Reoptimization of NP-hard Problems

Author(s):
Zych, Anna

Publication Date:
2012

Permanent Link:
https://doi.org/10.3929/ethz-a-007161496

Diss. ETH No. 20257

Reoptimization of NP-hard Problems

A dissertation submitted to

ETH Zürich

for the degree of

Doctor of Sciences

presented by

Anna Zych

Master of Science, Uniwersytet Jagielloński, Kraków

Master of Science, Vrije Universiteit, Amsterdam

born on March 7, 1980

citizen of Poland

accepted on the recommendation of

Prof. Dr. Peter Widmayer, examiner

Prof. Dr. Juraj Hromkovič, co-examiner

Prof. Dr. Giorgio Ausiello, co-examiner

2012

Abstract

This dissertation is purely and entirely dedicated to the concept of reoptimization. This concept employs a special kind of additional knowledge: under the assumption that we are given an instance of an optimization problem together with a good solution for it, we want to efficiently compute a good solution for a locally modified input instance. In this thesis, we study different reoptimization approaches and apply them to a wide selection of reoptimization problems. The contributions of the dissertation are the following.

In Chapter 4, we survey the reoptimization results available in the literature. For many of these results, we briefly sketch the techniques behind them. The problems we survey include different reoptimization variants of the Traveling Salesman problem, the Minimum Steiner Tree problem, the β-Minimum Steiner Tree problem, hereditary problems and many other [5, 1, 20, 50, 12, 26, 27, 97, 31, 72].

Studying a variety of reoptimization problems brought us to the conclusion, that in most cases the best approximation results are obtained using a certain method that we call the *self-reduction* method. In Chapter 5, we abstract this method and present it in a certain generality, independent of the concept of reoptimization. In order to obtain a good approximation algorithm for a reoptimization problem, this method needs to be combined with a greedy method, which is different for different kinds of local modifications. We classify four major types of local modifications and provide the general approximation algorithms for three of the classified types. We argue, that the fourth type is usually hopeless to reoptimize.

In Chapter 6, we apply the general algorithms to different reoptimization variants of the weighted and unweighted versions of the following problems: the Minimum Set Cover problem, the Maximum Independent Set problem, the Maximum Clique problem, the Minimum Dominating Set problem and the Minimum Vertex Cover problem. We complement these results by proving, for each of these problems, that the approximation ratios provided by our general algorithms cannot be improved, in most cases under the assumption that $P \neq NP$. This shows that our general ratios are tight in the following sense: to obtain an approximation ratio better than the general algorithms provide, one needs to explore the specific nature of the underlying optimization problem. The results in Chapter 6 were previously published in [18].

In Chapter 7, we study the reoptimization of the Shortest Common Superstring problem. The reoptimization scenario we focus on here is adding a string to the set of strings given in the input. We explain why the *self-reduction* method cannot be directly applied and briefly sketch the way to adapt it. The ratio of the parametrized approximation algorithm resulting from the adapted method can be arbitrarily close to 1.6 for string addition and 13/7 for string removal. The running time is in both cases $\mathcal{O}(m^{2k}n^{2k}(kmn+k(n^3+m)))$, where n is the number of strings, m is the sum of the lengths of the strings and k is a constant parameter. The larger k, the better the approximation ratio. The actual contribution of this chapter is the analysis of the quadratic time *OneCut* algorithm for adding a string. We prove, that it achieves an approximation ratio of 11/6. All these results can be found in [16, 17].

Finally, in Chapter 8, we study four variants of the Minimum Steiner Tree reoptimization problem: turning a terminal into a non-terminal and vice-versa, and increasing and decreasing the cost of an edge. The direct application of the general algorithms results in approximation algorithms with sub-exponential running time. We show how to parametrize the *self-reduction* method to obtain polynomial-time approximation algorithms with ratios converging to the ones guaranteed by the general algorithms. Not surprisingly, the runtime of these parametrized algorithms is exponential with respect to the parameter. Next, we show how to improve the general ratios for the reoptimization scenarios that turn a vertex into a terminal and vice-versa, and the scenario of decreasing the cost of an edge. The resulting approximation ratios are 1.22 for the modifications altering the terminal set, 1.281 for increasing the cost of an edge and 1.25 for decreasing the cost of an edge. As an alternative, we present reoptimization algorithms for the studied modifications that do not use the *self-reduction* method. They do not provide as good approximation ratios, but substantially improve the running time. A part of these results was published in [97, 15].

Zusammenfassung

Diese Dissertation widmet sich ganz und gar dem Konzept der Reoptimierung. Dieses Konzept nutzt ein besonderes, zusätzliches Wissen: unter der Annahme, dass uns eine Instanz eines Optimierungsproblems zusammen mit einer guten Lösung gegeben ist, wollen wir eine gute Lösung für eine lokal modifizierte Eingabeinstanz berechnen. In dieser Dissertation studieren wir verschiedene Ansätze der Reoptimierung und wenden diese auf eine breite Auswahl an Reoptimierungsproblemen an. Die Ergebnisse dieser Dissertation sind im Folgenden beschrieben.

In Kapitel 4 geben wir einen überblick über die in der Literatur verfügbaren Resultate zur Reoptimierung. Für viele dieser Resultate skizzieren wir kurz die dahinter stehenden Techniken. Die betrachteten Probleme beinhalten unter anderem verschiedene Varianten zur Reoptimierung des Traveling Salesman Problems, des Minimum Steiner Tree Problems, des β-Minimum Steiner Tree Problems und Problemen der Heredität [5, 1, 20, 50, 12, 26, 27, 97, 31, 72].

Das Studium verschiedener Reoptimierungsprobleme hat uns zu der Erkenntnis gebracht, dass in den meisten Fällen die besten Approximationsresultate durch Verwendung einer gewissen Methode erzielt werden, welche wir als Selbstreduktionsmethode (ßelf-reduction method") bezeichnen. In Kapitel 5 abstrahieren wir diese Methode und präsentieren sie in einer gewissen Allgemeingültigkeit, unabhängig vom Konzept der Reoptimierung. Um einen guten Approximationsalgorithmus für ein Reoptimierungsproblem zu erhalten, muss diese Methode mit einer Greedy-Methode kombiniert werden, welche von der Art der lokalen Modifikation abhängt. Wir klassifizieren lokale Modifikationen in vier Haupttypen und stellen allgemeine Approximationsalgorithmen für drei dieser Typen vor. Ferner kommen wir zu dem Schluss, dass es hoffnungslos ist, Reoptimierung auf den vierten Typ anzuwenden.

In Kapitel 6 wenden wir die allgemeinen Algorithmen auf verschiedene Reoptimierungsvarianten von gewichteten und ungewichteten Versionen der folgenden Probleme an: das Minimum Set Cover Problem, das Maximum Independent Set Problem, das Maximum Clique Problem, das Minimum Dominating Set Problem und das Minimum Vertex Cover Problem. Wir ergänzen diese Resultate durch einen Beweis, für jedes dieser Probleme, dass die durch unsere allgemeinen Approximationsalgorithmen garantierte Approximationsrate nicht verbessert werden kann, meistens unter der Annahme, dass $P \neq NP$. Dies zeigt, dass unsere allgemeinen Approximationsraten im folgenden Sinne scharf sind: Um eine bessere als die durch die allgemeinen Algorithmen garantierte Approximationsrate zu erhalten, muss man die Eigenheiten des unterliegenden Optimierungsproblems ausnutzen. Die Ergebnisse in Kapitel 6 wurden bereits in [18] veröffentlicht.

In Kapitel 7 studieren wir die Reoptimisierung des Shortest Common Superstring Problems. Wir betrachten das Reoptimisierungsszenario in welchen wir einen String zu den im Input gegebenen Strings hinzufügen. Wir erläutern, weshalb die Selbstreduktionsmethode nicht direkt angewand werden kann und skizzieren kurz, wie sie anzupassen ist. Die Rate des parametrisierten Approximationsalgorithmus, welcher aus der Anpassung der Methode hervorgeht, kann beliebig nah an 1.6 für das Hinzufügen von Strings und 13/7 für das Entfernen von Strings sein.

Die Laufzeit ist in beiden Fällen $\mathcal{O}(m^{2k}n^{2k}(kmn+k(n^3+m)))$, wobei n die Anzahl der Strings, m die Summe der Längen der Strings und k einen konstanten Parater bezeichet. Je grösser k, desto besser ist die Approximationsrate. Der eigentliche Beitrag dieses Kapitels ist die Analyse des linearen OneCut Algorithmus für das Hinzufügen eines Strings. Wir beweisen, dass es eine Approximationsrate von 11/6 erreicht. All diese Resultate finden sich in [16, 17].

Schliesslich studieren wir in Kapitel 8 vier Varianten des Minimum Steiner Tree Reoptimisierungsproblems: Hier kann ein Terminal in ein Nicht-Terminal gewandelt werden, und umgekehrt, sowie die Kosten einer Kante herauf- oder herabgesetzt werden. Die direkte Anwendung der allgemeinen Algorithmen liefert Approximationsalgorithmen mit subexponentieller Laufzeit. Wir zeigen, wie man die Methode der Selbstreduktion parametrisieren kann, um polynomielle Approximationsalgorithmen zu erhalten, deren Approximationsraten zu denen konvergieren, welche von den allgemeinen Algorithmen garantiert werden. Es überrascht nicht, dass die Laufzeit dieser parametrisierten Algorithmen exponentiell im Parameter ist. Als Nächstes zeigen wir, wie man die allgemeinen Approximationsraten für die Reoptimierungsszenarien verbessert, bei denen ein Knoten in ein Termial verwandelt werden kann (und umgekehrt). Die resultierenden Approximationsraten sind 1.22 fr eine Modifikation der Menge der Terminale und 1.281 für eine Modifikation der Kosten der Kanten. Als Alternative dazu stellen wir Reoptimisierungsalgorithmen vor, welche nicht auf der Methode der Selbstreduktion aufbauen. Diese liefern keine guten Approximationsraten, aber verbessern die Laufzeit erheblich. Ein Teil dieser Resultate ist in [97] veröffentlicht.