Verification Problem of Maximal Points under Uncertainty

George Charalambous and Michael Hoffmann

Department of Computer Science, University of Leicester
gc100@mcs.le.ac.uk, mh55@mcs.le.ac.uk

Abstract. The study of algorithms that handle imprecise input data for which precise data can be requested is an interesting area. In the *verification under uncertainty* setting, which is the focus of this paper, an algorithm is also given an assumed set of precise input data. The aim of the algorithm is to update the smallest set of input data such that if the updated input data is the same as the corresponding assumed input data, a solution can be calculated. We study this setting for the maximal point problem in two dimensions. Here there are three types of data, a set of points $P = \{p_1, \ldots, p_n\}$, the uncertainty areas information consisting of *areas of uncertainty* A_i for each $1 \leq i \leq n$, with $p_i \in A_i$, and the set of $P' = \{p'_1, ..., p'_k\}$ containing the assumed points, with $p'_i \in A_i$. An *update* of an area A_i reveals the actual location of p_i and *verifies* the assumed location if $p'_i = p_i$. The objective of an algorithm is to compute the smallest set of points with the property that, if the updates of these points verify the assumed data, the set of maximal points of P can be computed. We show that the maximal point verification problem is NP-hard, by a reduction from the minimum set cover problem.

1 Introduction

Nowadays more and more information is available. With a flood of sensors connected to a network, such as GPS-enabled mobile phones, up-to-date readings of these sensors are generally available. Algorithms that perform based on such information might not have the precise data available to them, as for example in some situations the traffic of collecting all such information would cause a problem on its own. In other situations, obtaining the up-date information of all sensors is costly in time and battery power or even other charges may occur. A practical solution is to work with slightly out of date data where possible and only request up to date information where needed. For example a sensor may automatically send its current measurement if this exceeds some predefined bounds of the latest send one. Hence based on the last send information a possible band or area is known where the current measurement of the sensor is within. If this rough information is not enough then the precise measurement can be obtained. Therefore an algorithm may have some precise data while at the same time some uncertain data, for which if needed an update request can be made and the precise measurement can be obtained. Problems under uncertainty capture this setting. The aim is to make the fewest update requests that allows the calculation to succeed. In the *verification under uncertainty* setting, which is the focus of this paper, an algorithm is also given an assumed set of precise input data. The aim of the algorithm is to update the smallest set of input data such that if the updated input data is the same as the corresponding assumed input data, a solution can be calculated.

While mobile devices moving in the plane is a classical example to motivate the study of geometric problems in the uncertainty setting, its applications are found in many different areas. For example data is collected in a large number of databases and distributed systems, such as prices and customer ratings of products. As the price and rating of a product may vary over time, an algorithm that has to identify all top items from a collection may work with uncertain data and only request more accurate prices and ratings if needed. With the idea of maximal points in mind, top items would be such that there is no other better in price and rating.

Work in computing under uncertainty falls in three main categories: In the *adaptive online* setting an algorithm initially knows only the uncertainty areas and performs updates one by one (determining the next update based on the information from previous updates) until it has obtained sufficient information to determine a solution. Algorithms are typically evaluated by competitive analysis, comparing the number of updates they make with the minimum number of updates that, in hindsight, would have been sufficient to determine a solution (referred to as the offline optimum). In the *non-adaptive online* setting an algorithm is also given only the uncertainty areas initially, but it must determine a set U of updates such that after performing all updates in U it is guaranteed to have sufficient information to determine a solution. Finally, there is the *verification* setting that was already described above. It is worth noting that the optimal update set of the *verification* setting is also the offline optimum of the adaptive online setting. Therefore, algorithms solving the verification problem are also useful for the experimental evaluation of algorithms for the adaptive online setting.

In this paper, we consider the Maximal Point Verification problem. The maximal point problem is a classical problem. Many aspects of the problem have sparked interesting research. It can be stated as follows: In this paper we considering the 2-dimensional case. Given a set P of points and a partial order of the points, return all points where no point in P is higher. Typically the partial order is based on the coordinates of the points in the following way: a point p is higher than a point q if p_x is greater or equal than q_x and p_y is greater or equal than q_y and $p \neq q$. Such a partial order naturally extends to higher dimensions, but in this paper we only consider 2-dimensional points.

A formal definition of the Maximal Point Verification problem (MPV) is given in Definition 2.

Our main result is, as stated in Theorem 1, that by a reduction from the minimum set cover problem the MPV problem is also NP-hard. In our construction of the reduction each uncertain area contains either a single point (e.g the data is known precisely) or contains just two points. Hence, an MPV problem remains

NP-hard even when restricted to areas of uncertainty that contain at most two points. It remains, however, open if the same holds when each uncertain area is connected.

The effect of our result is significant for experimental evaluation of algorithms in the online and verification setting of the maximal point problem under uncertainty. It strengthens the role of constant competitive online algorithms, as they also represent a constant approximation algorithm for the verification setting. Finally it gives rise to find new restrictions on the uncertainty areas, such that the verification problem becomes solvable in polynomial time, and captures a large variety of applications for maximal points under uncertainty.

Related work:

Kahan [6] presented a model for handling imprecise but updateable input data. He demonstrated his model on a set of real numbers where instead of the precise value of each number an interval was given. That interval when updated reveals that number. The aim is to determine the maximum, the median, or the minimal gap between any two numbers in the set, using as few updates as possible. His work included a competitive analysis for this type of online algorithm, where the number of updates is measured against the optimal number of updates. For the problems considered, he presented online algorithms with optimal competitive ratio. Feder et al. [4] studied the problem of computing the value of the median of an uncertain set of numbers up to a certain tolerance. Applications of uncertainty settings can be found in many different areas including structured data such as graphs, databases, and geometry. The work presented in this paper mainly concerns the latter two areas.

Bruce et al. [1] studied the geometric uncertainty problem in the plane. Here, the input consists of points in the plane and the uncertainty information is for each point of the input an area that contains that point. They gave a definition of the Maximal Point under Uncertainty as well as the Convex Hull under Uncertainty. They presented algorithms with optimal competitive ratio for both problems. The algorithms used are based on a more general technique called *witness set algorithm* that was introduced in their paper.

In [2], Erlebach et al. studied the adaptive online setting for minimum spanning tree (MST) under two types of uncertainty: the edge uncertainty setting, which is the same as the one considered by Feder et al. [3], and the vertex uncertainty setting. In the latter setting, all vertices are points in the plane and the graph is a complete graph with the weight of an edge being the distance between the vertices it connects. The uncertainty is given by areas for the location of each vertex. For both settings, Erlebach et al. presented algorithms with optimal competitive ratio for the MST under uncertainty. The competitive ratios are 2 for edge uncertainty and 4 for vertex uncertainty, and the uncertainty areas must satisfy certain restrictions (which are satisfied by, e.g., open and trivial areas in the edge uncertainty case). A variant of computing under uncertainty where updates yield more refined estimates instead of exact values was studied by Gupta et al. [5].

A different setting of the MST under vertex uncertainty was studied by Kamousi et al. [7]. They assume that point locations are known exactly, but each point i is present only with a certain probability p_i. They show that it is #P-hard to compute the expected length of an MST even in 2-dimensional Euclidean space, and provide a fully polynomial randomized approximation scheme for metric spaces.

Structure of the paper. In Section 2 we give formal definitions and preliminaries. In Section 3 we present our construction of an MPV problem out of a minimum set cover problem. In Section 4 we demonstrate relation between the solutions of these two problems. In Section 5 we complete the proof of Theorem 1.

2 Preliminaries

The general setting of problems with areas of uncertainty can be described in the following way: Each problem instance $P = (C, \mathcal{A}, \phi)$ consists of several components. The ordered set of data $C = \{c_1, \ldots, c_n\}$ is also called a *configuration.* $\mathcal{A}$ is an ordered set of areas $\mathcal{A} = \{A_1, \ldots, A_n\}$, such that $c_i \in C$ is an element of A_i for $1 \leq i \leq n$. The sets A_i are called *areas of uncertainty* or *uncertainty areas* for C. We say that an uncertainty area A_i that consists of a single element is *trivial.* ϕ is a function such that $\phi(C)$ is the set of solutions for P. (The function ϕ is the same for all instances of a problem and can thus be taken to represent the problem.) The aim is to calculate a solution in $\phi(C)$ based on the information of $\mathcal{A}$. If that is not possible, *updates* to elements of $\mathcal{A}$ can be made. These updates alter the set $\mathcal{A}$: After updating A_i, the new ordered set of areas of uncertainty for C is $\{A_1, \ldots, A_{i-1}, \{c_i\}, A_{i+1}, \ldots, A_n\}$. Hence the exact value of c_i is now revealed.

In the online setting, the set C is not known to the algorithm; the algorithm has to request updates until the set $\mathcal{A}$ is precise enough to allow the calculation of a solution in $\phi(C)$ based on $\mathcal{A}$. The *verification setting* is similar. The set C, however, is now also given to the algorithm. This additional information is not used to calculate $\phi(C)$ directly, but is used to determine which update requests should be made so that $\phi(C)$ can be calculated based on $\mathcal{A}$. Updating all non-trivial areas would reveal/verify the configuration C and would obviously allow us to calculate an element of $\phi(C)$ (under the natural assumption that ϕ is computable). A set of updates that reveal enough information of the configuration C such that an element of $\phi(C)$ can be calculated is an *update solution.* The aim of the algorithm is to use the smallest possible number of updates. For a given instance of a problem, we denote an update solution of minimal size also as an optimal update solution.

We use the uncertainty setting in the context of the Maximal Point problem. For this all points discussed in the paper are points in the 2D plane. So a point p may be written in coordinate form (p_x, p_y). We say a point $p = (p_x, p_y)$ is *higher* than a point $q = (q_x, q_y)$ if $p_x >= q_x$ and $p_y >= q_y$ and $p \neq q$. Note that this induces a partial order and leads to the following definition of a maximal point among a set of points.

Definition 1. *Let P be a set of points and p be a point in P. The point p is said to be maximal among P if there does not exist a point in P that is higher than p. Otherwise p is non-maximal among P.*

In the 'under Uncertainty' setting for the Maximal Point problem the set of points $P = \{p_1, \ldots, p_n\}$ is the configuration of the problem. The set of uncertainty areas consists of an area for each point in P. The solution $\phi(P)$ is the index set I such that p_i is maximal among P if and only if $i \in I$. Formally,

Definition 2. *A Maximal Point Verification problem, MPV for short, is a pair $(\mathcal{A}, P)$, where P is a set of points and $\mathcal{A}$ is a set of areas for P. The aim is to identify the smallest set of areas in $\mathcal{A}$, that when updated verifies the maximal points among P as maximal based on the information of $\mathcal{A}$ and the results of the updates.*

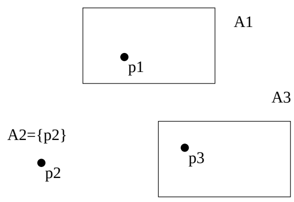

Fig. 1. Example of an MPV problem

In the example shown in figure 1, the problem consists of three points (p_1, p_2 and p_3) and three areas A_1, A_2 and A_3. The area A_2 consist only of the point p_2 and hence A_2 is a trivial area and the location of p_2 is already verified. For every point in the area A_1 there does not exist a point in A_2 or A_3 that is higher. Therefore, regardless of where p_1 lies in A_1 and where p_3 is located in A_3, the point p_1 will be maximal in P. Based only on the areas of uncertainty the point p_3 may or may not be maximal in P. So updates have to be requested to verify some points, and therefore to make the problem solvable based on the initial areas of uncertainty and the information retrieved by the updates. The set $\{A_1, A_3\}$ is clearly an update solution as after updating these two sets the location of p_1 and p_3 are verified and both are maximal points in P. However the set $\{A_1\}$ is also an update solution as after verifying the location of p_1, neither p_1 nor p_2 are higher than any point in A_3. Hence even without verifying the location of p_3 within the area A_3 both p_1 and p_3 must be both maximal in P. In this example the set $\{A_2\}$ is also an optimal update solution as without any update the maximal points can not be calculated. We finish this example by noting that updating just A_3 is not an update solution. While this verifies the exact location of p_3, the area A_1 still contains some points that are higher than p_3 and some that are not. So without also verifying the location of p_1 it is not clear whether $p3$ is a maximal point among P or not.

We will use the following notations:

An area A is said to be *maximal* among a set of areas $\mathcal{A}$ if there does not exist a point in any areas in $\mathcal{A} - A$ that is higher than any point in A. Similarly, an area A is said to be *non-maximal* among a set of areas $\mathcal{A}$ if for every point $p \in A$ there is an area $B \in \mathcal{A}$ such that every point in B is higher than p.

We also note that an area might be neither maximal nor non-maximal. If this is the case then the set of maximal points cannot be calculated. In other words a problem is *solved* if and only if all areas in $\mathcal{A}$ are either maximal or non-maximal among $\mathcal{A}$.

In the last part of this section we recall the Minimum Set Cover problem. The Minimum Set Cover (MSC) problem consists of a universe U and a family $\mathcal{S}$ of subsets of U. The aim is to find a family of sets in $\mathcal{S}$ of minimal size that covers U. It was shown by [8] that the problem is NP-Hard. Without loss of generality we assume that every element in U is found in at least one set of $\mathcal{S}$ and that all sets in $\mathcal{S}$ have size of at least 2.

Theorem 1. *Solving the Maximal Point Verification problem is NP-hard.*

3 MP-construction

In this section we give the construction of an MPV problem out of an MSC problem. We call the instance of the MSC problem MC $= (U, \mathcal{S})$ with $U = \{1, \ldots, n\}$ and $\mathcal{S} = \{S_1, \ldots, S_k\}$. The instance of the MPV will be denoted by MP.

The idea behind the construction is to have different types of areas in MP representing different aspects of MC. A set of areas (B's) will correspond to elements of U and another set of areas (A's) will correspond to elements of each $S_j \in \mathcal{S}$. The areas are positioned that for each area corresponding to an element of U, at least one area corresponding to the occurrence of i in the set S_j must be included in any update solution. With the help of another set of areas (D's), the areas corresponding to element of a set S_j are linked together. So, if an update solution contains one area corresponding to an element of a set S_j the update solution can be modified to include all areas that correspond to elements of S_j without increasing the size of the update solution.

The construction is done by using three different types of gadgets, which are all of rectangular shape. These gadgets are placed in the plane in such a way that no point in one gadget is higher than any point in another gadget. This can be achieved by placing all gadgets diagonally top-left to bottom-right in the plane, see figure 2.

Type 1 gadget. For each $i \in U$ there exists one gadget of type 1. This contains the point b_i, which is the lower left corner of the gadget, and multiple disjoint points along the diagonal of the gadget. For each set $S_j \in \mathcal{S}$ that contains i, a point a^i_j is placed on the diagonal. See figure 3.

Type 2 gadget. For each set $S_j \in S$ there exists one gadget of type 2. This contains for every $i \in S_j$ a point c^i_j along the diagonal of the gadget such that

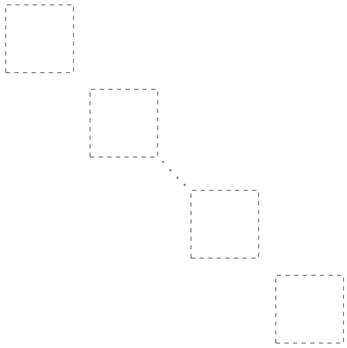

Fig. 2. Placement of gadgets

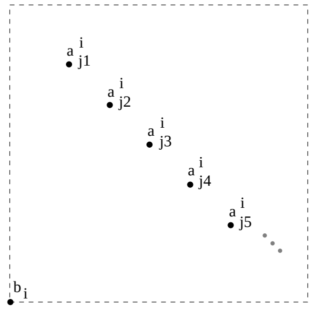

Fig. 3. Type 1 gadget

all points are pairwise disjoint. In addition points $d_j^1, \ldots, d_j^t$ with $t = |S_j| - 1$ are placed in such a way that for each d_j^r with $1 \leq r \leq t$ there exist exactly two points c_j^i and $c_j^{i'}$ that are higher. Furthermore any two neighbouring points c_j^i and $c_j^{i'}$ are higher than exactly one type d point. This can be done easily by placing the type d points along a line that is parallel to the diagonal, and closer to the bottom-left corner of the gadget than the diagonal. See figure 4.

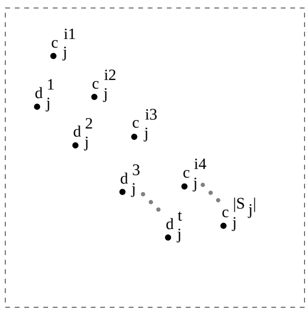

Fig. 4. Type 2 gadget

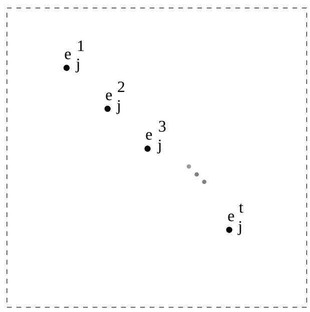

Fig. 5. Type 3 gadget

Type 3 gadget. For each set $S_j \in \mathcal{S}$ there exists one gadget of type 3. This gadget just consists of $|S_j| - 1$ disjoint points $e_j^1, \ldots, e_j^t$ placed along the diagonal. See figure 5.

The various points placed in the three gadgets, are now used to define the areas of uncertainty $\mathcal{A}$, and the set of price points P for MP.

Out of the points from the different gadgets we build the following sets where each set corresponds to an area for MP. For all $i \in U$ let B_i be the set containing only b_i. For all $i \in U$ and $S_j \in \mathcal{S}$ with $i \in S_j$ let A_j^i be the set containing the two points a_j^i and c_j^i. For all $S_j \in \mathcal{S}$ and $1 \leq r \leq |S_j| - 1$ let D_j^r be the set containing the two points d_j^r and e_j^r.

To handle these sets better in the remaining part of the paper we group some of these areas together. We say $A_j = \{A_j^i \mid i \in S_j\}$ and $D_j = \{D_j^1, \ldots, D_j^t\}$ with $t = |S_j| - 1$. We also note that $|A_j| = |D_j| + 1 = |S_j|$.

Further we say B is the set of all areas that correspond to an element of U (or formally $B = \{B_1, \ldots, B_n\}$), A is the set of all areas that correspond to an element of any set $S_j \in \mathcal{S}$ (or formally $A = \cup_{S_j \in \mathcal{S}} A_j$) and D is the set of all areas in any D_j (or formally $D = \cup_{S_j \in \mathcal{S}} D_j$).

This allows us to define our instance of the MPV in the following way: MP $= (\mathcal{A}, P)$ with $\mathcal{A} = B \cup A \cup D$ and $P = \{b_1, \ldots, b_n\} \cup \{a_j^i \mid i \in S_j\} \cup \{e_j^r \mid 1 \leq r \leq |S_j| - 1\}$.

We are now analysing the constructed problem MP and highlight properties that are needed in the further section.

Size of MP. There exist exactly n type 1 gadgets where each contains one type b point. Each type 1 gadget also contains at most k number of type a points. There exist exactly k type 2 gadgets. Each contains at most n type c points and $n-1$ type d points. There exist exactly k type 3 gadgets. Each contains at most $n-1$ type e points.

Hence for the MP constructed we have $n + 2k$ gadgets and at most $n * (1 + k) + k * (2n - 1) + k * (n - 1) = n + 4nk - 2k$ points. As each point only lies in one area of uncertainty also $|\mathcal{A}|$ is at most $n + 4nk - 2k$ and so the input size of MP is polynomial in the size of MC.

Maximal points among P. A point a_j^i for some j and i is part of a type 1 gadget and is clearly maximal among all points placed in the gadget. As two different gadgets are located so that no point of one is higher than a point of another, all points a_j^i are maximal in P. The same follows for the all points e_j^r in type 3 gadgets and therefore all such points are also maximal among P.

As for every $i \in U$ there must exists at least one $S_j \in \mathcal{S}$ with $i \in S_j$, by the construction of the type 1 gadget for i, also the point a_j^i was added to that gadget. As all such points are higher than b_i the point b_i is non-maximal among P.

Maximal areas among $\mathcal{A}$. Each area in A consists of two points a_j^i and c_j^i. One is located inside a type 1 gadget and the other inside a type 2 gadget. For both points there is no area in $\mathcal{A}$ with a higher point, and therefore even without any updates all areas in A are maximal.

For each area $B_i \in B$ there exist some areas in A with a point above B_i and one point not above B_i. So among $\mathcal{A}$ the area B_i is neither maximal nor non-maximal and further updates are needed.

Each area in D has two points. One is located in a type 3 gadget which is clearly maximal; and one located in a type 2 gadget where there are two areas in A that contain points that are higher. So among $\mathcal{A}$ it is neither maximal nor non-maximal and further updates are needed.

Update solutions for MP. Following from the above analysis of maximal areas among $\mathcal{A}$ we have the following remark:

Remark 1. A set of areas is an update solution if and only if it contains for each i an area A_j^i for some j, and also for each area in D either this area or the two areas in A that are potentially higher.

Following from this only updates of areas in A_j and D_j will help to identify areas of D_j to be maximal. Based on the construction of type 2 gadgets, updating k areas of A_j can at most identify $k-1$ areas of D_j as maximal. Hence the smallest update set that identifies all areas of D_j as maximal is D_j itself. Any other set of updates must be bigger. Formally:

Remark 2. Let R be an update solution. Then for j the set R must contain either D_j or it must contain at least $|D_j|+1$ areas of $D_j \cup A_j$.

This leads to the following Lemma:

Lemma 1. *Let R be an update solution for MP and let $A_i^j \in R$ for some j and i be an area. Then $R' = R - D_j + A_j$ is also an update solution and $|R'| \leq |R|$.*

Proof. Since R is an update solution, by Remark 1 for every $i \in U$ the set R must contain an area $A_{j'}^i$ for some j'. As R' in constructed by potentially removing areas of D and adding areas of A the set R' must also contain the area $A_{j'}^i$.

Let $D_{j''}^r \in D$. Again by Remark 1 either $D_{j''}^r \in R$ or the two elements in A that are higher than $D_{j''}^r$ are in R. If R contains the two areas in A that are higher than $D_{j''}^r$ then also R' must contain these areas as no area in A was removed when creating R'. If $D_{j''}^r \in R$ also R' must contain $D_{j''}^r$ unless $j'' = j$. In that case as all areas in A_j were added to R' it must also contain the two areas in A_j that are higher than $D_{j''}^r$. Hence by Remark 1 also R' is an update solution.

We now show that $|R'| \leq |R|$. As $A_j^r \in R$, by Remark 2 R must contain at least $|D_j|+1$ areas out of $D_j \cup A_j$. We noted in the construction of MP that $|A_j| = |D_j| + 1 = |S_j|$. So R' includes exactly $|D_j|+1$ areas out of $D_j \cup A_j$. As R and R' only differ in selection of areas of D_j and A_j we have that $|R'| \leq |R|$.

4 Relating Update Solutions to Covers

In this section we show how to construct a cover of MC out of an update solution of MP and vice versa. We will also note how the size of the update solutions and covers relate to each other.

From update solution to cover. Let R be an update solution for MP.

Before creating the cover we create a different update solution R'. The set R' is based on R but for all j such that there exists an i with $A_j^i \in R$ all potential areas of D_j are removed from R and all areas of A_j are added. By Lemma 1, we have that R' is also an update solution with no greater size than R. Furthermore by doing so, the update solution R' contains for every index j either the set A_j or D_j but never a mixture.

The cover $\mathcal{C}$ is constructed based on R' in the following way. For each index j such that $A_j \subseteq R'$ we choose the set $S_j \in \mathcal{S}$ to be included in $\mathcal{C}$ and otherwise not.

This is denoted as:

$$\mathcal{C} = \{S_j \in \mathcal{S} \mid A_j \subseteq R'\}$$

We now show that $\mathcal{C}$ is a cover, in other words that every element of U is found in at least one set of $\mathcal{C}$.

Let some $i \in U$ for the MC. Then in MP there exists the area B_i. By remark 1 there exists a j with $A_j^i \in R'$. Since this area A_j^i was constructed in the creation of MP we have that $i \in S_j$. As A_j^i is also in R' the set A_j must be a subset of R and $S_j \in \mathcal{C}$.

We note that the construction of $\mathcal{C}$ is done in polynomial time and the sizes of R, R' and C relate to each other in the following way.

By the construction of R' we have:

$$|R'| = \sum_{A_j \subseteq R'} |A_j| \ + \sum_{A_j \not\subseteq R'} |D_j|$$

As $|A_j| = |D_j| + 1 = |S_j|$ for all j we get by the construction of $\mathcal{C}$ that:

$$|R'| = \sum_{S_j \in \mathcal{C}} |S_j| \ + \sum_{S_j \in \mathcal{S}-\mathcal{C}} (|S_j| - 1)$$

$$= |\mathcal{C}| + \sum_{S_j \in \mathcal{C}} (|S_j| - 1) \ + \sum_{S_j \in \mathcal{S}-\mathcal{C}} (|S_j| - 1)$$

$$= |\mathcal{C}| + \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$$

Since by Lemma 1 we get $|R| \geq |R'|$ we have:

$$|R| \geq |\mathcal{C}| \ + \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$$

We summarise our results on the construction of $\mathcal{C}$ in the following Lemma:

Lemma 2. *Let R be an update solution for MP. Then a cover of MC can be constructed in polynomial time with* $|R| \geq |\mathcal{C}| + \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$.

From cover to update solution. Similarly to the construction of a cover for MC out of a given update solution of MP, we now show how to construct an update solution for MP out of a given cover for MC.

Let $\mathcal{C}$ be a cover for MC.

The set R of areas in MP is based on $\mathcal{C}$ as follows: For each index j such that $S_j \in \mathcal{C}$ we choose the set A_j to be included in R. For each index j such that $S_j \in (\mathcal{S} - \mathcal{C})$ we choose the set D_j to be included in R.

This is denoted as:

$$R = (\bigcup_{S_j \in \mathcal{C}} A_j) \cup (\bigcup_{S_j \in (\mathcal{S} - \mathcal{C})} D_j)$$

We note that the following: Firstly, let $i \in U$. Since $\mathcal{C}$ is a cover there exists a j such that $S_j \in \mathcal{C}$ and $i \in S_j$. Hence, by the construction of MP the area A_j^i exists in MP. As $S_j \in \mathcal{C}$ we have that $A_j \subseteq R$ and in particular $A_j^i \in R$.

Secondly, let $D_j^r \in D$. If $S_j \notin \mathcal{C}$ the set R contains D_j and therefore also D_j^r. Otherwise R contains A_j and therefore also the two areas in A_j that are higher than D_j^r.

So R satisfies both condition of remark 1 and is hence an update solution for MP.

We recall from the MP-construction that for every set S_j there is a set A_j and a set D_j such that $|A_j| = |D_j| + 1 = |S_j|$. So,

$$|R| = \sum_{S_j \in \mathcal{C}} |S_j| \; + \sum_{S_j \in \mathcal{S} - \mathcal{C}} (|S_j| - 1)$$

$$= |\mathcal{C}| + \sum_{S_j \in \mathcal{C}} (|S_j| - 1) \; + \sum_{S_j \in \mathcal{S} - \mathcal{C}} (|S_j| - 1)$$

$$= |\mathcal{C}| + \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$$

This leads to the following lemma:

Lemma 3. *Let $\mathcal{C}$ be a cover of MC. Then there exists an update solution R for MP with $|\mathcal{C}| \leq |R| + \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$.*

5 NP-Hardness proof

We have shown so far how an instance MP of the Maximal Point Verification problem can be constructed out of an instance MC of the Minimum Set Cover problem, how to get a corresponding solution from one problem to the other and how the sizes of the solutions are related. We now argue that an optimal update solution corresponds to a minimal cover.

Lemma 4. *Let R be an optimal update solution for MP. Then the cover $\mathcal{C}$ constructed out of R is a minimal cover for MC.*

Proof. Let's assume there exists a cover $\overline{\mathcal{C}}$ for MC such that $|\overline{\mathcal{C}}| < |\mathcal{C}|$.

Let $\overline{R}$ be the update solution for MP constructed from $\overline{\mathcal{C}}$ as shown in Section 4. Then by Lemmas 2 and 3 we have that

$$|\mathcal{C}| \leq |R| - \sum_{S_j \in \mathcal{S}} (|S_j| - 1)$$

and

$$|\overline{\mathcal{C}}| = |\overline{R}| - \sum_{S_j \in \mathcal{S}} (|S_j| - 1).$$

Since $|\overline{\mathcal{C}}| < \mathcal{C}$ so must $|\overline{R}| < |R|$. This is a contradiction as R was a minimal update solution. So, $\mathcal{C}$ must be a minimal cover of MC.

We are using the established results to prove theorem 1.

Proof. In Section 3 we have presented the construction of a MPV problem for a given MSC problem. As noted in Section 3 the size of the MPV problem is polynomial in the size of the MSC problem and the construction can be done in polynomial time.

By Lemma 4 a solution of the MPV can be used to construct a solution of the MSC problem. As remarked in Section 4 that construction is polynomial in the size of the MPV problem.

Hence, if the MPV problem is solvable in polynomial time, then this must also be the case for the MSC problem. By [8], the MSC problem is shown to be NP-hard. So, also the MPV problem is NP-hard.

References

1. Richard Bruce, Michael Hoffmann, Danny Krizanc, and Rajeev Raman. Efficient update strategies for geometric computing with uncertainty. *Theory of Computing Systems*, 38(4):411–423, 2005.
2. Thomas Erlebach, Michael Hoffmann, Danny Krizanc, Matús Mihalák, and Rajeev Raman. Computing minimum spanning trees with uncertainty. In Susanne Albers and Pascal Weil, editors, *STACS*, volume 1 of *LIPIcs*, pages 277–288. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008.
3. T. Feder, R. Motwani, L. O'Callaghan, C. Olston, and R. Panigrahy. Computing shortest paths with uncertainty. *Journal of Algorithms*, 62(1):1–18, 2007.
4. T. Feder, R. Motwani, R. Panigrahy, C. Olston, and J. Widom. Computing the median with uncertainty. *SIAM Journal on Computing*, 32(2):538–547, 2003.
5. Manoj Gupta, Yogish Sabharwal, and Sandeep Sen. The update complexity of selection and related problems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011)*, volume 13 of *LIPIcs*, pages 325–338. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.
6. Simon Kahan. A model for data in motion. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC'91)*, pages 267–277, 1991.
7. Pegah Kamousi, Timothy M. Chan, and Subhash Suri. Stochastic minimum spanning trees in Euclidean spaces. In *Proceedings of the 27th Annual ACM Symposium on Computational Geometry (SoCG'11)*, pages 65–74. ACM, 2011.
8. Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.