# Assigning Sensors to Missions with Demands

Amotz Bar-Noy[1,2] Theodore Brown[1,3] Matthew P. Johnson[1]
Thomas La Porta[4] Ou Liu[1] Hosam Rowaihy[4]

[1] The Graduate Center of the City University of New York
[2] Brooklyn College, City University of New York
[3] Queens College, City University of New York
[4] Pennsylvania State University

**Abstract.** We introduce SEMI-MATCHING WITH DEMANDS (SMD), which models the problem in sensor networks when individual sensors must be assigned to sensing tasks. If there are multiple sensing tasks or *missions* to be accomplished simultaneously, and if sensor assignment must be exclusive, then this is a bipartite semi-matching problem. Each mission is associated with a demand value and a profit value; each sensor-mission pair is associated with a utility offer (possibly 0). The goal is a sensor assignment that maximizes the profits of the satisfied missions (with no credit for partially satisfied missions). SMD is an **NP**-Complete problem which is as hard to approximate as MAXIMUM INDEPENDENT SET. Therefore we investigate less difficult constrained versions of the problem. We give a simple greedy $\Delta$-approximation algorithm for a degree-constrained version ($\Delta$-SMD), in which each mission receives positive utility offers from at most $\Delta$ sensors. For small $\Delta$, we show that $\Delta$-SMD is equivalent to $k$-SET PACKING (with $k = \Delta$), which yields a polynomial-time $(\Delta+1)/2$-approximation. For $\Delta = 2$, we solve the problem optimally by reduction to maximum matching. Finally, we introduce a geometric version which remains strongly **NP**-Complete but has a PTAS.

## 1 Introduction

A sensor network consists of a large number of small sensing devices that are able to collect information about their surroundings. When a sensor network is deployed in the field it may be tasked with achieving multiple, possibly conflicting, missions. Hence, schemes that match sensor resources to mission demands become necessary. If there are multiple sensors and multiple missions, we must choose the best matching of sensors to missions. A given sensor may offer different missions varying amounts of information (because of geometry, obstructions, or remaining battery level, for example), or none at all.

Missions may vary in both importance (*profit*) and difficulty (*demand*), and these properties need not be correlated. An ongoing surveillance mission may be expensive but only modestly important, whereas an urgent mission for information about one particular spot may be low-demand but very important. In many applications, partial satisfaction will be no better than zero satisfaction. If the goal of a given mission is to reconstruct the 3D shape of an object, for example, then this may be accomplished with images from two cameras, but an image from just one camera will be useless. Indeed, accepting the single image could actually be harmful since the drain on the sensor's battery could preclude a future mission that might otherwise have been satisfiable. In our model we only receive profit, therefore, from missions whose demands are fully met. Hence the problem is to choose the "best" assignment of sensors to missions, in the sense that profits from satisfied missions are maximized.

In this paper, we define the sensor-mission matching problem and show that it is not just **NP**-Complete but extremely hard to approximate. As a result, we suggest constrained versions for which approximation algorithms do exist. In the first one, we bound the number of sensors that may offer contributions to any single mission. This is a reasonable assumption in realistic settings in which sensors have a limited sensing range and the sensors are distributed in such a way as to limit sensing redundancy. Indeed, covering an entire field using as few sensors as possible is an important problem in sensor networks (see Section 2). In the second constrained version, we assume that sensors and missions are located at points in the plane or higher-dimensional metric space; we also assume a bounded sensing range and a bounded density of sensors and missions. These are reasonable assumptions for realistic settings in which sensors are physical objects that exist in the world and that take up space, and that missions can be similarly localized.

The rest of this paper is organized as follows. Section 2 discusses some related work in sensor networks and in assignment problems. In Section 3 we formally define the sensor assignment problem and study its computational complexity. In Section 4, we introduce the degree-bounded version. We give a $\Delta$-approximation greedy algorithm for degree $\Delta$ and an efficient optimal algorithm for degree 2. We also show that the problem is equivalent to $\Delta$-Set Packing, which yields a $(\Delta + 1)/2$-approximation. In Section 5 we introduce a geometric version and give a shifting-based PTAS. Finally, Section 6 concludes the paper.

## 2 Related work

**Sensor networks.** The general problem of choosing sensors to achieve an objective has received sizable attention lately. Several objectives of selection have been considered. In [19, 21], for example, the authors look at solving the coverage problem using the fewest sensors in order to conserve energy. The techniques used range from dividing the nodes in the network into a number of sets and rotating them, activating one at a time [19], to using Voronoi diagram properties [2] to ensure that the area of a node's diagram is as close to its sensing area as possible [21]. Our work, however, focuses on contention between multiple missions with varying profit values.

Sensor selection schemes have also been proposed to efficiently track and locate targets, using several techniques. In [23], for example, the most informative sensor for tracking a target is chosen based on the concept of information gain. This information is then passed on to the next active node which is chosen by considering the expected target's path. The problem of target localization using acoustic sensors is considered is considered by [14]. The goal is to minimize the mean squared error of the target location as perceived by the active nodes.

There has also been some work in defining frameworks of the single and multiple mission assignment problem which relates to what we examine here. For example, [4] defines a framework for modeling the assignment problem using notions of utility and cost. The goal is to find a solution that maximizes the utility while staying under a predefined budget. In [17], a market-based approach is used in which sensors provide information or "goods" and each sensor has a certain budget. Our problem differs in two important respects. First, we maximize the profits uncategorically; the only budget is the available sensors. Second, we do not simply maximize total received utility. Solution quality for our problem is more strict; it is based on the end result of satisfying missions.

**Algorithms.** Although we use the terminology of sensors and missions for concreteness, SMD can be viewed as a more general problem of resource allocation. An alternative interpretation regards scheduling jobs on unrelated parallel machines. As in other (maximization) scheduling problems [22], the goal is a schedule that maximizes profit earned from jobs completed, subject to certain constraints. The twist is that each job specifies not the set of *machines* that can perform it, but the set of *families of machines* that can perform it. (A job may be too difficult to be performed by any single machine.) The feasibility constraint is that no machine can be assigned more than one "sub-job".

We now relate SMD to the Generalized Assignment Problem (GAP) [7], which is a generalization of the well known Bin Packing problem. Like in Bin Covering, bins in GAP have storage capacities and the task is to maximizes profits earned from storing items in the bins; the difference is that the amount of space required to store a given item varies from bin to bin. Analogously, SMD can be viewed as a generalization of the less well known Bin Covering problem, in which the goal is to use the items to fill completely as many bins as possible. The generalization is again that the space taken (utility given) by an item (sensor) depends on the bin (mission) it is assigned to.

Many weaker and often fractional models of sensor-mission matching can be reduced to maximal matching or network flow problems, and thus can be solved optimally in polynomial time [1]. A survey of the different sensor selection and assignment schemes including simple theoretical models of the problem can be found in [20].

## 3 Problem definition

Given is a complete weighted bipartite graph, whose vertex sets consist of sensors $S = \{S_1, ..., S_n\}$ and missions $\{M_1, ..., M_m\}$. A positively weighted edge $(S_i, M_j)$ means that $S_i$ is applicable to $M_j$. The weight of the edge $(e_{ij})$ indicates the utility (or quality of information) that $S_i$ could contribute to $M_j$ if this pairing were chosen. Also given is a positive-valued demand $d_j$ associated with each mission $M_j$, indicating the total utility the mission requires. What we seek is a *semi-matching* of sensors to missions, so that (ideally) each mission demand is satisfied. That is, a sensor may be assigned to at most one of the missions to which it is applicable, but it is legal for a mission to accept utility from multiple sensors. Of course, satisfying all missions may not be feasible; in general, the goal is to maximize a weighted sum of the *satisfied missions*. We assume there is a profit $p_j$ associated with achieving mission $M_j$. We then seek to maximize the total satisfied profit. Note that there is no profit awarded for a partially satisfied mission in this model.

Unless there is structure in the weights of the sensor-mission edges, for example if they relate to the geometry of node positions, we can assume without loss of generality that each demand is 1. For each mission $M_j$ with demand $d_j$, simply divide edge value $e_{ij}$ by $d_j$ to obtain an instance with unit demands. Unless otherwise stated, we will assume this normalization henceforth, though it is sometimes convenient to allow for non-unit demands. With this in mind, we define the problem formally.

**Instance:** A weighted bipartite graph $G = (S, M, P, E)$, where $S = \{S_1, ..., S_n\}$ is a collection of sensors, $M = \{M_1, ..., M_m\}$ is a collection of missions, $P = \{p_1, ..., p_m\}$ is a collection of positive mission profits, and $E$ is a collection of non-negative weights for the edges $S \times M$.

**Goal:** Find a semi-matching $F \subseteq E$ (no two chosen edges share the same *sensor*), in which $\sum_{M_j \in A} p_j$ is maximized, where $A \subseteq M$ is the set of missions satisfied by $F$ (i.e., $\sum_{(i,j) \in F} e_{ij} \geq 1$).

It may be easier to understand the problem in its Integer Programming formulation. The formulation below employs two sets of decision variables: $y_j$, indicating whether mission $M_j$ is satisfied, and $x_{ij}$ indicating whether sensor $S_i$ is assigned to mission $M_j$. Finding a solution can be seen as a two-step process: decide which missions to satisfy, and then decide how to satisfy them. Each mission $M_j$ has a constraint requiring that the sum of utility received by $M_j$ be at least the value $y_j$, which is 0 or 1. When $y_j = 0$, this constraint is automatically satisfied.

**Maximize:** $\sum_j p_j y_j$
**Such that:** $\sum_{i=1}^{n} x_{ij} e_{ij} \geq y_j$, for each mission $M_j \in A$,
$\sum_{j=1}^{m} x_{ij} \leq 1$, for each sensor $S_i$, and
$x_{ij} \in \{0, 1\}$, for each variable $x_{ij}$ and $y_j \in \{0, 1\}$, for each variable $y_j$

Note that if we had not normalized to unit demands, the first constraint would be: $\sum_{i=1}^{n} x_{ij} e_{ij} \geq y_j d_j$, for each mission $M_j \in A$. The corresponding linear program relaxes the constraint on each variable from $\{0, 1\}$ to $[0, 1]$.

*Remark 1.* This IP has unbounded integrality gap, since instances can be constructed in which $OPT_{LP} = m/2$ and $OPT_{IP} = 1$, where $m$ is the number of missions. To create such an instance (see Fig. 1), introduce and connect a separate sensor to each *pair of missions*, so that each mission has $m - 1$ neighbors, and set all demands to $m - 1$ and all profits to 1. Then setting all edge weights to $1/2$ will clearly half-satisfy each mission, but only one can be satisfied integrally.

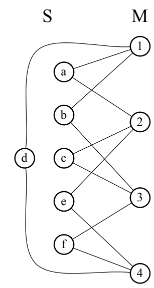


**Fig. 1.** Integrality gap instance.

A relaxed version of this problem, in which profits are awarded fractionally for partial satisfaction *and* sensors can be assigned fractionally to multiple missions, can be solved with this formulation by Linear Programming, and certain versions can be solved by reduction to MAX-FLOW (see [20]), but this version cannot.

**Proposition 1.** *SMD is* ***NP****-Complete and at least as* hard to approximate *as* MAXIMUM INDEPENDENT SET *(*MIS*).*

*Proof.* We do a gap-preserving reduction from the MIS (see Fig. 2). Let $G = (V, E)$ be a graph, with $|V| = n$, $|E| = m$. Our resulting problem instance has a mission $M_j$ corresponding to each vertex $v_j$, and a sensor $s_{ij}$ corresponding to each edge $e_{ij}$. Each sensor $s_{ij}$ will be connected to missions $M_i$ and $M_j$, each with a utility of 1. The demand $d_j$ of mission $M_j$ is set to $deg(v_j)$, and all mission profits are set to 1. (Notice that for any isolated vertex, the corresponding mission will have 0 demand.) We include vertex $v_j$ in the resulting maximum independent set iff mission $M_j$ is satisfied. Because profits are set to unity, it follows that the maximum independent set size will equal the maximum possible profit. To see that the reduction is gap-preserving, simply set $f_1(x) = f_2(y) = m$ and $\alpha = \beta = 1$ in the standard *gap-preserving reduction* definition. Finally, for **NP**-Completeness, it is easy to verify positive instances of the corresponding decision problem. QED

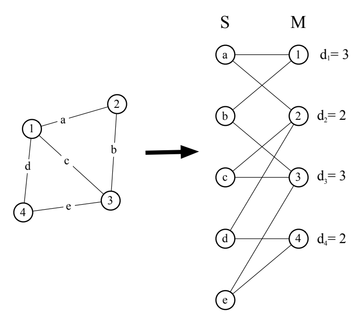


**Fig. 2.** Reduction of MAXIMUM INDEPENDENT SET to SMD.

Because of this reduction, known hardness results for MIS also apply to SMD. MAXIMUM INDEPENDENT SET is the same as MAXIMUM CLIQUE on the complement graph, and MAXIMUM CLIQUE is known to be hard to approximate within $|V|^{1-\epsilon}$ for any $\epsilon > 0$, unless **NP=ZPP** (and hard within $|V|^{1/2-\epsilon}$, even

**Algorithm 1** $\Delta$-Approximation Greedy Algorithm

```
for each mission M_j in order of decreasing P_j do
  for each still-available sensor S_i in order of decreasing e_ij do
    assign S_i to M_j
    if M_j is satisfied then
      break
    end if
  end for
  if M_j is not satisfied then
    return any sensors assigned to it
  end if
end for
```

without this assumption) [9]. This means that the best achievable approximation ratio can be little better than $m$, i.e., satisfying *only about one mission out of* $m$, which can be done by simple greedy algorithms.

We briefly note two special cases of this problem. The hardness properties above apply to both special cases.

**Profits = demands:** Set $p_j$ to the original (pre-normalization) demand $d_j$.
**Cardinality:** Set $p_j = 1$, in which case the objective function is simply the number of satisfied missions.

## 4 Degree-bounded approximation problem

Because of the difficulty of the approximation problem as defined, we will constrain it in order to render it more tractable. Let $|OPT|$ indicate an optimal solution to a given problem instance, and let $|ALG|$ indicate a corresponding approximate solution. We will say that an algorithm is a *c-approximation* for $c \geq 1$ if $c \leq \frac{|OPT|}{|ALG|}$ for every problem instance.

We will assume that the problem instance has *bounded degree*, in the following sense. If a sensor $S_i$ makes a non-zero offer to a mission $M_j$, then say that $S_i$ is $M_j$'s *neighbor*. Then the assumption is that no mission has more than $\Delta$ neighbors, for some small constant $\Delta$. (If all zero-weight edges are removed, this is the same as saying $\Delta$ bounds the degrees of all mission nodes in the SMD graph.) We call this problem $\Delta$-SMD.

A simple greedy algorithm considers missions in decreasing order of profit. For each mission, the algorithm assigns it available sensors in decreasing order of offer utility, until the mission is satisfied. If the mission does not succeed, then all sensors are returned. Assuming that $m = O(n \log n)$, the running time is $O(mn \log n)$.

**Definition 1.** *Let a* star *consist of a mission and a* minimally satisfying *set of sensors for it. The sensor set is minimal in the sense that no proper subset of it would completely satisfy the mission in question. (Notice that a given mission* *may in general be part of many stars.) Say that a mission is* tight *if it has degree $\Delta$ and requires all $\Delta$ sensors in order to be satisfied. Two stars* overlap *if they share one or more sensors, if they share a mission, or both, including the case that the stars are identical.*

**Proposition 2.** *Algorithm 1 produces a $\Delta$-approximation.*

*Proof.* Let $OPT$ be the chosen set of missions in some optimal solution, and let $ALG$ be the missions chosen by Algorithm 1. We want to show that $OPT \leq \Delta \cdot ALG$, i.e., that

$$\sum_{M_j \in OPT} p_j \leq \sum_{M_{j'} \in ALG} \Delta \cdot p_{j'} \tag{1}$$

To prove Ineq. 1, we account for each term $p_j$ on the LHS with one of the terms $\Delta \cdot p_{j'}$ on the RHS. For each $M_j \in OPT$, say that $M_j$ *charges to* the highest-profit mission $M_{j'} \in ALG$ whose star overlaps with $M_j$'s star, and write $M_j \in ch(M_j')$. (There must be one such $M_{j'}$.) Then let $M_{j'}$ be an arbitrary mission in ALG. $M_{j'}$ is either tight or not. Suppose tight, in which case that mission has only one star. If $M_{j'} \in OPT$, then only $M_{j'}$ itself charges to $M_{j'} \in ALG$; if $M_{j'} \notin OPT$, then at most $\Delta$ stars in $OPT$ can charge to $M_{j'}$ (those that share at least one of its sensors). Now suppose $M_{j'}$ is not tight, so that it contains $\leq \Delta - 1$ sensors. Then at most $\Delta$ stars in $OPT$ can charge to $M_{j'}$ (those that share at least one of its sensors, and possibly one that shares its mission). Thus we have

$$\sum_{M_j \in ch(M_j')} p_j \leq \Delta \cdot p_{j'} \tag{2}$$

By summing Ineq. 2 over all missions in $ALG$, we obtain Ineq. 1. QED

It is easy to construct an example with $\Delta + 1$ missions to show that the $\Delta$ bound is tight.

**Corollary 1.** *By the* Maximum Independent Set *reduction, $\Delta$-SMD (for $\Delta \geq 3$) is* ***APX****-hard [18]. Given the constant-factor-approximation of Algorithm 1, $\Delta$-SMD (for $\Delta \geq 3$) is* ***APX****-complete.*

**Proposition 3.** 2-SMD *is in* ***P****.*

*Proof.* We reduce to the (weighted) maximal matching problem (see Fig. 3). The node set for the resulting graph will consist of the 2-SMD instance's sensors and missions. Whenever a mission $M_j$ will be satisfied only by both its neighbors $S_{i1}, S_{i2}$, draw an edge $(S_{i1}, S_{i2})$ with weight equal to the mission profit; whenever a mission $M_j$ will be satisfied by a single sensor $S_i$, draw an edge $(S_i, M_j)$ with weight equal to the mission profit. Now find a maximal weighted matching in this (non-bipartite) graph in polynomial time. Each selected edge corresponds to a satisfied mission. It is clear that no sensor or mission will be used more than once. The optimal solution values of the matching graph and the SMD are by construction the same. QED

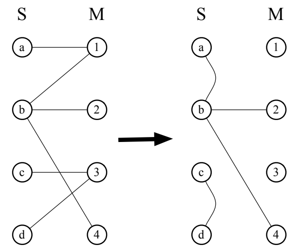


**Fig. 3.** Converting SMD to graph.

Since the graph of a 2-SMD instance is sparse, the maximal weighted matching can be found in time $O(m^2 \log m)$ [8], where $m$ is the number of missions. The time for the maximal matching is the dominant component of the running time.

We now relate $\Delta$-SMD and $\Delta$-Set Packing. These problems turn out to be equivalent for $\Delta$ "small enough". In the Set Packing problem, we are given a family of subsets of a universe of elements. Each subset has a positive weight. The goal is to choose a max-weight family of subsets without using any element twice. $\Delta$-Set Packing is the variant of Set Packing in which each set has at most $\Delta$ elements.[5]

**Proposition 4.** $\Delta$-SMD *reduces to* $\Delta$-Set Packing, *when* $\Delta = O(\log nm)$.

*Proof.* The idea of the reduction is that each star in our SMD instance will become a set in the Set Packing instance. Since a given mission may have degree $\Delta$, it can have $O(2^\Delta)$ many stars. Because of the bound on $\Delta$, however, the resulting Set Packing instance will be at most polynomially larger than the initial SMD instance size. We must also correct for missions that are tight. A naive reduction will map such missions to sets of size $\Delta + 1$. Since a mission can have only one tight star of size $\Delta$, however, the mission need not be included in the resulting star. Choosing a max-weight family of disjoint sets will now be the same as choosing a max-weight set of disjoint stars. It is interesting to note that the distinction between sensors and missions has now disappeared: no element can be used more than once. QED

**Proposition 5.** $\Delta$-Set Packing *reduces to* $\Delta$-SMD.

*Proof.* Each element in the $\Delta$-Set Packing universe will correspond to a sensor in the resulting $\Delta$-SMD instance. For each set in the $\Delta$-Set Packing instance, we create a separate mission that requires all the sensors in this set in order to be satisfied. Then each mission has degree at most $\Delta$ by construction. QED

For small enough $\Delta$ relative to problem instance size, $\Delta$-Set Packing and $\Delta$-SMD are equivalent. Therefore for such $\Delta$, $\Delta$-SMD can be converted from a covering problem to a packing problem. An existing local search algorithm from Berman [3] gives a $(\frac{\Delta+1}{2}+\epsilon)$-approximation for $\Delta$+1-Claw-Free MIS,

[5] The parameter used is typically $k$, but we are interested in the case $k = \Delta$.

which $\Delta$-Set Packing reduces to. ($\epsilon$ is a running-time parameter, specifically, a $\frac{k}{(k-1)}\frac{\Delta+1}{2}$ approximation can be found in time polynomial in $O(kn)$.) Hence there is a $\frac{\Delta+1}{2}$ approximation for $\Delta$-SMD (for small $\Delta$). It was recently shown [10] that even for the cardinality version, approximating $\Delta$-Set Packing within a factor better than $\frac{\Delta}{\ln \Delta}$ is **NP**-hard.

Following Jain et al.'s LP for Facility Location [13], we can define a simpler IP formulation in terms of *stars*. When $\Delta = O(\log nm)$, there will only be polynomially many stars for a single problem instance. For a given element $A$ (either sensor or mission), there will certainly be at most polynomially many stars containing $A$. In the following IP, decision variable $y_t$ indicates that we choose star $t$; we have a constraint for each element. Intuitively, this IP has the advantage that it has just one set of decisions to make: which stars to pick? The profit for a star is simply the profit for the mission it includes; each sensor or mission can be used at most once.

**Maximize:** $\sum_t p_t y_t$
**Such that:** $\sum_{R_t:A_s \in R_t} y_t \leq 1$, for each sensor or mission $A_s$ and
$y_t \in \{0,1\}$, for each variable $y_t$

*Remark 2.* This IP has integrality gap at least $\frac{\Delta+1}{2}$, since instances can be constructed in which $OPT_{LP} = \frac{\Delta+1}{2}$ and $OPT_{IP} = 1$, where $m$ is the number of missions. In fact, $\frac{\Delta+1}{2}$ is also a lower bound on the integrality gap of the first IP formulation, in the case of bounded degree.

## 5 Geometric approximation problem

We now introduce Geometric SMD (or GeomSMD), in which all nodes (i.e., sensors and missions) lie in the plane (or higher-dimensional space) and geometrically inspired constraints are imposed. First, each sensor and mission now lie at a particular point in the plane. Second, we assume sensors have a bounded sensing range, i.e., $e_{ij}$ can only be non-zero when the distance between $S_i$ and $M_j$ is less than this range. (Without loss of generality, let the sensing range be 1. In this case, every star will lie in a unit disk.) We also assume a geometric analog to bounded degree, specifically an upper bound on the number of sensors or missions contained in any unit disk. This constraint will be satisfied automatically if the graph is *drawn in a civilized manner* [12], i.e, any two nodes are separated by some minimum global distance $\lambda > 0$. Hence GeomSMD is a special case $\Delta$-SMD for some $\Delta$.

The **NP**-hardness argument will involve the Unit-Disk Maximum Independent Set (UD-MIS) problem [5]. That is the variant of MIS in which the problem instance is the *intersection graph* of a set of unit disks lying in the plane. Equivalently, UD-MIS can be defined so that given a set of points in the plane, two points are connected by an edge iff their distance is strictly less than a global constant. We will argue that the **NP**-hardness proof for UD-MIS also applies to a density-bounded UD-MIS. The **NP**-hardness proof for UD-MIS from [5] is recounted in [15].

**Algorithm 2** Shifting PTAS (error $\epsilon$)

$k \leftarrow \lceil 2/\epsilon \rceil$; $S = \emptyset$
**for** each $(i, j) \in [0, k)^2$ **do**
  lay the mesh with offset $(i, j)$; $S_{ij} \leftarrow \emptyset$
  **for** each cell $C_t$ within the mesh **do**
    $S_{ij} \leftarrow S_{ij} \cup opt(C_t)$
  **end for**
  **if** $val(S) < val(S_{ij})$ **then**
    $S \leftarrow S_{ij}$
  **end if**
**end for**

**Proposition 6.** GeomSMD *is strongly* ***NP****-Complete.*

*Proof.* We reduce from 3SAT to UD-MIS [5, 15] to GeomSMD. Given the 3SAT instance, first apply the known UD-MIS reduction, which results in a UD-MIS graph and a number $k$ (there is an independent set of size $k$ iff the formula was satisfiable). It is clear that the resulting UD-MIS instance can be drawn with at most O(1) disk per unit square (by inspection, at most 4 disks intersect at any point, and squeezing a chain of 3 disks so that all 3 centers lie in a unit disk will introduce a new edge). There are $O(\#vars \cdot \#clauses)$ disks, clearly polynomial in the 3SAT size.

Now take this UD-MIS decision-problem instance $(G, k)$ and convert it into a GeomSMD decision-problem instance $(G', k)$, by replacing each disk with a mission at the disk's center, and every maximal intersection of disks with a sensor needed by all of them. Since each mission needs all the sensors lying in its disk in order to be satisfied, $k$ missions can be satisfied iff $k$ independent disks can be chosen. Since in the UD-MIS construction each unit square contains at most $O(1)$ such intersections, in the resulting GeomSMD instance, each unit square will contain at most $O(1)$ missions and $O(1)$ sensors, and the sensing distance is respected by construction. Thus we have reduced 3SAT to an instance of GeomSMD. QED.

Since 3SAT is strongly **NP**-Complete, it follows that an FPTAS is impossible for GeomSMD. A PTAS, which we now give, is the best that can be hoped for. We employ the shifting technique introduced by Hochbaum & Maass [11]. Within a $c \times c$ cell, there will be at most $O(\Delta c^2)$ sensors and missions, for some constant $\Delta$. As $c$ increases, the area crossing an edge will be decrease. For $3 \times 3$ cells, for example, the proportion inside with be 4/9. For a $c \times c$ cell, the internal portion can be solved brute-force in time exponential in $c$ and $\Delta$, but polynomial in the problem instance size $nm$.

We now give a PTAS,[6] employing the shifting used by Unit-Disk Maximum Independent Set [16] (following the presentation in [6]). For now, assume for simplicity that all points are bounded by a square region $I$ of size polynomial

[6] Although we focus on the plane, it is easy to extend to a fixed higher dimension $D$.

in the input size. Then for a desired error bound $\epsilon$, we can choose $c = \lceil 2/\epsilon \rceil$. Now lay a grid on the plane with integer coordinates and cells of size $c \times c$. Each $(i, j) \in [0, c)^2$ corresponds to a possible offset for the grid. For a given grid position, we eliminate all sensor-mission edges that are not fully contained within a single cell. Within any cell, there are $O(\Delta c^2)$ sensors and missions; therefore we can find the optimal assignment by enumerating the $O((\Delta c^2)^{\Delta c^2})$ possible assignments within a single cell. The solution for a given offset pair $(i, j)$ is the union of the solutions for the individual cells. We compute the solution for each possible offset pair.

We now justify our initial assumption. If the points lie in an extremely large region, then the method as stated may not run in polynomial time, since there may be exponentially many cells to check. This can be easily fixed. First, notice that there will be at most polynomially many non-empty cells. These can be found by iterating through the point coordinates. For each non-empty cell, we can "grow" it outward, to obtain a maximally non-empty region. Performing this action on every non-empty cell (i.e., Union-Find) produces a polynomial collection of independent regions. Now simply run the original algorithm on the union of these independent regions, rather than on the entire space.

**Proposition 7.** *Algorithm 2 is a PTAS.*

*Proof.* Consider the optimal star-set OPT with total profit $P_{opt}$. By the Shifting Lemma [11], there must be some vertical offset $j$ that crosses a subset of OPT with total profit at most $P_{opt}/c$. Similarly, there must be some horizontal offset $i$ that crosses a subset of OPT with total weight at most $P_{opt}/c$. Therefore the union of the cell optimal solutions for this $(i, j)$ will be within factor $1 - (2c - 1)/c^2 = O(1 - 2/c) = O(1 - \epsilon)$ of the optimal. QED

## 6 Conclusion

In this paper, we introduced a sensor-mission matching problem. We analyzed its complexity, defined constrained versions, and presented approximation algorithms for them. There are many open problems, such as:

- Seek efficient combinatorial algorithms for *GeomSMD* with constant approximation ratio.
- Seek LP-based $(\Delta + 1)/2$ approximations than run in bounded time.
- Close the approximation gap for $\Delta$-SMD between $\Delta/\ln\Delta$ and $(\Delta + 1)/2$.

## References


1. R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows.* Prentice Hall, 1993.
2. F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 1991.
3. P. Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. In *Proceedings of SWAT*, pages 214–219, 2000.

4. J. Byers and G. Nasser. Utility-based decision-making in wireless sensor networks. In *Proceedings of the IEEE Workshop on Mobile and Ad Hoc Networking and Computing*, 2000.
5. B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Math*, 86:165–177, 1990.
6. T. Erlebach and J. Fiala. Independence and coloring problems on intersection graphs of disks. *manuscript*, 2001.
7. L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of SODA 06*, pages 611–620, 2006.
8. Z. Galil, S. Micali, and H.N. Gabow. An O(EV log V) algorithm for finding a maximal weighted matching in general graphs. *SIAM J. Comput.*, 15(1):120–130, 1986.
9. J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.
10. E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating k-set packing. *Computational Complexity*, 15(1):20–39, 2006.
11. D.S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *J. ACM*, 32(1):130–136, 1985.
12. H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.. Stearns. Nc-approximation schemes for np- and pspace-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.
13. K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V.V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *Journal of the ACM*, 50(6):795–824, 2003.
14. L. Kaplan. Global node selection for localization in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):113–135, January 2006.
15. M.V. Marathe, V. Radhakrishnan, H.B. Hunt III, and S.S. Ravi. Hierarchically specified unit disk graphs. *Theor. Comput. Sci.*, 174(1-2):23–65, 1997.
16. T. Matsui. Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In *JCDCG*, pages 194–200, 1998.
17. T. Mullen, V. Avasarala, and D. L. Hall. Customer-driven sensor management. *IEEE Intelligent Systems*, 21(2):41–49, Mar/April 2006.
18. C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.
19. M. Perillo and W. Heinzelman. Optimal sensor management under energy and reliability constraints. In *Proceedings of the IEEE Conference on Wireless Communications and Networking*, March 2003.
20. H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta. A survey of sensor selection schemes in wireless sensor networks. In *SPIE Defense and Security Symposium Conference on Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, 2007.
21. K. Shih, Y. Chen, C. Chiang, and B. Liu. A distributed active sensor selection scheme for wireless sensor networks. In *Proceedings of the IEEE Symposium on Computers and Communications*, June 2006.
22. S. C. Sung and M. Vlach. Maximizing weighted number of just-in-time jobs on unrelated parallel machines. *J. Scheduling*, 8-5:453–460, 2005.
23. F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.