Information and Computation 257 (2017) 65–78

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

FPT approximation schemes for maximizing submodular functions ☆

Piotr Skowron

TU Berlin, Berlin, Germany

A R T I C L E I N F O

Article history:
Received 7 December 2016
Received in revised form 10 August 2017

Keywords:
Parameterized complexity
Fixed parameter tractability
Approximation algorithms
Computational social choice
Matching

A B S T R A C T

We investigate the existence of approximation algorithms for maximization of submodular functions, that run in a fixed parameter tractable (FPT) time. Given a non-decreasing submodular set function $v\colon 2^X \to \mathbb{R}$ the goal is to select a subset S of K elements from X such that $v(S)$ is maximized. We identify three properties of set functions, referred to as p-separability properties, and we argue that many real-life problems can be expressed as maximization of submodular, p-separable functions, with low values of the parameter p. We present FPT approximation schemes for the minimization and maximization variants of the problem, for several parameters that depend on characteristics of the optimized set function, such as p and K.

1. Introduction

We study (exponential-time) approximation algorithms for maximizing non-decreasing submodular set functions. A set function $v\colon 2^X \to \mathbb{R}$ is submodular if for each two subsets $A \subseteq B \subset X$ and each element $x \in X \setminus B$ it holds that $v(A \cup \{x\}) - v(A) \geq v(B \cup \{x\}) - v(B)$; v is non-decreasing if for each two subsets $A \subseteq B \subset X$ it holds that $v(A) \leq v(B)$. We consider the problem where the goal is to select a subset S of K elements from X such that the value $v(S)$ is maximal.

Maximization of non-decreasing submodular functions is a very general problem that is extensively used in various research areas, from recommendation systems [1,2], through voting theory [3,1], image engineering [4–6], information retrieval [7,8], network design [9,10], clustering [11], speech recognition [12], to sparse methods [13,14]. Algorithms for maximization of non-decreasing submodular functions are applicable to other general problems of fundamental significance, such as the MaxCover problem [15,16]. The universal relevance of the problem implies that the existence of good (approximation) algorithms for it is highly desired.

Indeed, the problem has already received a considerable amount of attention in the scientific community. For instance, it is known that the problem is NP-hard (on the other hand, Iwata et al. [17] have shown that minimization of submodular functions is solvable in polynomial time) and that the greedy algorithm, i.e., the algorithm that starts with the empty set and in each of K consecutive steps adds to the partial solution such an element from X that increases the value of the optimized function most, is an $(1 - 1/e)$-approximation algorithm for maximization of non-decreasing submodular functions [18]. The same approximation ratio can be achieved for the distributed [19] and online [20] variants of the problem. Algorithms for maximizing non-monotone submodular functions have been studied by Feige et al. [21], and the approximability of the problem with additional constraints has been investigated by Calinescu et al. [22], Sviridenko [23], Lee et al. [24], and

☆ The preliminary version of this paper was presented at the 12th Conference on Web and Internet Economics (WINE-2016).
E-mail address: p.k.skowron@gmail.com.

https://doi.org/10.1016/j.ic.2017.10.002

Vondrák et al. [25]. For the survey on maximization of submodular functions we refer the reader to the work of Krause and Golovin [26].

Unfortunately, the approximation guarantees of the greedy algorithm cannot be improved without compromising the efficiency of computation (there are some notable exceptions when the submodular function has a specific structure, for instance when it has low curvature [27,28]). Indeed, the MaxCover problem can be expressed as maximization of a non-decreasing submodular function, yet it is known that under standard complexity assumptions no polynomial-time algorithm can approximate it better than with ratio $(1 - 1/e)$ [29]. Motivated by this fact, and provoked by the desire to obtain better approximation guarantees, we turn our attention to algorithms that run in super-polynomial time. In our studies we follow the approach of parameterized complexity theory and look for algorithms that run in fixed parameter tractable time (in FPT time), for some natural parameters. To the best of our knowledge, FPT approximation of optimizing submodular functions has not been considered in the literature before.

Parameterized complexity theory aims at investigating how the complexity of a problem depends on the size of different parts of input instances, called parameters. An algorithm runs in FPT time for a parameter P if it solves each instance I of the problem in time $O(f(|P|) \cdot \mathrm{poly}(|I|))$, where f is a computable function. This definition excludes a large class of algorithms, such as the ones with complexity $O(|I|^{|P|})$. From the point of view of parameterized complexity, FPT is seen as the class of easy problems. Intuitively, the complexity of an FPT algorithm consists of two parts: $f(|P|)$, which is relatively low for small values of the parameter, and $\mathrm{poly}(|I|)$ which is relatively low even for larger instances, because of polynomial relation between the computation time and the size of an instance. For details on parameterized complexity theory, we point the reader to appropriate overviews [30–33].

When referring to running times of the algorithms we assume that the size of the input is $|X|$, and we count each access to the submodular function as an elementary operation (i.e., we assume that the access to the submodular function is given by an oracle). For example, when we say that an algorithm for maximizing submodular functions runs in polynomial-time, then in particular this means that the number of accesses of this algorithm to the submodular functions is of the order of $O(|X|^c)$ for some constant c.

We identify several parameters that we believe are suitable for the analysis of the complexity of the maximization of non-decreasing submodular functions. Perhaps the most natural parameter to consider is the required size of solutions, K. Our other parameters depend on characteristics of the optimized set function. Specifically, we define a new property of set functions, called p-separability, and provide evidence that p is a natural parameter to consider. We do that in Section 4, by presenting several examples of real-life computational problems that can be expressed as maximization of submodular p-separable set functions, where the value of p is small.

Our main contribution is a presentation and an analysis of a few algorithms for the problem. We construct fixed parameter tractable approximation schemes, i.e., collections of algorithms that run in FPT time and that can achieve arbitrarily good approximation ratios. We provide algorithms for two variants of the problem: in the first variant, referred to as the *maximization variant*, the goal is to maximize the value $v(S)$. In the second one, referred to as the *minimization variant*, the goal is to minimize $\big(v(X) - v(S)\big)$. While these two variants of the problem have the same optimal solutions, they are not equivalent in terms of their approximability. Indeed, if there exists a solution S with objectively high value, i.e., if $v(S)$ is close to $v(X)$, then an approximation algorithm for the minimization variant of the problem will be usually superior. For instance, if there exists a solution S such that $v(S) = 0.95 \cdot v(X)$, then a 2-approximation algorithm for the minimization variant of the problem is guaranteed to return a solution with the value better than $0.9 \cdot v(X)$. On the other hand, a $1/2$-approximation algorithm for the maximization variant of the problem may return in such a case a solution with value $0.475 \cdot v(X)$. Conversely, if the value of an optimal solution is significantly lower than the value of the whole set X, then a good approximation algorithm for the maximization variant of the problem will produce solutions of a better quality.

Our algorithms run in FPT time for the parameter (K, p), where K is the size of the solution, and p is the lowest value such that the set function is p-separable. To address the case of functions which are not p-separable for any reasonable values p, we define a weaker form of approximability, referred to as approximation of the *minimization-or-maximization variant*—here, the goal is to find a subset S that is good in one of the previous two metrics. Such algorithms are also desired as they are guaranteed to find good approximation solutions, provided high quality solutions exist (i.e., if values of the optimal solutions are close to $v(X)$). We show that there exists a randomized FPT approximation scheme for minimization-or-maximization variant of the problem for the parameter $\left(K, \frac{\sum_{x \in X} v(\{x\})}{v(X)}\right)$.

We believe that the consequences of our general results are quite significant. In particular, in Section 4, we prove the existence of FPT approximation schemes for some natural problems in the computational social choice, in the matching theory, and in the theoretical computer science.

2. Notation and definitions

Let X denote the universe set. We consider a set function $v : 2^X \to \mathbb{R}$ that is non-negative, i.e., such that for each $S \subseteq X$ we have $v(S) \geq 0$. We recall that a set function v is submodular if for each two subsets $A \subseteq B \subset X$ and each element $x \in X \setminus B$ it holds that $v(A \cup \{x\}) - v(A) \geq v(B \cup \{x\}) - v(B)$. There are numerous equivalent conditions characterizing submodular functions—for a survey we refer the reader to the seminal article of Nemhauser et al. [18]. It is easy to see that if the set function v is non-decreasing and submodular, then for each two subsets $A \subseteq B \subset X$ and each element $x \in X$ it holds that $v(A \cup \{x\}) - v(A) \geq v(B \cup \{x\}) - v(B)$ (here, we do not have to assume that $x \in X \setminus B$).