Information Content of Online Problems Advice versus Determinism and Randomization

Author(s):
Smula, Jasmin

Publication Date:
2015

Permanent Link:

https://doi.org/10.3929/ethz-a-010497710 →

Diss. ETH No. 22591

Information Content of Online Problems
Advice versus Determinism and Randomization

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

JASMIN SMULA

Dipl.-Inf., TU Dortmund
born on September 14, 1984
citizen of Germany

accepted on the recommendation of

Prof. Dr. Juraj Hromkovič, examiner
Prof. Dr. Peter Widmayer, co-examiner
Prof. Dr. Rastislav Královič, co-examiner
Dr. Dennis Komm, co-examiner

2015

Abstract

In online computation, an algorithm has to solve some optimization problem while receiving the input instance gradually, without any knowledge about the future input. Such an online algorithm has to compute parts of the output for parts of the input, based on what it knows about the input so far and without being able to revoke its decisions later. Almost inevitably, the algorithm makes a bad choice at some point that leads to a solution that is suboptimal with respect to the whole input instance. Compared to an offline algorithm that is given the entire input instance at once, the online algorithm thus has a substantial handicap. Developing online algorithms that nonetheless compute solutions of some adequate quality is a large and rich field of research within computer science.

The quality of online algorithms is traditionally measured in terms of the competitive ratio, which compares the solutions computed by the online algorithm to those of an optimal offline algorithm. Depending on the online problem at hand, it can differ considerably how much an online algorithm's competitiveness suffers from the lack of knowledge about the input. For some problems such as the ski rental problem, there are online algorithms that can guarantee to compute 2-competitive solutions on any input; on the other hand, there are online problems for which no competitive algorithms exist at all (meaning there are no algorithms achieving any constant competitive ratio).

In some sense, this way of measuring the hardness of online problems is rather rough as a complete lack of knowledge is unrealistic for many real-world applications. With its additional knowledge about the input, the optimal offline algorithm has a huge advantage compared to any online algorithm. Therefore, another way of measuring the complexity of a given online problem has been introduced: the *advice complexity*. Advice complexity theory deals with the question of how much information an online algorithm lacks to be able to compute solutions with some satisfactory competitive ratio. More precisely, assuming some all-knowing oracle of unlimited computing power that knows the entire input, we are interested in the number of bits that are necessary and/or sufficient for any online algorithm to compute solutions of some specified competitive ratio; we call these bits *advice bits* and the corresponding algorithm an *online algorithm with advice*. The mini-

mum number of advice bits necessary and sufficient to be optimal is called the *information content* of the online problem at hand.

The advice complexity and the information content have already been analyzed for many different online problems and are a topic of ongoing research. In this thesis, we further investigate the following online problems with respect to their advice complexity; the k-server problem, the disjoint path allocation problem, online graph searching and graph exploration, and the string guessing problem.

For the k-server problem, we prove a lower bound on the advice complexity of any online algorithm with advice with a competitive ratio of up to $3/2$ that already holds for paths of length 2. Our result improves upon the yet best known trade-off, which only applies for competitive ratios of less than $5/4$. For finite paths of arbitrary length, we give another lower bound yielding better results for near-optimal competitive ratios.

For the disjoint path allocation problem, we show a general lower bound on the advice complexity for a wide range of competitive ratios, from constant up to logarithmic in the path length. From this general lower bound, several lower bounds for concrete ranges of the competitive ratio can be derived. One result we present in this thesis is a lower bound of a linear number of advice bits necessary to achieve any constant strict competitive ratio. This bound implies a surprising threshold behavior of the advice complexity of the disjoint path allocation problem on paths. Although a double logarithmic number of advice bits is sufficient to obtain a competitive ratio that is logarithmic in the path length, any sublinear number of additional advice bits is not enough to further decrease this competitive ratio by another constant factor.

For the graph searching problem, we give asymptotically matching lower and upper bounds of $\Theta(n/c)$ advice bits to achieve c-competitiveness, for any (not necessarily constant) c. In the context of graph exploration, we present a lower-bound result that makes use of a new reduction technique developed to prove trade-offs between the number of advice bits necessary and the competitive ratio of a certain class of online problems, which includes the graph exploration and the graph searching problem.

We also investigate the advice complexity of the string guessing problem in a new probabilistic model featuring a more powerful adversary. In this scenario, we consider two different ways of modeling the oracle. As the setting contains a probabilistic element, the quality of the solution computed by an algorithm is a random variable, and an algorithm can only try to optimize the expected value of this random variable in its favor. For both kinds of oracles considered, we give asymptotically matching lower and upper bounds for the number of advice bits necessary and sufficient to obtain solutions for which this expected value is optimal.

Zusammenfassung

Algorithmen für Online-Optimierungsprobleme erhalten die jeweilige Eingabe-Instanz stückweise, ohne dabei zukünftige Teile der Eingabe zu kennen. Dies bedeutet, dass ein solcher Online-Algorithmus Teile der Ausgabe berechnen muss, die nur auf dem bereits bekannten Teil der Eingabe basieren. Ferner darf er seine Entscheidungen hiernach nicht mehr revidieren. Es ist fast unvermeidbar, dass er dabei früher oder später eine schlechte Entscheidung trifft, die zu einer suboptimalen Lösung bezüglich der gesamten Instanz führt. Wir sehen, dass Online-Algorithmen einen offensichtlichen Nachteil gegenüber Offline-Algorithmen haben, die die gesamte Eingabe von Beginn an kennen. Die Entwicklung von Online-Algorithmen, die dennoch eine gute Lösungsqualität erzielen, ist ein grosses und interessantes Forschungsgebiet innerhalb der Informatik.

Die Qualität von Online-Algorithmen wird klassischerweise durch den kompetitiven Faktor beschrieben, der ihre Lösungen mit denen von optimalen Offline-Algorithmen vergleicht. Je nach betrachtetem Problem ist der Qualitätsverlust aufgrund der teils unbekannten Eingabe sehr unterschiedlich ausgeprägt. Es existieren Online-Probleme, beispielsweise das Ski-Rental-Problem, für die Online-Algorithmen bekannt sind, die eine 2-kompetitive Lösung auf jeder Eingabe garantieren können. Andererseits gibt es Online-Probleme, für die keine kompetitiven Algorithmen existieren (was bedeutet, dass es keine Algorithmen mit konstantem kompetitiven Faktor für sie gibt).

Der kompetitive Faktor ist allerdings in einem gewissen Sinne recht grob, da es in vielen praktischen Situationen unrealistisch ist anzunehmen, dass gar nichts über die Eingabe bekannt ist. Durch sein zusätzliches Wissen über die Eingabe hat ein optimaler Offline-Algorithmus einen enormen Vorteil gegenüber Online-Algorithmen. Aus diesem Grund wurde ein weiteres Mass für die Komplexität von Online-Problemen eingeführt: *Advice-Komplexität*. Das Modell der Advice-Komplexität untersucht die Frage, was für Informationen ein Online-Algorithmus benötigt, um Lösungen mit einem zufriedenstellenden kompetitiven Faktor zu erreichen. Hierzu gehen wir von einem allwissenden Orakel aus, das über unbeschränkte Ressourcen verfügt und welches die gesamte Eingabe kennt. Dieses Orakel kann dem Online-Algorithmus binäre Informationen über die Eingabe

bereitstellen; diese werden als *Advice-Bits* bezeichnet und ein entsprechender Algorithmus als *Online-Algorithmus mit Advice*. Wir interessieren uns für die Anzahl dieser Advice-Bits, die nötig bzw. ausreichend sind, damit Lösungen mit einem gegebenen kompetitiven Faktor berechnet werden können. Die minimale Anzahl, die es ermöglicht eine optimale Lösung zu berechnen, wird als *Informationsgehalt* des gegebenen Online-Problems bezeichnet.

Die Advice-Komplexität und der damit verbundene Informationsgehalt wurden bereits für viele verschiedene Online-Probleme untersucht und sind Gegenstand laufender Forschung. In dieser Arbeit untersuchen wir die folgenden Probleme hinsichtlich ihrer Advice-Komplexität: Das k-Server-Problem, das Disjoint-Path-Allocation-Problem, Online-Graph-Searching und Online-Graph-Exploration und das String-Guessing-Problem.

Für das k-Server-Problem beweisen wir eine untere Schranke für die Advice-Komplexität für einen beliebigen Online-Algorithmus mit Advice, der einen kompetitiven Faktor von bis zu $3/2$ erzielt. Diese Schranke gilt bereits für Pfade der Länge 2. Unser Ergebnis verbessert den bisher besten bekannten Trade-Off, der nur eine Aussage über kompetitive Faktoren von bis zu $5/4$ macht. Für endliche Pfade beliebiger Länge zeigen wir eine weitere untere Schranke, die verbesserte Resultate für fast-optimale kompetitive Faktoren liefert.

Des Weiteren zeigen wir eine allgemeine untere Schranke für das Disjoint-Path-Allocation-Problem, die für einen grossen Bereich von kompetitiven Faktoren gilt, von konstanten Werten bis hin zu Werten, die logarithmisch in der Pfadlänge sind. Dieses allgemeine Ergebnis impliziert diverse untere Schranken für konkrete Bereiche des erreichbaren kompetitiven Faktors. Wir erhalten so unter anderem eine untere Schranke, die aussagt, dass eine lineare Anzahl an Advice-Bits nötig ist, um einen beliebigen konstanten strikten kompetitiven Faktor garantieren zu können. Dies zeigt wiederum ein überraschendes Schwellwertverhalten der Advice-Komplexität des Disjoint-Path-Allocation-Problems auf Pfaden. Obwohl bereits doppelt logarithmisch viele Advice-Bits ausreichen, um einen kompetitiven Faktor zu erhalten, der logarithmisch in der Pfadlänge ist, reicht keine sublineare Anzahl an Advice-Bits, um diesen Faktor um eine weitere Konstante zu verbessern.

Für das Graph-Searching-Problem zeigen wir asymptotisch scharfe untere und obere Schranken von $\Theta(n/c)$ Advice-Bits, um c-Kompetitivität zu erreichen, für jedes beliebige (nicht notwendigerweise konstante) c. Für Graph-Exploration beweisen wir eine untere Schranke mithilfe einer neuen Reduktionstechnik, die es erlaubt, für eine gewisse Klasse von Online-Problemen Trade-Offs zwischen der Anzahl der Advice-Bits und dem kompetitiven Faktor zu beweisen, zu der sowohl Graph-Exploration als auch Graph-Searching gehört.

Wir untersuchen darüber hinaus das String-Guessing-Problem in einem neuen probabilistischen Modell, das einen stärkeren Gegenspieler besitzt. In diesem Szenario betrachten wir zwei verschiedene Arten, das Orakel zu modellieren. Da das

gewählte Setting eine probabilistische Komponente enthält, ist die Ausgabequalität der vom Online-Algorithmus berechneten Lösung eine Zufallsvariable, und der Algorithmus kann lediglich versuchen, deren Erwartungswert zu seinen Gunsten zu optimieren. Für beide untersuchten Orakel-Modelle beweisen wir asymptotisch scharfe untere und obere Schranken bezüglich der Anzahl an Advice-Bits, die nötig und ausreichend sind, um diesen Erwartungswert zu optimieren.