Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Exact and approximate algorithms for movement problems on (special classes of) graphs ☆,☆☆

Davide Bilò [a], Luciano Gualà [b], Stefano Leucci [c,*], Guido Proietti [c,d]

[a] Dipartimento di Scienze Umanistiche e Sociali, Università di Sassari, Italy
[b] Dipartimento di Ingegneria dell'Impresa, Università di Roma "Tor Vergata", Italy
[c] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila, Italy
[d] Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy

ARTICLE INFO

Article history:
Received 17 October 2014
Received in revised form 14 March 2016
Accepted 10 September 2016
Available online xxxx
Communicated by D. Peleg

Keywords:
Motion planning problems
Approximation algorithms
Shortest path movement
Graph algorithms

ABSTRACT

When a large collection of objects (e.g., robots, sensors, etc.) has to be deployed in a given environment, it is often required to plan a coordinated motion of the objects from their initial position to a final configuration enjoying some global property. In such a scenario, the problem of minimizing some function of the distance travelled, and therefore of reducing energy consumption, is of vital importance. In this paper we study several motion planning problems that arise when the objects initially sit on the vertices of a graph, and they must be moved so as that the final vertices that receive (at least) one object induce a subgraph enjoying a given property. In particular, we consider the notable properties of connectivity, independence, completeness, and finally that of being a vertex-cutset w.r.t. a pair of fixed vertices. We study these problems with the aim of minimizing a number of natural measures, namely the average/overall distance travelled, the maximum distance travelled, and the number of objects that need to be moved. To this respect, we provide several approximability and inapproximability results, most of which are tight.

1. Introduction

In many practical applications a number of centrally controlled objects need to be moved in a given environment in order to complete some task. Problems of this kind often occur in robot motion planning where we seek to move a set of robots from their starting position to a set of ending positions such that a certain property is satisfied. For example, if the robots are equipped with a short range communication device we might want to move them so that a message originating from one of the robots can be routed to all the others. If the robots' goal is to monitor a certain area we might want to move them so that they are not too close to each other. Other interesting problems include gathering (placing robots next to each other), monitoring of traffic between two locations, building interference-free networks, and so on. To make things harder, objects to be moved are often equipped with a limited supply of energy. Preserving energy is a critical problem in ad-hoc networking, and movements are expensive. To prolong the lifetime of the objects we seek to minimize the energy

☆ Research partially supported by the Research Grant PRIN 2010 "ARS TechnoMedia" funded by the Italian Ministry of University and Research.
☆☆ A preliminary version of this work appeared in the *Proceedings of the 20th Colloquium on Structural Information and Communication Complexity (SIROCCO'13)*, LNCS 8179, Springer, 322–333, 2013. http://dx.doi.org/10.1007/978-3-319-03578-9_27.

* Corresponding author.
E-mail addresses: davide.bilo@uniss.it (D. Bilò), guala@mat.uniroma2.it (L. Gualà), stefano.leucci@univaq.it (S. Leucci), guido.proietti@univaq.it (G. Proietti).

http://dx.doi.org/10.1016/j.tcs.2016.09.007
0304-3975/

consumed during movements and thus the distance travelled. Sometimes, instead, movements are cheap but before and/or after an object moves it needs to perform expensive operations. In this scenario we might be interested in moving the minimum number of objects needed to reach the goal.

In this paper, we assume the underlying environment is actually a *network*, which can be modelled as an undirected graph G. The moving objects in this graph are centrally controlled *pebbles* that are initially placed on vertices of G, and that can be moved to other vertices by traversing the graph edges. To this respect, we study several movement planning problems that arise by various combinations of final positioning goals and movement optimization measures. In particular, we focus our study on the scenarios where we want the pebbles to be moved to a *connected subgraph* (Con), an *independent set* (Ind), or a *clique* (Clique) of G, while minimizing either the *overall movement* (Sum), the *maximum movement* (Max), or the *number of moved pebbles* (Num). We also give some preliminary results on the problem of moving the pebbles to an *s-t vertex-cutset*, i.e., a set of vertices whose removal makes two given vertices s, t disconnected (s-t-Cut) while minimizing the above measures.

We will denote each of the above problems with ψ–c, where ψ represents the goal to be achieved and c the measure to be minimized. For a more rigorous definition of the problems we refer the reader to Section 2.

Related work. Although movement problems were deeply investigated in a distributed setting (see [14] for a survey), quite surprisingly the centralized counterpart has received attention from the scientific community only in the last few years.

The first paper which defines and studies these problems in this latter setting is [6]. In their work, the authors study the problem of moving the pebbles on a graph G of n vertices so that their final positions form a *connected component*, a *path* (directed or undirected) *between two specified nodes*, an *independent set*, or a *matching* (two pebbles are matched together if their distance is exactly 1).

Regarding connectivity problems, in [6] the authors show that all the variants are hard and that the approximation ratio of Con-Max is between 2 and $O(1+\sqrt{k/c^*})$, where k is the number of pebbles and c^* denotes the measure of an optimal solution. This result has been improved in [3], where the authors show that Con-Max can be approximated within a constant factor. In [6] it is also shown that Con-Sum and Con-Num are not approximable (in polynomial time) within $O(n^{1-\epsilon})$ (for any positive ϵ) and $o(\log n)$, respectively, while they admit approximation algorithms with ratios of $O(\min\{n \log n, k\})$ and $O(k^{\epsilon})$, respectively. Moreover, the authors also provide an exact polynomial-time algorithm for Con-Max on trees.

Concerning independence problems, in [6] the authors remark that it is NP-hard even to find any feasible solution on general graphs since it would require to find an independent set of size at least k. This clearly holds for all three objective functions. For this reason, they study a Euclidean variant of these problems where pebbles have to be moved on a plane so that their pairwise distances are strictly greater than 1. In this case, the authors provide an approximation algorithm that guarantees an additive error of at most $1+1/\sqrt{3}$ for Ind-Max, and a polynomial-time approximation scheme for Ind-Num.

More recently, in [9], a variant of the classical facility location problem has been studied. This variant, called *mobile facility location*, can be modeled as a movement problem and is approximable, in polynomial time, within $(3+\epsilon)$ (for any constant $\epsilon > 0$) if we seek to minimize the total movement [1], while the variant where the maximum movement has to be minimized admits a tight 2-approximation [6,9]. Moreover, as it is frequent in the practice to have a small number of pebbles compared to the size of the environment (i.e., the vertices of the graph), the authors of [7] turn to study fixed-parameter tractability. They show a relation between the complexity of the problems and their *minimal configurations* (sets of final positions of the pebbles that correspond to feasible solutions, such that any removal of an edge makes them unacceptable). Finally, we mention that in [2] it was considered a set of vertex-to-vertex motion planning problems within a simple polygon, with the aim of forming final configurations enjoying some kind of *visual connectivity* among the pebbles.

Our results. We start by studying connectivity motions problems in the case where pebbles move on a tree, and we devise two polynomial-time dynamic programming algorithms for Con-Sum and Con-Num. These algorithms complement the already known polynomial-time algorithm for Con-Max on trees shown in [6].

Then, we study independence motion problems on graphs where a *maximum independent set* (and thus a feasible solution for the corresponding motion problem) can be computed in polynomial time. This class of graphs includes, for example, perfect and claw-free graphs. More precisely, we show that Ind-Max and Ind-Sum are NP-hard even on bipartite graphs (which are known to be perfect graphs [4]). Moreover, we devise three exact polynomial-time algorithms: one for solving Ind-Max on paths, and the other two for solving Ind-Sum and Ind-Num on trees, respectively. Moreover, we devise a polynomial-time approximation algorithm for Ind-Max whose returned solution cost is within an additive error of 1 from the optimum (this is clearly tight).

Concerning the problem of moving pebbles towards a clique of a general graph, we prove that all the three variants are NP-hard. Then, we provide an approximation algorithm for Clique-Max with an additive error of 1 (this result is clearly tight). Moreover, we show that both Clique-Sum and Clique-Num are approximable (in polynomial time) within a factor of 2, but they are not approximable within a factor better than $10\sqrt{5}-21 > 1.3606$, unless $\mathsf{P} = \mathsf{NP}$. If the *unique game conjecture* [12] is true, then both problems are not approximable within a factor better than 2 and the provided approximation algorithms are tight. These results are obtained by showing a non-trivial relation with the *minimum vertex cover* problem. We also show that an exact solution for Clique-Num can be computed in polynomial time on every class of graphs for which a *maximum-weight clique* can be found in polynomial time (these classes of graphs also include perfect and claw-free graphs).