimmt). Könnte man mit einem polynomiellen Algorithmus ein NP-schweres Problem lösen, dann wäre P $=$ NP. Also kann das betrachtete Optimierungsproblem kein PTAS besitzen.

Eine weitere Methode, die Existenz eines PTAS nachzuweisen, ist der Beweis der APX - Schwere eines Problems. Analog zur polynomiellen Reduktion, die die polynomielle Lösbarkeit bewahrt, gibt es approximierbarkeitsbewahrende Reduktionen. Zu diesen gehört die L-Reduktion, mit der man die Instanzen eines Problems auf die eines anderen abbilden und die dort gewonnenen Lösungen zurücktransformieren kann. Die L-Reduktion garantiert, dass die optimalen Lösungen der beiden Probleme nahe beieinander bleiben und erhöht die Abweichung der Lösung vom Optimum nur um einen multiplikativen Faktor. Die Existenz eines PTAS für das eine Problem ist somit an die des anderen gebunden. Um zu zeigen, dass ein Problem kein PTAS besitzt, genügt es also das betrachtete Problem auf ein APX-schweres Problem zu reduzieren.