Sensor Deployment for Failure Diagnosis in Networked Aerial Robots: A Satisfiability-Based Approach

Fadi A. Aloul[1] and Nagaragan Kandasamy[2]

[1] Department of Computer Engineering, American University of Sharjah, UAE
faloul@aus.edu
[2] Department of Electrical and Computer Engineering, Drexel University, USA
kandasamy@ece.drexel.edu

Abstract. Unmanned aerial vehicles (UAVs) represent an important class of networked robotic applications that must be both highly dependable and autonomous. This paper addresses sensor deployment problems for distributed failure diagnosis in such networks where multiple vehicles must agree on the fault status of another UAV. Sensor placement is formulated using an integer linear programming (ILP) approach and solved using Boolean satisfiability (SAT)-based ILP solvers as well as generic ILP solvers. Our results indicate that the proposed models are tractable for medium-sized UAV networks.

Keywords: 0-1 ILP, SAT, UAV networks, fault diagnosis, distributed systems.

1 Introduction

Unmanned aerial vehicles (UAVs) represent an important class of robotic applications for distributed sensing and control. A collection of vehicles must perform a shared task while coordinating the required inter-vehicle actions using wireless communication. Examples include remote sensing, surveillance and patrol, and data collection over areas dangerous to human intervention. Such UAV networks have significant cost constraints. However, they must be both highly dependable and largely autonomous, requiring only high-level guidance from ground controllers.

Sensing and surveillance applications require that UAV node maintain a tight spatial formation or physical topology, including specified inter-node distances. In a typical decentralized formation-control scheme, each node receives information from neighboring nodes such as their position and velocity, and uses this data for local control aimed at maintaining its position within the topology [6]. Therefore, correct and timely information flow between nodes is critical to maintaining a stable topology.

To maintain the specified topology of a UAV network comprising nodes $N_1,...,N_q$, each N_j must communicate some critical information such as its position and velocity to neighboring nodes. Hardware (software) failures may, however, cause the node to transmit erroneous values. Though *physical redundancy* in the form of replicated sensors and processors can mask such node failures, it also adds to N_j 's

J. Marques-Silva and K.A. Sakallah (Eds.): SAT 2007, LNCS 4501, pp. 369 – 376, 2007.

cost, weight, and power consumption. A low-cost alternative is failure diagnosis using *analytical redundancy* [8] where other nodes in the topology use their local sensors and an appropriate mathematical model to estimate the values sent by N_j, and compare discrepancies between the actual and estimated values.

This paper addresses sensor deployment problems for distributed failure diagnosis in wireless UAV networks where multiple nodes must agree on the fault status of another node. We assume that a node N_i in this topology requires a testing configuration-a set of sensors-to monitor N_j for example, if N_i has a GPS sensor, and additionally, a 3D laser range finder, it can, using these sensors and an appropriate mathematical model, independently estimate N_j 's position. Several choices of testing configurations are typically available for N_i, differing from each other in their monitoring range, detection capabilities, and cost. (Another possible testing configuration on N_i may comprise a 2D laser range finder and an omni-directional camera.) Also, the sensors themselves may have varying operating distances. Clearly, long-range sensors can monitor multiple nodes, and at greater distances. However, the use of such expensive sensors may substantially increase the overall system cost. On the other hand, if only short-range sensors are used, effective diagnosis may only be achieved with a large number of such sensors. Therefore, efficient sensor selection and placement strategies are needed to minimize system cost while achieving the desired level of diagnosability.

Previous research has addressed distributed system diagnosis under the assumption that processing units test each other and exchange the test results to identify failures [2]. Failed units are then removed from future computations. Several variants of this problem have been studied in the literature, including diagnosing transient and intermittent faults [10], probabilistic diagnosis [4], and failure diagnosis in random, sparse, and highly regular topologies [7]. Since explicit tests are typically difficult to obtain in practice, various comparison-based approaches have also been proposed, where tasks are duplicated on multiple units and their results compared to identify faulty ones [3]. A good survey of prior diagnosis-related research is presented in [2]. The above papers, however, don't address the sensor selection and placement problems for failure diagnosis in wireless networks.

The authors of [5] present a method to identify faulty processors in ad hoc wireless networks via a comparison-based diagnosis model. They present algorithms for both fixed and time-varying network topologies, and show that diagnosis efficiency is significantly reduced when the topology changes with time. As before, sensor selection and placement problems are not addressed.

The sensor placement problem is related to both the alarm and guard placement problems [12, 13]. In [12], alarms are placed on the nodes of a failure propagation graph such that one failed node is uniquely and efficiently identified. A fault propagates along this graph activating one or more alarms and the diagnosis algorithm finds the node responsible for causing them. The guard placement problem can be informally stated as that of determining the minimum number of guards, each having a certain monitoring range, to cover the interior of an art gallery, represented as a polygon [13].

This paper uses an integer linear programming (ILP) approach to solve sensor deployment problems for distributed failure diagnosis in UAV networks. We specifically target popular UAV formations such as mesh, diamond, and circular topologies [15, 16] , and provide exact solutions for topologies up to 40 nodes, representative of topology sizes assumed by researchers while developing formation control algorithms [15, 17].

The proposed method aims to minimize both the testing and communication costs associated with identifying a bounded number of faulty UAV nodes. (In a typical wireless network, it is desirable to minimize the transmitting range of individual nodes to reduce power consumption and network interference.) Assuming an upper bound *f* on the number of node failures, we formulate and solve ILP models for the following optimization problem: Given a topology comprising *q* empty slots and an equal number of UAV nodes, each having a specific testing and communication configuration, allocate nodes to slots such that *system diagnosability*, in terms of the number of diagnosed nodes, is maximized. The above is termed the *MaxD* problem.

The model are solved using two different 0-1 ILP (SAT-based and generic-based) solvers [1, 9] and their performance is compared. Our experiments indicate that these models are tractable for topologies up to forty nodes.

The rest of this paper is organized as follows. Section 2 discusses some modeling assumptions and the distributed diagnosis approach. We develop ILP models for the *MaxD* problem in Section 3 and solve them in Section 4. We conclude this paper in Section 5.

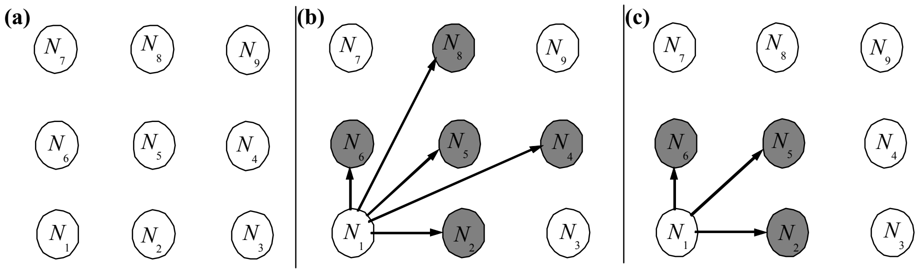

Fig. 1. (a) A grid topology of UAVs. The testing edges induced on the other nodes when N_1 chooses (b) a testing configuration T_1 and (b) a shorter-range testing configuration T_2 .

2 Preliminaries

This section describes the assumed system model and discusses the distributed diagnosis approach. The combinatorial nature of the sensor selection and placement problems of interest is briefly outlined.

2.1 System Model

We assume a distributed system where UAV nodes communicate with each other over a wireless network having limited bandwidth and must maintain the specified physical topology. Fig. 1(a) shows a grid topology for UAVs. High-level controllers coordinate with other nodes of interest to maintain the topology while feedback-control loops regulate local dynamics on each node.

A node N_i's position within a topology is given in the (x_i, y_i, z_i) dimensions and the distance between nodes N_i and N_j is

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (1)$$

When N_i has a choice of testing configurations, we let T_{ik} denote the k^{th} such configuration with testing range *range*(T_{ik}) and cost a_{ik} ; if $range(T_{ik}) \geq D_{ij}$, then N_i can test (or monitor) N_j using configuration T_{ik}. Similarly, if C_{il} denotes the l^{th} communication configuration on N_i having cost b_{il}, then node N_i can transmit messages to N_j if $range(C_{il}) \geq D_{ij}$. (Also, whenever the context is clear, we will refer to the k^{th} testing and l^{th} communication configuration on a node simply as T_k and C_l, respectively.)

As noted in Section 1, controllers on each N_j must communicate some critical information such as its position and velocity to neighboring nodes to maintain the desired topology. We assume that N_j may suffer operational failures including permanent and transient ones, thereby transmitting erroneous (sensor) information to its neighbors. Therefore, N_j must be diagnosed and removed from participating in future formation-control computations.

2.2 Distributed Diagnosis

Distributed diagnosis in a topology such as Fig. 1(a) requires that multiple testing nodes agree on the fault status of a testee node N_j. This is achieved using a 2-phase approach as follows. During phase 1, each testing node independently evaluates the information transmitted by N_j. These local decisions are then consolidated via a suitable agreement algorithm during phase 2 to obtain a global view of N_j's status. Similar 2-phase diagnosis schemes have been previously proposed to identify faulty processors [14].

We assume an analytical redundancy-based checking scheme that is executed locally on node N_i to evaluate the information sent by N_j. Node N_i uses its onboard testing configuration and an appropriate mathematical model to independently estimate N_j's sensor values. These estimates are compared to the actual values sent by N_j to generate a residue or error. During phase 2, N_i exchanges the locally generated residue with other testing nodes within communication range. Since multiple testers may employ both design and data diversity, i.e., use various testing configurations and/or models to estimate the same values, these residues may differ slightly from each other, and yet be correct. Therefore, each tester obtains a voted residue value using an approximate agreement algorithm, and evaluates it against an a *priori* defined threshold to diagnose N_j. If all testers perceive N_j's failure uniformly, then a suitable agreement algorithm is the median voter which selects the middle value from an odd number of residues by eliminating those residue pairs differing by the greatest amount [11]. At the end of phase 2, all fault-free nodes correctly identify N_j 's status.

Assuming an upper bound f on the number of node failures in the topology, we need at least $2f$+1 tester nodes to diagnose another node. The distributed approach described above also tolerates failures during the diagnosis process itself and increases confidence in the corresponding decisions. Finally, to reduce the cost of diagnosis, not all sensors on N_j are diagnosed. A few critical sensors are typically selected and checkers implemented to diagnose them.

Returning to Fig. 1, a testing configuration selected for N_i induces corresponding testing edges on neighboring nodes where $N_i \to N_j$ indicates that N_i can monitor N_j. Fig. 1(b) shows the edges generated when N_1 chooses a testing configuration T_1. Fig. 1(c), on the other hand, shows the case where a testing configuration T_2 with a shorter range is used. We also assume "line-of-sight" testing, i.e., there must be an uninterrupted path between the testee and tester nodes. Therefore, in Fig. 1, node N_1 cannot test N_3, N_7, and N_9, since they are not in the line-of-sight.

3 Problem Formulation

Given a topology with q empty slots and an equal number of nodes, each with a specific testing and communication configuration, allocate nodes to slots such that system diagnosability, in terms of the number of diagnosed nodes, is maximized. We assume an upper bound f on the number of node failures. We define the following decision variables.

$x_{ij} = 1$ if N_i occupies slot j; 0 otherwise

$m_{ij} = 1$ if node placed in slot i can moniter the node in j; 0 otherwise

$d_i = 1$ if the node placed in slot i is diagnosable; 0 otherwise

$p_{ij} = 1$ if a node N_i can communicate with N_j ; 0 otherwise

We maximize the cost function

$$\sum_{i=1}^{q} d_i \tag{2}$$

subject to the following constraints. A node N_i must be allocated to exactly one slot.

$$\sum_{i=j}^{q} x_{ij} = 1 \quad \forall i \tag{3}$$

Conversely, each slot j must have exactly one node allocated to it.

$$\sum_{j=1}^{q} x_{ij} = 1 \quad \forall j \tag{4}$$

Let s_{ij} denote the set of nodes, when placed in slot *i*, can monitor slot *j*; node $N_k \in s_{ij}$ if it has a testing configuration T_{lk} such that $range(T_{lk}) \geq D_{ij}$. Constraint (5) sets the decision variable m_{ij} to indicate if a chosen node-to-slot allocation enables slot *i* to test slot *j*, and constraint (6) ensures that a node allocated to slot *j* is monitored by at least $2f + 1$ other nodes.

$$\sum_{N_k \in s_{ij}} x_{ki} - m_{ij} \geq 0 \quad \forall i, \forall j, i \neq j \tag{5}$$

$$\sum_{i=1}^{q} m_{ij} \geq 2f + 1 \quad \forall j, i \neq j \tag{6}$$

Constraint (7) sets the decision variable p_{ij} indicating if a chosen node-to-slot allocation enables slot *i* to transmit to slot *j*. Let s_{ij} now denote the set of nodes, when placed in slot *i*, have the transmission range to reach slot *j*; node $N_k \in s_{ij}$ if its communication configuration C_{lk} is such that $range(C_{lk}) \geq D_{ij}$.

$$\sum_{N_k \in s_{ij}} x_{ki} - p_{ij} \geq 0 \quad \forall i, \forall j, i \neq j \tag{7}$$

Constraints (8), (9), and (10) select exactly $2f + 1$ slots to diagnose the node placed in slot *j*. Note that the node placed in slot *j* must transmit its sensor values to every member of the selected subset; otherwise it is not diagnosable. For example, if under some node-to-slot allocation, slot *j* cannot transmit its sensor values to a slot *i* chosen to monitor it, i.e., $z_{ij} = 1$ and $p_{ij} = 0$, then clearly d_j must be 0 to satisfy constraint (10).

$$m_{ij} - z_{ij} \geq 0 \quad \forall j, \forall i, i \neq j \tag{8}$$

$$\sum_{i=1}^{q} z_{ij} = 2f + 1 \quad \forall j, i \neq j \tag{9}$$

$$d_i + z_{ij} - p_{ij} \leq 1 \tag{10}$$

Finally, for each slot *j*, the $2f + 1$ slots chosen to diagnose it must be able to exchange the test results amongst themselves and reach an agreement during phase 2 of the diagnosis process. These slots must be fully connected or else the node in slot *j* cannot be diagnosed. For example, consider a pair of slots *i* and *k* chosen to diagnose slot *j*, i.e., $z_{ij} = z_{kj} = 1$. However, if slots *i* and *k* cannot exchange their test results, i.e., if $p_{ik} = 0$ or $p_{ki} = 0$, then clearly d_j must be zero to satisfy both constraints (11) and (12).

$$d_j + z_{ij} + z_{kj} - p_{ik} \leq 2 \tag{11}$$

$$d_j + z_{ij} + z_{kj} - p_{ki} \leq 2 \quad \forall i, j, i \neq j, k \neq (i \vee j) \tag{12}$$

4 Performance Evaluation

We solve the ILP model developed in Section using the generic-based ILP solver CPLEX [9] and the SAT-based 0-1 ILP solver PBS ver. 4 [1]. The CPLEX and PBS solvers were executed on an Intel Xeon 3 GHz machine with 4 GB RAM. The results presented in this section assume *grid* topologies, though the models are directly applicable to other important formations such as circles and diamonds.

Both CPLEX and PBS were used to solve the *MaxD* model for different topology sizes. For each experiment, we generated a grid topology comprising *q* empty slots. Assuming an equal number of nodes, a specific testing and communication configuration was pre-selected for each node such that the distribution of configurations to nodes was uniform. The time-out periods for the CPLEX and PBS solvers were set to 10,000 seconds.

Table 1 summarizes the results obtained by CPLEX and PBS, in terms of the number of diagnosable nodes, for $f = 1, 2$. Optimal results are shown in boldface in the figures. We assume five testing (communication) configurations corresponding to α values of 0, 0.25, 0.5, 0.75, and 1. The results show that PBS outperforms CPLEX in the $f = 1$ case. Both solvers time-out trying to prove the solution optimality in the $f = 2$ case. To summarize, the *MaxD* model appears tractable for medium-size topologies up to 40 nodes.

Table 1. Number of nodes diagnosed under the difference fault models; five testing (communication) configurations corresponding to α = 0, 0.25, 0.5, 0.75, 1 are assumed

Nodes(q)	***f* = 1**				***f* = 2**			
	PBS		**CPLEX**		**PBS**		**CPLEX**	
	Cost	**Time**	**Cost**	**Time**	**Cost**	**Time**	**Cost**	**Time**
4x5	**20**	0.26	**20**	164	16	t/o	15	t/o
5x5	**25**	1.7	**25**	237	20	t/o	19	t/o
5x6	**30**	1.59	**30**	2163	24	t/o	24	t/o
6x6	**36**	10.44	33	t/o	28	t/o	0	t/o

5 Conclusions

This paper has addressed the problem of sensor deployment for distributed failure diagnosis in UAV networks. The *MaxD* model allows designers to specify the placement of nodes within a given topology to maximize system diagnosability while incurring no additional testing costs. The ILP model was solved using the generic-based ILP solver CPLEX and SAT-based 0-1 ILP solver PBS, and experimental results indicate that they are tractable for medium-size topologies. For larger topologies, a straightforward (and sub-optimal) solution is to partition the given topology into portions tractable for the ILP models, and solve the resulting sub-problems in parallel. We will investigate this and other approximation methods in future work.

References

[1] F. Aloul, A. Ramani, I. Markov, and K. Sakallah, "Generic ILP Versus Specialized 0-1 ILP: An Update," *Proc. IEEE/ACM Conf. Computer Aided Design* (ICCAD), 450-457, November 2002.

[2] M. Barborak, M. Malek, and A. Dahbura, "The Consensus Problem in Fault-Tolerant Computing," *ACM Computing Surveys*, 25(2), 171-219, June 1993.

[3] D. Blough and H. Brown, "The Broadcast Comparison Model for On-line Fault Diagnosis in Multicomputer Systems: Theory and Implementation," *IEEE Trans. Comp.*, 48(5), 470-493, May 1999.

[4] D. Blough, G. Sullivan, and G. Masson, "Efficient Diagnosis of Multiprocessor Systems under Probabilistic Models," *IEEE Trans. Computers*, 41(9), 1126-1136, September 1992.

[5] S. Chessa and P. Santi, "Comparison-Based System-Level Fault Diagnosis in Ad-hoc Networks," *Proc. IEEE Symposium Reliable Distributed Systems*, 257-266, 2001.

[6] J. Fax and R. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Transactions on Automatic Control*, 49(9), 1465-1476, September 2004.

[7] D. Fussel and S. Rangarajan, "Probabilistic Diagnosis of Multiprocessor Systems with Arbitrary Connectivity," *Proc. IEEE Symposium on Fault-Tolerant Computing*, 560-565, 1989.

[8] J. Gertler, "Fault Detection and Diagnosis in Engineering Systems," *Marcel Dekker*, NewYork, 1998.

[9] ILOG CPLEX, *http://www.ilog.com/products/cplex*

[10] W. Kozlowski and H. Krawczyk, "A Comparison-Based Approach to Multi-Computer System Diagnosis in Hybrid Fault Situations," *IEEE Trans. Computers*, 40(11), 1283-1287, November 1991.

[11] P. Lorczak, A. Caglayan, and D. Eckhardt, "A Theoretical Investigation of Generalized Voters for Redundant Systems," *Proc. IEEE Symposium on Fault-Tolerant Computing*, 444-451, 1989.

[12] N. Rao, "Computational Complexity Issues in Operative Diagnosis of Graph-Based Systems," *IEEE Trans. Computers*, 42(4), 447-457, April 1993.

[13] J. O'Rourke, "Art Gallery Theorems and Algorithms," *Oxford University Press*, Oxford, 1987.

[14] C. J. Walter, P. Lincoln, and N. Suri, "Formally Verified On-Line Diagnosis," *IEEE Trans. Software Engineering*, 23(11), 684-721, November 1997.

[15] J. Fax and R. Murray. "Information Flow and Cooperative Control of Vehicle Formations," *Proc. IFAC World Congress*, July 2002.

[16] A. Pant et al., "Mesh Stability of Unmanned Aerial Vehicle Clusters," *Proc. American Control Conf.*, 2001.

[17] J. Desai, J. Ostrowski, V. Kumar. "Control of Changes in Formation for a Team of Mobile Robots," *Proc. IEEE Conf. Robotics & Automation*, 1556-1561, May 1999.