Optimal Placement of Stereo Sensors[1]

Mohammad Al Hasan
Department of Computer Science
alhasan@cs.rpi.edu

Krishna K. Ramachandran
Department of Electrical, Computer and Systems Engineering
ramak@rpi.edu

John E. Mitchell
Department of Mathematical Sciences
mitchj@rpi.edu
http://www.rpi.edu/~mitchj

Rensselaer Polytechnic Institute
Troy, NY 12180

September 29, 2006.
Revised: December 29, 2006

Abstract

Autonomous wireless devices such as sensor nodes and stereo cameras, due to their low cost of operation coupled with the potential for remote deployment, have found a plethora of applications ranging from monitoring air, soil and water to seismic detection and military surveillance. Typically, such a network spans a region of interest with the individual nodes cooperating to detect events and disseminate information. Given a deployment of sensors and targets over a region, a sensor pairing is desired for each target that optimizes the coverage under certain constraints. This problem can be modeled as an integer programming problem and solved using branch-and-cut. For larger problems, it is necessary to limit the number of variables, and a GRASP routine was developed for this purpose. Valid cutting planes are developed and computational results presented.

Keywords: Sensors, integer programming, branch-and-cut, GRASP

[1]Research supported in part by NSF grant numbers DMS–0317323 and CMS–0301661

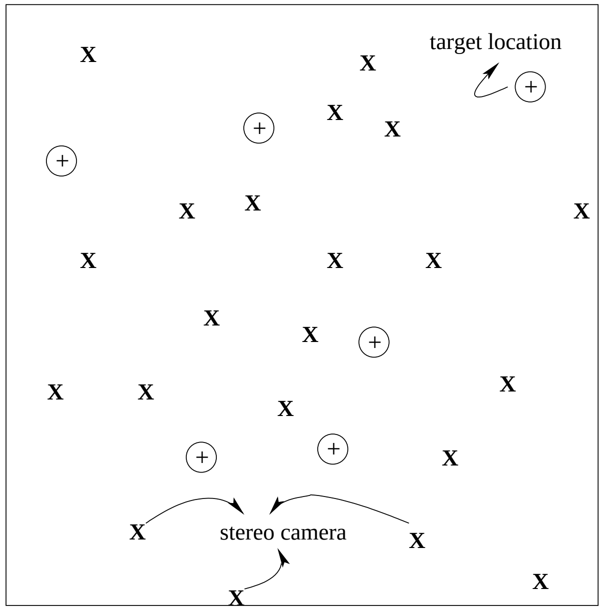

Figure 1: An example topology of sensor nodes deployed for target detection.

1 Introduction

Autonomous wireless devices such as sensor nodes and stereo cameras, due to their low cost of operation coupled with the potential for remote deployment, have found a plethora of applications ranging from monitoring air, soil and water to seismic detection and military surveillance [2, 13, 16, 17]. Typically, such a network, as shown in Figure 1, representing a dense deployment of wireless nodes, spans a region of interest with the individual nodes cooperating to detect events and disseminate information.

Existing research has primarily concentrated on developing algorithms, be it distributed or centralized, to optimize network longevity metrics [3]. This is due to the fact that the physical constraints of battery-powered sensors impose limitations on their processing capacity and longevity. As battery power in the nodes decays, certain parts of the network may become disconnected or the coverage may shrink, thereby reducing the reliability and the potency of the sensor network. Given such a scenario, it is desirable that the sensors are deployed in a manner facilitating optimization of network properties such as coverage and connectivity. In particular, there exists a class of problems, termed topology control problems, that address the assignment of power values to the nodes of an ad hoc network so as to result in a graph topology satisfying certain specified properties [11].

In the current work, we consider a variant of the topology control problem wherein random deployment of stereo cameras over a region is carried out with the objective of assigning the optimal pair for each stationary target contingent on certain constraints. While there is a general consensus that random deployment of nodes yields poor performance, analytical studies characterizing the performance bounds attainable have not been addressed in the literature. In many cases, one might be constrained to opt for a random deployment and not have control over node placement. For instance, when sensors are deployed under extreme conditions such as fire, rain etc., one can at best hope for a random dispersal of nodes over the region to be monitored. Under such circumstances, it is imperative that the researcher has an estimate on the number of sensors to be deployed in order to establish desired network properties. This provides our motivation for seeking to characterize the performance of such random node deployments.

Battery power is a prime resource in sensor networks, and needs to be conserved in order to prolong the connectivity as well as the coverage of the network. Optimal power consumption and its variants form the central theme of the bulk of the current research on sensor networks. The objective of the current work is twofold: to study the feasibility of obtaining a coverage of target locations subject to certain constraints and to propose optimization frameworks for random deployment of sensors addressing the constraints such as minimizing sensor movement and number of sensors moved while ensuring coverage of all the targets.

The rationale behind the feasibility study is that attaining satisfactory bounds on random deployment can preclude the need for complex distributed or centralized algorithms to be executed at the sensors. This in turn reduces the power consumption of the sensor nodes. Also, the choice of minimizing the total movement of sensors stems from the fact that power expenditure is directly proportional to the distance moved. Here, we assume that the target is stationary while the sensor cameras have the ability to move around in the deployed region. Consider the following scenario: sensors are deployed randomly over a region with the objective of detecting certain events of interest, termed targets, whose coordinates are known; the objective then is to cover each event (target) with minimal movement of the sensor nodes. A target is said to be covered if there exist at least 2 sensors that meet user defined constraints on the radial distance from the target and the angle subtended by them at the target. The problem statement and the constraints imposed on the sensor coverage can then be formulated as follows:

Problem Statement: Given parameters $d_1 \leq d_2$ and θ, and a deployment

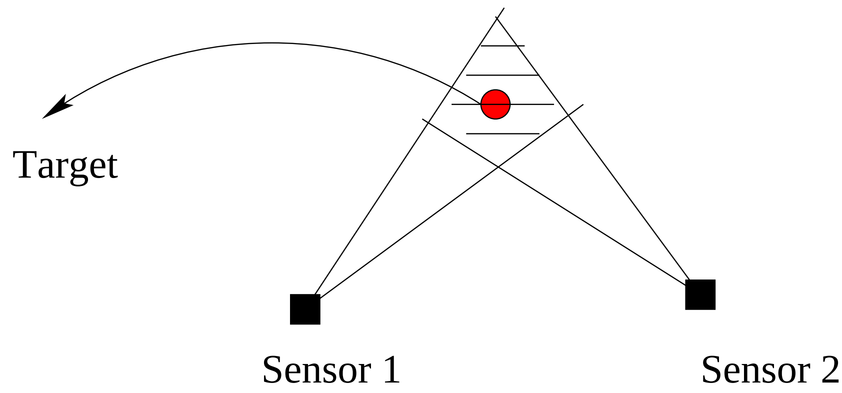

Figure 2: Target estimation scenario.

of s sensors and t targets ($s \geq 2t$) over a region, obtain a sensor pairing for each target that optimizes the coverage under the following constraints:

For each target position x, assign a unique pair of sensors and place them at positions s_1 and s_2 such that

$$d_1 \leq d(x, s_i) \leq d_2 \text{ for } i = 1, 2 \tag{1}$$

where $d(x, s_i)$ denotes the Euclidean distance between x and s_i. Further, position the two sensors so that the angle $\phi(s_1, x, s_2)$ defined by the target and the two sensors satisfies

$$\phi(s_1, x, s_2) \leq \theta \tag{2}$$

We now elaborate on the formulation of the coverage requirements. The line of sight of a sensor is best characterized by an unbounded conical region. As a result, a single sensor can only detect the presence or absence of a target but not its location. The position estimation of a target location, therefore is determined by the area of intersection of two or more sensors, as shown in Figure 2. Indeed it is desirable for the sensors to position themselves in a manner that minimizes this intersecting area in which the target is located. This is so because the smaller the area, lesser is the error in estimation. Two factors that are key to the estimation process are the radial distances of the sensors and their angular orientation with the target. For example, the conical sensing area increases with increasing radial distance, and this coupled with the sensors subtending an obtuse angle at the target would translate to a high estimation error, since the area of intersection in which the target can be located is large. Thus, while determining coverage requirements, it is imperative that

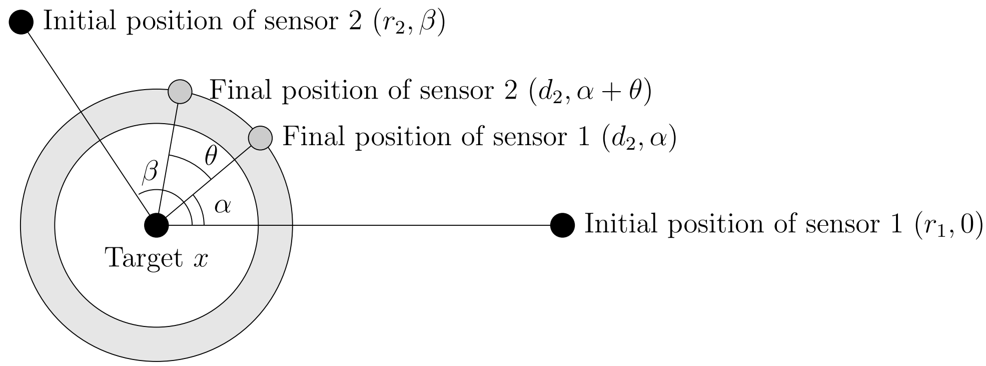

Figure 3: Two sensors and a target. Both sensors must be moved into the annulus, and they must be separated by no more than θ. The inner circle has radius d_1 and the outer circle has radius d_2. The positions are expressed in polar coordinates.

these concerns are addressed and this is the motivation behind the formulation of constraints (1) and (2).

In this paper, we formulate the problem of covering the targets with minimum total distance moved by the sensors as an integer programming problem. We can obtain exact solutions to problems with 50 targets in reasonable time. For larger problems, it is necessary to use heuristics to select good candidate sets of pairs of sensors for each target; the integer programming problem restricted to the corresponding set of variables can be solved effectively for problems with 200 targets. By examining problems with 50 targets, it appears that the heuristic methods for selecting sets of pairs are very effective and often lead to the optimal solution, especially if $s \geq 3t$.

2 Finding the cost of a pairing

Given a target position and two sensors, the placement of the two sensors that minimizes their total movement subject to (1) and (2) is found. To fix notation, the target x is regarded as the origin and all angles are measured from the line defined by x and the initial position of sensor 1. Assume without loss of generality that sensor 2 makes an angle β between 0 and π. This is illustrated in Figure 3.

The optimal cost of a pairing can be found by solving an optimization problem with a single variable, namely the angle α moved by sensor 1. The optimal value of α must lie between 0 and $\max\{0, \beta - \theta\}$. Choosing α corresponds to choosing a wedge to which both sensors must move, and the closest points in the wedge to the two sensors can then be found as follows. Let r_1 be the initial distance of sensor 1 from the target. For a given value of α, the best radius for sensor 1 is then

$$r_1'(\alpha) := \begin{cases} d_2 & \text{if } r_1 \cos\alpha \geq d_2 \\ d_1 & \text{if } r_1 \cos\alpha \leq d_1 \\ r_1 \cos\alpha & \text{otherwise} \end{cases}$$

If $\beta \leq \alpha + \theta$ then sensor 2 is moved radially to be within the annulus (if necessary). Otherwise, sensor 2 is moved to the angle $\alpha + \theta$, the quantity $r_2 \cos(\beta - \theta - \alpha)$ is calculated, and the best radius for sensor 2 is

$$r_2'(\alpha) := \begin{cases} d_2 & \text{if } r_2 \cos(\beta - \theta - \alpha) \geq d_2 \\ d_1 & \text{if } r_2 \cos(\beta - \theta - \alpha) \leq d_1 \\ r_2 \cos(\beta - \theta - \alpha) & \text{otherwise} \end{cases}$$

Thus, the positions of the sensors can be expressed in terms of the angle α, and the objective function is to minimize the sum of the distances moved by the two sensors. This one variable optimization problem was solved using a simple line search routine.

The objective function may be nonconvex in α. Nonetheless, it is quite flat, and so it is straightforward to find the optimal value to an acceptable accuracy. For example, the objective function is plotted in Figure 4 for the case $\theta = 0$, $d_1 = d_2 = 1$, $r_1 = 3$, $r_2 = 2$, and $\beta = 2.5$ radians. (Note that all angles in this paper are given in radians.)

3 Integer programming formulation

For each target i and each pair of sensors j and k, define the binary variable

$$x^i_{jk} = \begin{cases} 1 & \text{if target } i \text{ is covered by sensors } j \text{ and } k \\ 0 & \text{otherwise} \end{cases}$$

An integer programming formulation of the problem is as follows:

$$\begin{array}{llcll} \min & \sum_{i=1}^{t} \sum_{j=1}^{s-1} \sum_{k=j+1}^{s} c^i_{jk} x^i_{jk} & & & \\ \text{subject to} & \sum_{j=1}^{s-1} \sum_{k=j+1}^{s} x^i_{jk} & = & 1 & i = 1, \ldots, t \\ & \sum_{i=1}^{t} \sum_{k=1}^{j-1} x^i_{kj} + \sum_{i=1}^{t} \sum_{k=j+1}^{s} x^i_{jk} & \leq & 1 & j = 1, \ldots, s \\ & x^i_{jk} \text{ binary}, \ i = 1, \ldots, t, \ j = 1, \ldots, s-1, \ k = j+1, \ldots, s & & & \end{array} \qquad (IP)$$

We refer to the first set of constraints as **target cover constraints** and they ensure that a pair of sensors is assigned to each target. The second set of constraints are the **sensor matching constraints** and they ensure that no sensor is assigned to more

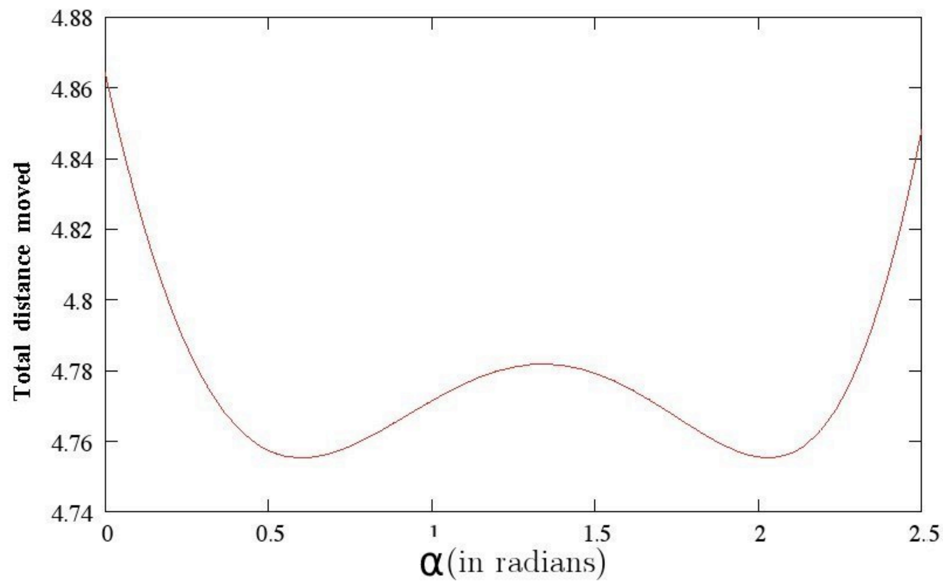

Figure 4: The objective function for a single target and two sensors, with required distances $d_1 = d_2 = 1$, initial radii $r_1 = 3$, $r_2 = 2$, required angle $\theta = 0$, and initial angle $\beta = 2.5$ radians.

than one target. The objective function coefficients c^i_{jk} are calculated as discussed in §2. The set of targets is denoted T and the set of sensors is denoted S.

Problem (IP) with general costs c^i_{jk} is an NP-hard problem, with reduction from the 3D-assignment problem (see Isler *et al.* [8]). Our problem is related to the Euclidean 3-matching problem. Johnsson *et al.* [9] showed that the general 3-matching problem is NP-complete.

The solution to the problem requires finding a matching of the sensors, with each matched pair assigned to a target. The polyhedral structure of the matching problem is well understood, with Edmonds [5] showing that only odd-set constraints are needed to define the convex hull of feasible integer solutions. These odd-set constraints can be generalized to our problem to give a valid inequality for each set W of r sensors where r is odd:

$$\sum_{i=1}^{t} \sum_{j,k \in W, j<k} x^i_{jk} \le \frac{r-1}{2} \tag{3}$$

This constraint is valid since r sensors can be used to cover at most $(r-1)/2$ targets. Summing the sensor matching constraints for the vertices in W, dividing by two, and rounding down gives (3). Hence, these cutting planes have Chvatal rank [4] equal to one. Further, when $|W| = 3$ the constraints are *clique inequalities*: a valid assignment can use at most one of the triples $\{(i, j, k) : i \in T, j \in W, k \in W\}$.

Each of our variables has three indices, with one index corresponding to a target and the other two indices corresponding to sensors. The problem is related to the three-index assignment problem, where all three indices are drawn from different sets. Interesting recent work on three-index and higher index assignment problems includes results on the optimal value by Grundel *et al.* [7] and a metaheuristic by Aiex *et al.* [1].

4 A GRASP routine

A greedy randomized adaptive search procedure (GRASP) was developed to find good feasible solutions. For larger problems, it was impracticable to solve the full integer programming formulation directly, so a subset of the triples was selected. One of the purposes of the GRASP routine was to aid in the selection of a very good selection of triples for the integer programming formulation with a restricted number of variables.

GRASP routines are metaheuristics designed to find very good solutions by sampling. They construct solutions in a similar manner to greedy heuristics, but rather than always making the greedy choice, one of the best few choices is made randomly. They are especially appropriate in our situation, where we use them as a method for selecting a good subset of variables to feed into an integer programming solver. This is because the GRASP will return many variables that are used in some solution that is close to optimal, and will not give variables that are unlikely to appear in an optimal solution. Another factor working in favor of GRASP is our assumption of a random deployment of sensors. This ensures that on average the sensors are spread out over the region as opposed to being clustered in a particular area. As a result when a sensor is chosen by GRASP, in all likelihood the pairing sensor is not too far apart, thereby helping to optimize the radial distance criterion. GRASP was introduced by Feo and Resende [6] and has been used for many combinatorial optimization problems. For example, Li *et al.* [10] have developed a GRASP for the quadratic assignment problem and Aiex *et al.* [1] presented a GRASP for the three-index assignment problem, problems which are somewhat related to our problem. For surveys see [12, 14].

The GRASP routine is described in Figure 5. The routine first finds a good perfect matching of the sensors and then assigns sensor pairs to targets. The constructed perfect matching is 2-optimal, in the sense that replacing any two edges in the matching with two unused edges will not result in a better perfect matching.

The final solution returned by the GRASP routine could be refined further with a

1. While there are unpaired sensors:
   (a) Randomly select an unpaired sensor.
   (b) Randomly select one of its three closest neighbors and pair off these sensors.
2. Use a 2-change routine to find a 2-optimal perfect matching of the sensors.
3. While there are uncovered targets:
   (a) Randomly select a target.
   (b) Randomly select one of the three closest sensors to the target.
   (c) Create a triple consisting of the target, the chosen sensor, and the pair of the sensor.

Figure 5: GRASP routine for generating good feasible solutions.

local optimization procedure. For our experiments, this was not necessary, since the set of triples determined by the GRASP routine enabled the exact solution of 90% of our randomly generated problems with 50 targets: see §5. Our approach could be regarded as a hybrid approach, combining a GRASP routine to select triples with an exact branch-and-cut approach to find the optimal solution drawn from those triples.

Isler *et al.* [8] looked at a greedy heuristic for this problem, as well as a 2-locally optimal heuristic with a $\frac{5}{3}$ approximation performance guarantee.

4.1 Path relinking

GRASP algorithms typically include a *path-relinking* component [15]. Path-relinking tries to find new solutions by combining known good solutions. The branch-and-cut approach could be regarded as a path-relinking strategy, since it looks at all possible combinations of the solutions to hand, as well as additional variables with small cost.

Given two feasible solutions, a path-relinking approach to our problem could move from one solution to the other by interchanging pairs of sensors, resulting in a path of feasible solutions. The value of these intermediate solutions along the path can be checked to see if they are better than the known solutions.

In the branch-and-cut setting, complete solutions of t triples are not necessary: the input to the branch-and-cut routine is a set of individual triples. The triples found along the path could be included in this set. Rather than limiting the algorithm to the triples found on paths, we constructed the sets S_i for $i = 1, \ldots, t$, consisting of all the sensors included in at least one pair assigned to target i in one of the five best solutions found by the GRASP routine. We then constructed all triples of the form $\{(i, j, k) : j, k \in S_i\}$. We discuss our results with this approach in the next section; the default setting was to *not* include these additional triples.

5 Computational results

Sensors and targets were randomly placed in the unit square using a uniform distribution. For a given number of targets, four parameters were varied: the number of sensors, the required angle θ, and the two radii d_1 and d_2. Five instances were generated for each set of parameters, and the tables below contain the means for the appropriate five instances. An example problem can be found in Figure 6.

Our algorithm consisted of two stages for problems with 50 targets. First, all the objective function coefficients c^i_{jk} were calculated by solving single variable optimization problems, as discussed in §2. The time for this calculation is listed as the

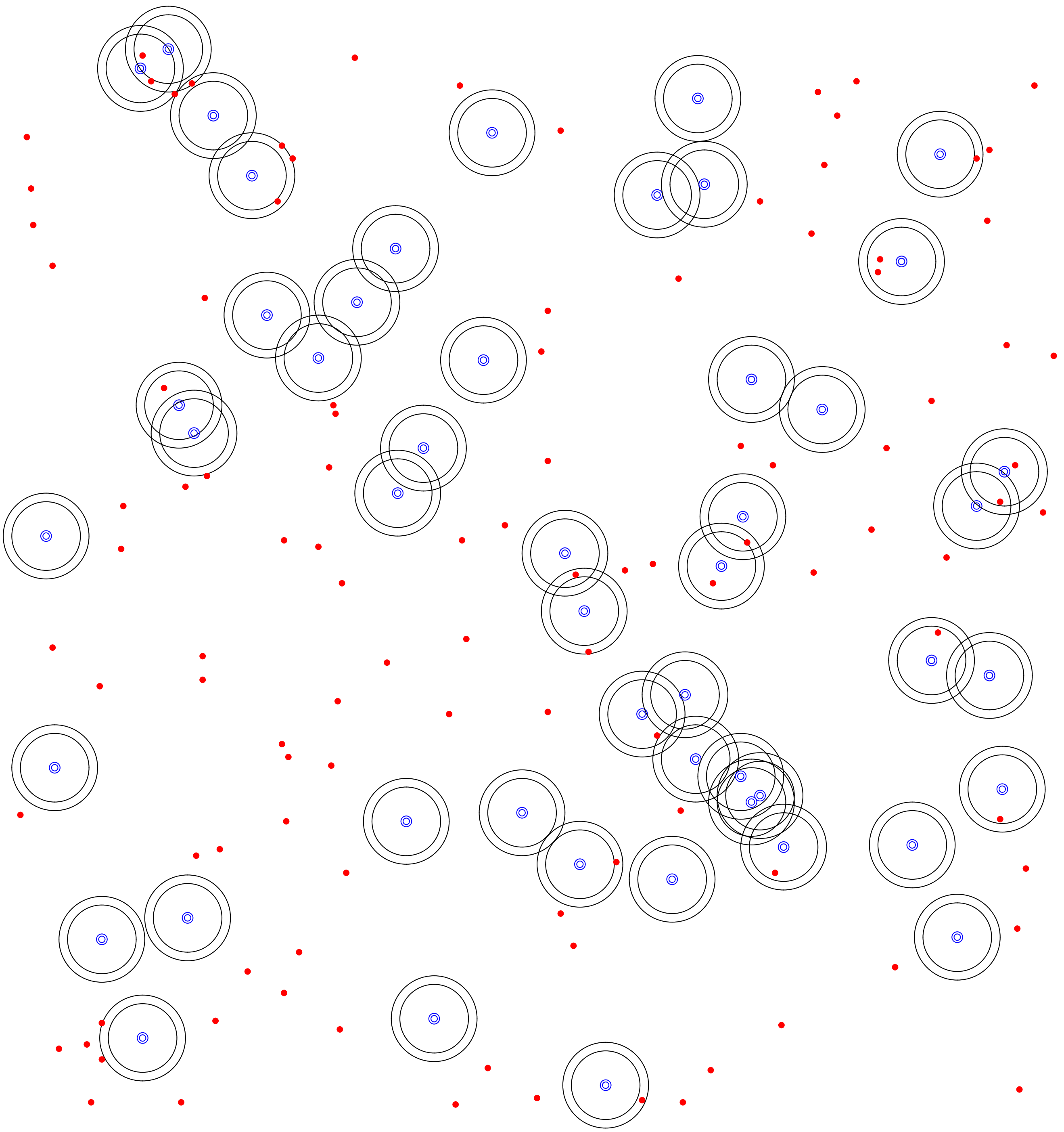

Figure 6: Illustration of a 50 target, 110 sensor problem. The solid circles denote the sensors while the smaller concentric circles denote the targets.

s	θ	d_1	d_2	Generation time	Solution time		Tree size	
				Mean	Mean	Max	Mean	Max
110	5.0	0.03	0.06	77.399	122.314	155.23	0.8	2
110	5.0	0.02	0.03	77.366	104.858	146.01	0.6	2
110	0.5	0.03	0.06	84.339	146.164	167.87	5.4	9
110	0.5	0.02	0.03	84.695	135.952	184.60	6.2	13
200	5.0	0.03	0.06	646.634	275.048	347.56	0.2	1
200	5.0	0.02	0.03	652.887	251.602	282.47	0.0	0
200	0.5	0.03	0.06	678.581	313.790	428.02	0.0	0
200	0.5	0.02	0.03	672.194	265.358	310.01	0.0	0

Table 1: Results for problems with 50 targets, working with all triples.

generation time in the tables below. Then the integer programming formulation (IP) was solved using CPLEX 9.1, with the time listed as *Solution time* in the tables. Note that CPLEX is able to exploit clique inequalities in its branch-and-cut approach. All results were obtained on a Sun Ultra 10 Workstation, with a 440MHz UltraSPARC-IIi processor. Computational results are given in Table 1. The final column gives the number of nodes in the branch-and-cut tree, as reported by CPLEX. The mean and the max of the solution time and tree size over the five instances are reported. The max generation time is not reported because the generation times for a given set of parameters varied by less than 5%.

With more targets, it is necessary to use a subset of the variables. To investigate the accuracy of the solution obtained with a reduced set of variables, the problems with 50 targets were attacked with a heuristically selected set of triples. We first ran the GRASP routine five times, and all triples found by the routine were included in the set of working variables. This ensures that the restricted integer program is feasible. Secondly, for each target we added the kt triples of smallest value that contain the given target, where $k = 10$ if $s < 3t$ and $k = 4$ if $s \geq 3t$. With this parameter choice, the number of variables grows quadratically in the number of targets for s=$O(t)$; the number of variables in the full formulation is $O(t^3)$. The results are contained in Table 2. The entries in the *Generation time* column now contain the time to calculate the objective function coefficients for the reduced set of variables; this includes the time for the GRASP routine. Note that the runtimes in Table 2 are far better than those in Table 1.

Recall that each line in Tables 1 and 2 corresponds to the average of five instances.

s	θ	d_1	d_2	Generation time	Solution time		Tree size	
				Mean	Mean	Max	Mean	Max
110	5.0	0.03	0.06	8.313	6.218	7.15	1.0	2
110	5.0	0.02	0.03	10.111	5.080	6.62	0.4	2
110	0.5	0.03	0.06	15.024	8.404	11.29	11.8	37
110	0.5	0.02	0.03	15.022	8.204	14.25	9.2	24
200	5.0	0.03	0.06	16.115	1.414	1.76	0.0	0
200	5.0	0.02	0.03	16.107	1.280	1.50	0.0	0
200	0.5	0.03	0.06	38.566	1.668	2.30	0.0	0
200	0.5	0.02	0.03	39.049	1.344	1.67	0.0	0

Table 2: Results for problems with 50 targets and a reduced set of triples.

For all 20 of the instances with 200 sensors, the optimal solution found using the reduced set of triples was identical to that found using the full set of triples. Four of the 20 instances with 110 sensors had inferior solutions with the reduced set of triples, namely two each of the two sets of instances with $\theta = 5$ (corresponding to no restriction on the angle). The relative errors are between 8×10^{-4} and 2.6×10^{-3}. Thus, the reduced set is sufficient in $36/40 = 90\%$ of the cases. When the number of sensors is close to twice the number of targets and where there is no restriction on the angle, it is occasionally useful to use two sensors from opposite sides to cover a particular target; the sensor matching heuristic is not good at detecting such pairs of sensors.

We experimented with the path-relinking strategy described at the end of §4.1 for the four problems where the reduced set of triples led to an inferior solution. The extra triples found using path-relinking did not improve the solution to any of these instances. Therefore, we did not use the path-relinking strategy for the experiments with 100 and 200 targets.

Because of the success of the algorithm with the reduced set of triples when there are 50 targets, we also attacked problems with 100 and 200 targets using a reduced set of triples. We used the same rules for determining an appropriate set of triples as for the 50 target case, and the results are contained in Tables 3 and 4. Comparing the *mean* and *max* values for the six sets of problems where the mean solution time was greater than 100 seconds shows that one problem in each of these sets was far harder than the others.

It is clear that the problems with $s \geq 3t$ are far easier than the problems where

s	θ	d_1	d_2	Generation time	Solution time		Tree size	
				Mean	Mean	Max	Mean	Max
220	5.0	0.03	0.06	89.279	44.462	59.99	9.4	24
220	5.0	0.02	0.03	89.100	35.262	46.54	4.8	13
220	0.5	0.03	0.06	144.430	326.062	1323.04	938.0	4573
220	0.5	0.02	0.03	144.799	273.464	1164.77	920.0	4524
400	5.0	0.03	0.06	208.770	6.516	9.67	0.4	2
400	5.0	0.02	0.03	208.753	6.218	9.22	0.6	3
400	0.5	0.03	0.06	390.253	7.320	13.16	2.4	12
400	0.5	0.02	0.03	390.949	6.974	8.52	0.2	1

Table 3: Results for problems with 100 targets and a reduced set of triples.

s	θ	d_1	d_2	Generation time	Solution time		Tree size	
				Mean	Mean	Max	Mean	Max
440	5.0	0.03	0.06	1164.06	705.026	1835.68	296.8	1102
440	5.0	0.02	0.03	1166.54	584.388	1712.05	186.6	789
440	0.5	0.03	0.06	1609.13	5733.162	15257.69	4845.2	15128
440	0.5	0.02	0.03	1598.26	730.480	1651.94	195.0	569
600	5.0	0.03	0.06	1753.79	48.952	65.33	1.4	5
600	5.0	0.02	0.03	1753.22	40.788	54.96	2.0	9
600	0.5	0.03	0.06	2641.71	90.778	199.44	39.6	176
600	0.5	0.02	0.03	2616.60	99.996	165.23	37.0	91

Table 4: Results for problems with 200 targets and a reduced set of triples.

$s \approx 2t$. The hardest problems are those with $\theta = 0.5$ and for some of these the solution time exceeds the generation time and the branch-and-cut tree is quite large. The solution time and tree size generally decrease as the radii decrease.

6 Conclusions and future research

The integer programming approach has enabled the solution of large instances. For comparison, Isler *et al.* [8] looked at similar problems with $t = 5$ or $t = 10$. When $s \geq 3t$, the integer programs appear to be relatively straightforward, and the set of triples found by the heuristics contains the optimal set for all of our test problems with $t = 50$. Hence, it appears that using a sufficiently large number of sensors will lead to good coverage, and such solutions can be found effectively with a hybrid GRASP/branch-and-cut approach.

For most problems, the generation time exceeds the solution time. The exceptions are the problems with $s \approx 2t$ and $\theta = 0.5$. For these problems, it may be worthwhile to develop more sophisticated cutting plane routines, allowing the exploitation of violated constraints of the form (3) for values of $|W| \geq 5$.

The generation time could be sped up using parallelization. For example, if the sensors have sufficient computing and battery power, they could be used to determine the objective function coefficients. For the problems where the generation time is the bottleneck, this parallelization would allow the solution of larger instances effectively, and would also increase the value of a more sophisticated cutting plane routine.

The algorithm has the structure of a branch-and-price-and-cut approach, except that we don't go back and do the pricing step in order to verify optimality. The integration of such a pricing step would probably allow the solution to optimality of the larger instances considered in this paper, especially those with $s \geq 3t$.

It would be desirable to extend this approach to give a robust solution. It is conceivable that some sensors may not work, and so alternative assignments may need to be made. Variants of the problem (IP) could be constructed to minimize the expected cost of covering all the targets, or to minimize the cost subject to the cost of meeting any failure falling within some tolerance.

Other problems can also be modeled using this framework. For example, Isler *et al.* [8] look at the "focus of attention" problem, where two sensors are assigned to a target with the aim of tracking the movement of the target. Values are determined for the assignment of a particular pair of sensors to a particular target, and then the aim is to choose an assignment to minimize the sum of these values. This is a problem in exactly the form of our problem (IP), but with a different method for calculating the objective function coefficients c^i_{jk}.

Acknowledgments

We are grateful to Volkan Isler for several useful discussions about this problem.

References

[1] R.M. Aiex, P.M. Pardalos, M.G.C. Resende, and G. Toraldo. GRASP with path-relinking for three-index assignment. *INFORMS Journal on Computing*, 17(2):224–247, 2005.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[3] J. L. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus. Deploying sensor networks with guaranteed capacity and fault tolerance. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing MobiHoc '05*, pages 309–319, 2005.

[4] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4:305–337, 1973.

[5] J. Edmonds. Maximum matching and a polyhedron with 0, 1 vertices. *Journal of Research National Bureau of Standards*, 69B:125–130, 1965.

[6] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[7] D. Grundel, C. A. S. Oliveira, and P. M. Pardalos. Asymptotic properties of random multidimensional assignment problems. *Journal of Optimization Theory and Applications*, 122(3):487–500, 2004.

[8] V. Isler, S. Khanna, J. Spletzer, and C. J. Taylor. Target tracking with distributed sensors: the focus of attention problem. *Computer Vision and Image Understanding Journal*, 100((1-2)):225–247, October-November 2005.

[9] M. Johnsson, G. Magyar, and O. Nevalainen. On the Euclidean 3-matching problem. *Nordic Journal of Computing*, 5(2):143–171, 1998.

[10] Y. Li, P. M. Pardalos, and M. G. C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In P. M. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, volume 16 of *DIMACS Series in*

Discrete Mathematics and Theoretical Computer Science, pages 237–261. American Mathematical Society, 1994.

[11] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing*, pages 123–134, 2002.

[12] L. S. Pitsoulis and M. G. C. Resende. Greedy randomized adaptive search procedures. In P. M. Pardalos and M. G. C. Resende, editors, *Handbook of Applied Optimization*, pages 168–183. Oxford University Press, 2002.

[13] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey of sensor networks. Technical Report UTDCS-10-03, Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, 2003.

[14] M. G. C. Resende and C. C. Ribeiro. Greedy radomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[15] M. G. C. Resende and C. C. Ribeiro. GRASP with path relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, pages 301–331. Springer, 2005.

[16] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless micro-sensor network models. *ACM Mobile Computing and Communications Review (MC2R)*, 6(2):28–36, 2002.

[17] M. Tubaishat and S. Madria. Sensor networks: An overview. *IEEE Potentials*, 22(2):20–23, 2003.