Movement Problems

CMSC 858F · Network Design

Alejandro Flores Saurabh Kumar

December 2015

1 Introduction

Movement Problems were introduced by [DHM+09], as a general framework that deals with the movement of "pebbles" along a graph in order to achieve some property. Formally, given a graph $G = (V, E)$ and initial pebble positions $\mathcal{P} \subseteq V$, a *movement problem* moves the pebbles to a final position, such that a particular property is fulfilled. Three natural objective functions are drawn from this framework: minimizing the total movement (Sum), the maximum movement (Max), or the number of pebbles moved (Num).

Based on different properties to fulfill, several problems where considered in [DHM+09] (for all three objective functions):

- Con: The graph induced by the pebbles must be connected.
- DirCon: Given G a directed graph, in the graph induced by the pebbles, there is a path between each pebble and some root node.
- Path: For a given pair of nodes s and t, there most be a path of pebbles between s and t in G.
- Ind: No two pebbles are adjacent (or in the same node).
- Match: There is a perfect matching for graph H, in which two pebbles are connected *iff* their final distance in G is at most 1.

Some of these problems (and others) where further explored by [BDZ11, DHM14]. Additionally, [FS11] proposed the Mobile Facility Location problem, in which both clients and facilities move in the graph s.t. each client moves to a node where a facility has also moved. The framework of movement problems was also extended into the geometrical setting by [AFGS15], where pebbles are points in the euclidean plane, and move to achieve some property.

2 The PathMax Problem

In this section we consider the PathMax problem. We make special emphasis in the algorithm proposed by [DHM+09], showing how to approximate the problem within a factor of $O(1 + \sqrt{m/\mathrm{OPT}})$.

Definition 2.1. *Given a graph G with some initial configuration of pebbles, and a pair of nodes s and t, the problem is to move the pebbles s.t. s and t are connected (i.e., there is a path of pebbles from s to t), while minimizing the maximum movement of each pebble.*

Theorem 2.1. PathMax *can be be approximated to* $O(1 + \sqrt{m/\mathrm{OPT}})$.

Proof. The algorithm proceeds as follows. First, we identify some nodes in the graph that *can't* be part of the optimum solution OPT. Every other node is considered *marked*. Then, we move each pebble to its nearest marked node. Now, we generate a shortest path from s to t only considering marked nodes, and distribute the pebbles so as to fill the path selected. A careful analysis shows that each step can be performed while bounding the maximum movement of the pebbles.

Below is a detailed description of the algorithm.

1. **Mark vertices that can be in OPT**

   A vertex v cannot be in OPT if the following holds: $\exists\ i \leq \min(d_{sv}, d_{tv})$ s.t. there are less than $2i + 1$ pebbles within a radius of $\mathrm{OPT} + i$ of v.

   Intuitively, if a node v is a part of a path from s to t, then for any circle of allowed radius i, there must be sufficient pebbles to connect v to two ends of the circle i.e there must be $2i + 1$ pebbles. Since each pebble can move at most OPT by definition, it must be at most $i + \mathrm{OPT}$ away (so that it can move OPT and reach the boundary of the circle in the worst case).

   Mark all the nodes that can be part of OPT based on the above rule.

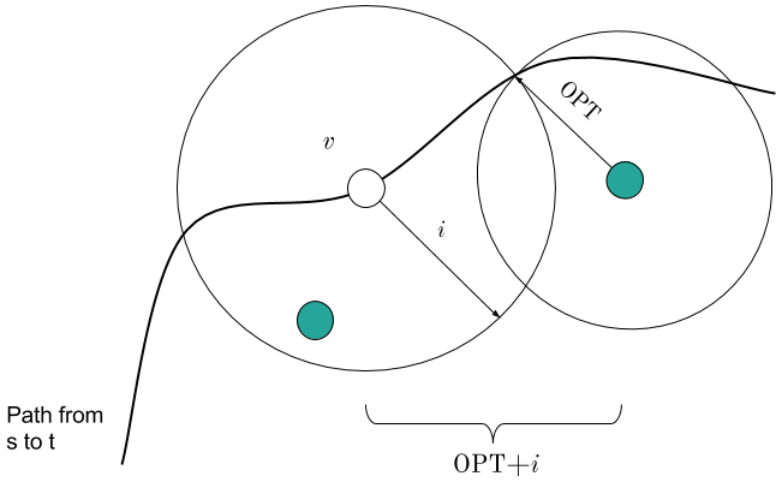

Figure 1: Step 1 - The green nodes represent pebbles and v is the node being considered

2. **Move pebbles to nearest marked vertex**

   Delete all pebbles that are at distance more than OPT from the nearest marked vertex, since these can never be part of the optimum solution. In this step, each pebble moves at most OPT.

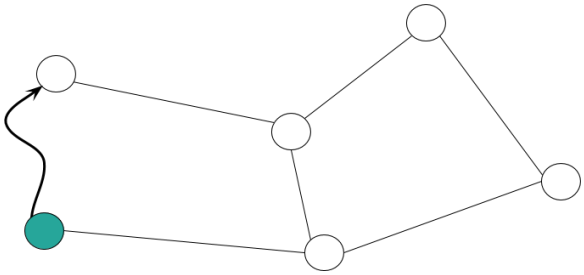

Figure 2: Step 2 - Move the pebbles to the nearest marked vertex

3. **Find a shortest path and select centers**

   Find a shortest path P from s to t, $P = \{s = v_0, v_1, v_2, \ldots, v_{|P|} = t\}$. Define the center vertices as $\{v_k, v_{3k+1}, v_{5k+2}, ...\}$. For each center, move all pebbles within radius k to that center. Since the distance between two centers is $2k + 1$, each pebble is assigned to at most one center. Then, spread the pebbles out to the M empty vertices. The maximum movement for this step is OPT + $2k$.

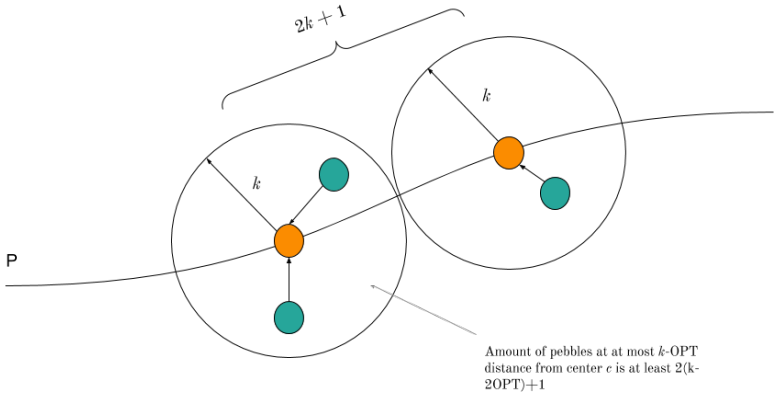

Figure 3: Step 3 - Move the pebbles to the centers

4. **"Good" and "Bad" pebbles**

   We define two types of pebbles: Good pebbles are those that are on P and form part of OPT. Bad pebbles are those that form part of OPT but are not on P (because they were more than k distance away from a center).

An example is shown in Figure 4. Because there are M empty vertices on P, we must have at least M bad pebbles.

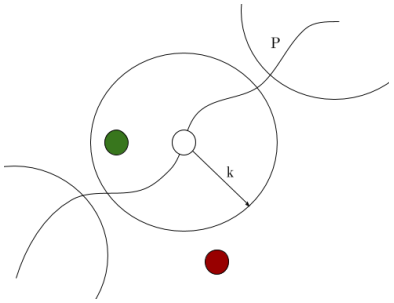

Figure 4: Step 4 - Good pebbles are shown in green, bad ones in red.

Let P_b be a bad pebble and P_g be a good one. Suppose the distance between their target positions in OPT is denoted by $H(P_b, P_g)$. Using bounds on the distances the pebbles have moved so far, we can obtain a $2k + 4\text{OPT} + H(P_b, P_g)$ bound on the maximum distance between a good and bad pebble as shown in Figure 5.

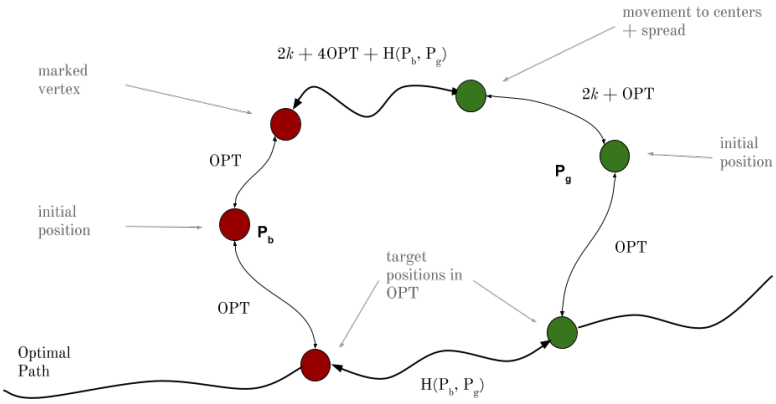

Figure 5: Step 4 - Given a good pebble (green) and bad pebble (red), we can obtain a bound on the distance between them.

Finally, we move bad pebbles to their nearest good pebble in the final configuration on P. On doing so, the pebble that was already at that position gets pushed one forward and each of the successive pebbles move one forward until an empty vertex gets filled. This process is repeated until the entire path is filled.

The analysis of this algorithm yields that the maximum movement of a pebble is $6k + (5 + 4m/k)\text{OPT}$. Then, by setting $k = \sqrt{m\text{OPT}}$, we obtain the desired bound for PathMax. □

References

[AFGS15] Nima Anari, MohammadAmin Fazli, Mohammad Ghodsi, and MohammadAli Safari. Euclidean movement minimization. *Journal of Combinatorial Optimization*, pages 1–14, 2015.

[BDZ11] Piotr Berman, Erik D Demaine, and Morteza Zadimoghaddam. O (1)-approximations for maximum movement problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 62–74. Springer, 2011.

[DHM+09] Erik D Demaine, MohammadTaghi Hajiaghayi, Hamid Mahini, Amin S Sayedi-Roshkhar, Shayan Oveisgharan, and Morteza Zadimoghaddam. Minimizing movement. *ACM Transactions on Algorithms (TALG)*, 5(3):30, 2009.

[DHM14] Erik D Demaine, MohammadTaghi Hajiaghayi, and Dániel Marx. Minimizing movement: Fixed-parameter tractability. *ACM Transactions on Algorithms (TALG)*, 11(2):14, 2014.

[FS11] Zachary Friggstad and Mohammad R Salavatipour. Minimizing movement in mobile facility location problems. *ACM Transactions on Algorithms (TALG)*, 7(3):28, 2011.