Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Sensor Selection and Placement for Failure Diagnosis in Networked Aerial Robots

Nagarajan Kandasamy
ECE Department
Drexel University
kandasamy@ece.drexel.edu

Fadi A. Aloul
Computer Engineering Department
American University of Sharjah
faloul@aus.edu

T. John Koo
EECS Department
Vanderbilt University
john.koo@vanderbilt.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 selection and placement problems for distributed failure diagnosis in such networks where multiple vehicles must agree on the fault status of another UAV. An integer linear programming (ILP) approach is proposed to solve these problems. The ILP models of interest are developed and solved using two different solvers. Experimental results indicate that the proposed models are tractable for medium-sized topologies.

Index Terms—UAV networks, fault diagnosis, ad hoc sensor networks, distributed systems.

I. 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 nodes in the UAV network 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 [7, 20]. 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 $N_1, \ldots, N_q$, 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 cost, weight, and power consumption. A low-cost alternative is failure diagnosis using *analytical redundancy* [9] 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 selection and placement 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, 15]. 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 [11, 13], probabilistic diagnosis [4], and failure diagnosis in random, sparse, and highly regular topologies [5] [8]. 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, 14, 18]. 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 [6] 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 selection problem is related to both the alarm and guard placement problems [16, 17]. In [16], alarms are placed on the nodes of a failure propagation graph such that a single system fault (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 [17].

This paper uses an integer linear programming (ILP) approach to solve sensor selection and placement problems for distributed failure diagnosis in UAV networks. We specifically target popular UAV formations such as mesh, diamond, and circular topologies [20] [22], and provide exact solutions for topologies up to 40 nodes, representative of topology sizes assumed by researchers while developing formation control algorithms [21] [23].

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 problems.

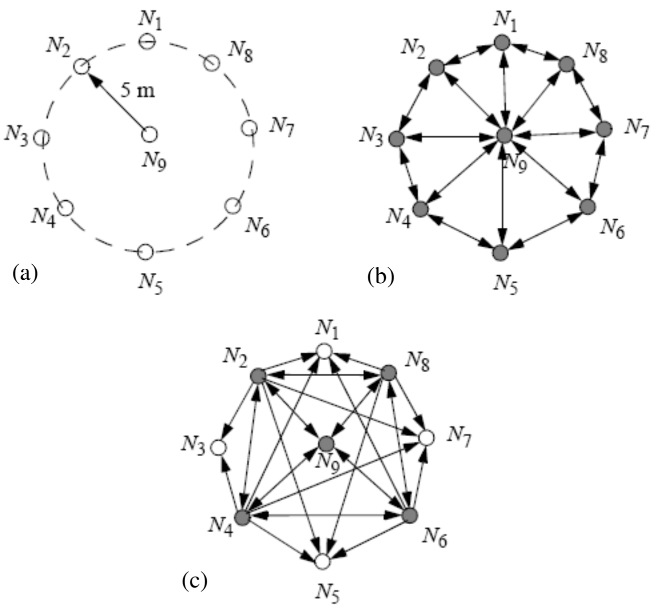


Fig. 1. (a) A circular UAV topology, and testing edges generated when (b) all nodes are equipped with configuration T_1 and (c) nodes N_2, N_4, N_6, and N_8 are equipped with T_2, and N_9 with T_1.

- ***Sensor selection.*** Given a topology comprising q UAV nodes, where each node can choose between multiple testing and communication configurations, select a configuration for each N_i to guarantee the diagnosis of any faulty node in the topology while minimizing overall testing cost. This is termed the *MinC* problem.
- ***Sensor placement.*** 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 proposed models are solved using two different ILP solvers [1, 10] and their performance compared. Our experiments indicate that these models are tractable for topologies up to forty nodes. We also briefly discuss how to extend our approach to obtain approximate solutions for larger topologies.

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 *MinC* and *MaxD* problems in Section 3 and solve them in Section 4. We conclude this paper in Section 5.

II. 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.

A. 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 circular topology of radius 5m. 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

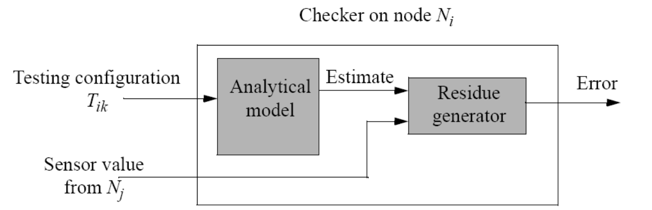


Fig. 2. An analytical redundancy-based checking scheme executed locally node N_i to evaluate N_j.

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

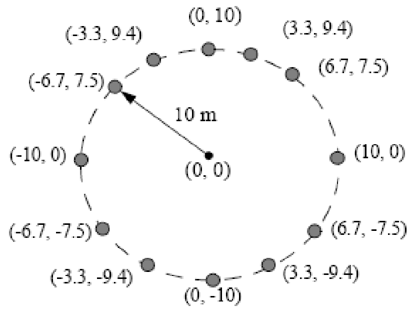
When N_i has a choice of testing configurations, we let T_{ik} denote the k^{th} such configuration with testing range $range(T_{ik})$ and cost a_{ik}; if $range(T_{ik}) \geq D_{ij}$, then N_i can test (or monitor) N_j using configuration T_{ik}. Similarly, if C_{il} denotes the l^{th} communication 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 I, 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.

B. 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 two-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 two-phase diagnosis schemes have been previously proposed to identify faulty processors [19].

Fig. 2 shows an analytical redundancy-based checking scheme executed locally on node N_i to evaluate the information sent by N_j. In the figure, 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 [12]. 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



(a)

Available testing (communication) configurations

(b)

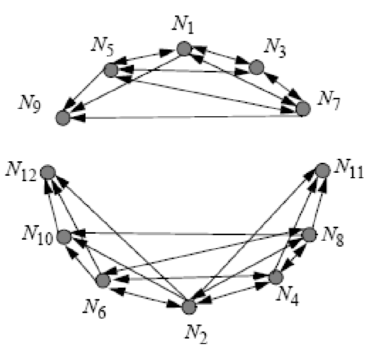
	T_1 (C_1)	T_2 (C_2)	T_3 (C_3)	T_4 (C_4)
Range (m)	5.0	10.0	15.0	20.0
Cost	1	2	3	4

The cost-optimal selection of configurations on nodes

(c)

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
T_i	T_2	T_3	T_2	T_3	T_3	T_3	T_3	T_3	-	T_2	-	-
C_i	C_2	C_2	C_2	C_2	C_3	C_3	C_3	C_3	C_3	C_3	C_3	C_3
Cost	4	5	4	5	6	6	6	6	3	5	3	3

Optimal diagnosis cost: 56



(d)

Fig. 3. (a) A circular topology of 10m radius with twelve nodes, (b) the available testing (communication) configuration choices, (c) cost-optimal selection of configurations for nodes, and (d) testing edges corresponding to the selected configurations guaranteeing the diagnosis of a node in the topology for $f = 1$.

are diagnosed. A few critical sensors are typically selected and checkers implemented to diagnose them.

Returning to Fig. 1, we assume two testing configuration choices T_1 and T_2 with monitoring ranges of 5m and 9.5m, respectively, for a node N_i. A configuration selected for N_i induces corresponding testing edges on neighboring nodes where $N_i \rightarrow N_j$ indicates that N_i can monitor N_j. Fig. 1(b) shows the edges generated when each N_i chooses T_1. Assuming full communication connectivity between nodes and $f = 1$, nine such testing configurations are needed to diagnose a node in the topology. Fig. 1(c), on the other hand, shows the case where T_2, placed on nodes N_2, N_4, N_6, and N_8, and T_1, placed on N_9, achieve the same level of diagnosability as Fig. 1(b). The overall diagnosis cost corresponding to Fig. 1(c) is less than that of Fig. 1(b) if $a_2 < 2 \cdot a_1$, where a_1 (a_2) denotes the dollar cost of T_1 (T_2).

The above example illustrates the combinatorial nature of the sensor selection and placement problems of interest, and ILP formulations provide a systematic and rigorous approach to exploring the search space for optimal solutions. Also, increased computing power and efficient implementations allow modern ILP solvers [1, 10] to tackle large optimization problems.

III. Problem Formulation

This section develops the ILP models for the sensor selection (*MinC*) and placement (*MaxD*) problems, and illustrates their applicability using small examples.

A. The *MinC* Problem

Consider a topology comprising q nodes, where each N_i may choose between n testing and m communication configurations $\{T_{i1}, T_{i2}, \ldots, T_{in}\}$ and $\{C_{i1}, C_{i2}, \ldots, C_{im}\}$, respectively. Assuming an upper bound f on node failures, the *MinC* optimization problem is to select the minimum-cost testing and communication configuration for the overall system topology while guaranteeing the diagnosis of any faulty node in the topology.

First, we introduce the following decision variables.

$$t_{ik} = \begin{cases} 1 : \text{if testing configuration } T_{ik} \text{ is chosen for node } N_i \\ 0 : \text{otherwise} \end{cases}$$

$$c_{il} = \begin{cases} 1 : \text{if comm. configuration } C_{il} \text{ is chosen for node } N_i \\ 0 : \text{otherwise} \end{cases}$$

$$m_{ij} = \begin{cases} 1 : \text{if a Node } N_i \text{ can moniter (test) node } N_j \\ 0 : \text{otherwise} \end{cases}$$

$$p_{ij} = \begin{cases} 1 : \text{if a node } N_i \text{ can communicate with node } N_j \\ 0 : \text{otherwise} \end{cases}$$

The following linear cost function must be minimized.

$$\sum_{i=1}^{q} \sum_{k=1}^{n} a_{ik} \cdot t_{ik} + \sum_{i=1}^{q} \sum_{l=1}^{m} b_{il} \cdot c_{il}$$

where a_{ik} and b_{il} denote the dollar cost corresponding to testing configuration T_{ik} and communication configuration C_{il}, respectively, on node N_i. The optimization is subject to the following constraints guaranteeing the distributed diagnosis of faulty nodes in the topology.

A node N_j must choose at most one of n available testing configurations.

$$\sum_{k=1}^{n} t_{ik} \le 1 \quad \forall j \tag{1}$$

A node N_j must also choose between one of m possible communication configurations.

$$\sum_{l=1}^{m} c_{il} = 1 \quad \forall j \tag{2}$$

Each node N_j must be monitored by at least $2f + 1$ other nodes. Constraint (3) sets the decision variable m_{ij} based on N_i's ability to monitor N_j. If s_{ij} denotes the set of all possible testing configurations on N_i capable of monitoring N_j, where a test configuration $T_{ik} \in s_{ij}$ if $range(T_{ik}) \ge D_{ij}$, then

$$\sum_{T_{ik} \in s_{ij}} t_{ik} - m_{ij} \geq 0 \quad \forall i, \forall j, i \neq j \tag{3}$$

Clearly, if $s_{ij} = 0$, then no test configuration available to N_i has the range to monitor N_j and m_{ij} must be 0 to satisfy the above constraint. If, however, $m_{ij} = 1$, i.e., N_i monitors N_j, then N_i must have selected a suitable testing configuration from s_{ij} (since $\sum_{T_{ik} \in s_{ij}} t_{ik} = 1$) to satisfy constraint (3).

Constraint (4) ensures that at least $2f+1$ nodes are chosen to monitor N_j.

$$\sum_{i=1}^{q} m_{ij} \geq 2f+1 \quad \forall j, i \neq j \tag{4}$$

Constraint (5) sets p_{ij} to indicate if node N_i can transmit messages to N_j using the chosen communication configuration. If s_{ij} now denotes the set of all possible communication configurations on N_i capable of transmitting to N_j, where configuration $C_{il} \in s_{ij}$ if $range(C_{il}) \geq D_{ij}$, then

$$\sum_{C_{il} \in s_{ij}} c_{il} - p_{ij} \geq 0 \quad \forall i, \forall j, i \neq j \tag{5}$$

Constraints (6) and (7) use a decision variable $z_{ij} \in (0, 1)$ to select a subset of the monitoring nodes−$2f+1$ to be exact−to participate in the distributed failure diagnosis scheme. Constraint (8) ensures that the node N_j being tested can transmit its actual sensor values to the monitoring nodes. (Recall from Fig. 2 that the checking scheme on a tester node compares the actual and estimated sensor values to generate a residue during phase 1 of the diagnosis process.)

$$m_{ij} - z_{ij} \geq 0 \quad \forall j, \forall i, i \neq j \tag{6}$$

$$\sum_{i=1}^{q} z_{ij} = 2f+1 \quad \forall j, i \neq j \tag{7}$$

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

Finally, the chosen subset of $2f+1$ nodes must have the communication means to exchange the residues amongst themselves and reach an agreement during phase 2 of the diagnosis process, i.e., these nodes must be fully connected. The following constraints enforce this requirement.

$$z_{ij} + z_{lj} - p_{il} \leq 1 \tag{9}$$

$$z_{ij} + z_{lj} - p_{li} \leq 1 \quad \forall i, j, l, i \neq j, l \neq (i \vee j) \tag{10}$$

For any pair of nodes, N_i and N_l, selected to participate in N_j's diagnosis, i.e., $z_{ij} = z_{lj} = 1$, N_i (N_l) must be able to transmit its locally computed residue to N_l (N_i). So, p_{il} and p_{li} must both be one to satisfy constraints (9) and (10), respectively.

B. The *MaxD* Problem

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. Again, we assume an upper bound *f* on the number of node failures. We define the following decision variables.

$$x_{ij} = \begin{cases} 1 : \text{if node } N_i \text{ occupies slot } j \\ 0 : \text{otherwise} \end{cases}$$

$$m_{ij} = \begin{cases} 1 : \text{if node placed in slot } i \text{ can moniter the node in } j \\ 0 : \text{otherwise} \end{cases}$$

Available testing (communication) configurations

	$T_1(C_1)$	$T_2(C_2)$	$T_3(C_3)$	$T_4(C_4)$	$T_5(C_5)$
Range (m)	3.4	7.5	11.7	15.9	20.0
Cost	1	2	3	4	5

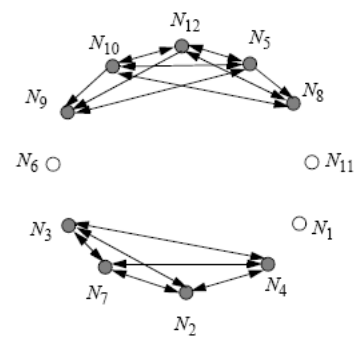
(a)

An *a priori* distribution of configurations to nodes

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}
T_i	T_1	T_2	T_3	T_4	T_5	T_1	T_2	T_3	T_4	T_5	T_1	T_2
C_i	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5	C_1	C_2

(b)

Node-to-slot allocation resulting in optimal diagnosability



(c)

Fig. 4. (a) The assumed testing and communication configurations, (b) the a priori selected configuration on each node, and (c) the node-to-slot allocation resulting in maximum system diagnosability for $f = 1$ and the corresponding testing edges.

$$d_i = \begin{cases} 1 : \text{if the node placed in slot } i \text{ is diagnosable} \\ 0 : \text{otherwise} \end{cases}$$

Note that m_{ij}, previously introduced in Section A, has been redefined for the *MaxD* problem.

We maximize the cost function

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

subject to the following constraints.

A node N_i must be allocated to exactly one slot.

$$\sum_{i=j}^{q} x_{ij} = 1 \quad \forall i \tag{11}$$

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

$$\sum_{j=1}^{q} x_{ij} = 1 \quad \forall j \tag{12}$$

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 $range(T_{lk}) > D_{ij}$. Constraint (13) sets the decision variable m_{ij} to indicate if a chosen node-to-slot allocation enables slot *i* to test slot *j*, and constraint (14) ensures that a node allocated to slot *j* is monitored by at least $2f+1$ other nodes.

$$\sum_{N_k \in s_{ij}} x_{ki} - m_{ij} \geq 0 \quad \forall i, \forall j, i \neq j \tag{13}$$

$$\sum_{i=1}^{q} m_{ij} \ge 2f+1 \quad \forall j, i \ne j \tag{14}$$

Constraint (15) 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 $range(C_{lk}) > D_{ij}$.

$$\sum_{N_k \in s_{ij}} x_{ki} - p_{ij} \ge 0 \quad \forall i, \forall j, i \ne j \tag{15}$$

Constraints (16), (17), and (18) 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 $p_{ji} = 0$, then clearly d_j must be 0 to satisfy constraint (18).

$$m_{ij} - z_{ij} \ge 0 \quad \forall j, \forall i, i \ne j \tag{16}$$

$$\sum_{i=1}^{q} z_{ij} = 2f+1 \quad \forall j, i \ne j \tag{17}$$

$$d_j + z_{ij} - p_{ji} \le 1 \tag{18}$$

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 (19) and (20).

$$d_j + z_{ij} + z_{kj} - p_{ik} \le 2 \tag{19}$$

$$d_j + z_{ij} + z_{kj} - p_{ki} \le 2 \quad \forall i, j, k, i \ne j, k \ne (i \vee j) \tag{20}$$

C. Examples

We now apply the previously developed ILP models to a small example. First, consider the *MinC* model. Fig. 3(a) shows a circular topology of 10m radius with twelve nodes (or slots), and Fig. 3(b) lists the four configuration choices available to each node, their testing and communication ranges, and corresponding costs. Fig. 3(c) shows the configurations selected by the ILP model for each node to obtain the cost-optimal solution and Fig. 3(d) shows the testing graph corresponding to these selections, guaranteeing the diagnosis of any node in the topology for $f = 1$. Note that a testing configuration is not selected for nodes N_9, N_{11}, and N_{12}. Also, the optimal cost achieved is 56 as compared to the worst-case cost of 120 (obtained when each node chooses the testing and communication configuration with the maximum range and cost), resulting in a cost savings of 53.3%.

The *MaxD* model uses the same topology in Fig. 3(a) with the five configurations listed in Fig. 4(a). Each node now has onboard, an *a priori* selected configuration, shown in Fig. 4(b). The node-to-slot allocation resulting in maximum diagnosability under the single fault model in shown in Fig. 4(c). Note that failures affecting nodes N_{11}, N_6, and N_1 cannot be diagnosed under this allocation.

IV. Performance Evaluation

We solve the ILP models developed in Section III using the CPLEX [10] and PBS [1] solvers. The CPLEX and PBS solvers were executed on Sunblade 1000 and 2.8 GHz Pentium 4 machines, respectively, each with 500 MB main memory. The results presented in this section assume circular topologies, though the models are directly applicable to other important formations such as meshes and diamonds.

Nodes(q)	C_{max}	*f = 1*		*f = 2*		*f = 3*	
		Cost	**Time**	**Cost**	**Time**	**Cost**	**Time**
4	40	**40**	0	uns	0	uns	0
8	80	**46**	0.29	**64**	0.19	**80**	0
12	120	**56**	167.18	**69**	10.28	**88**	9.59
16	160	**58**	69.81	**79**	67.82	**93**	67.02
20	200	**58**	424.75	89	t/o	**101**	251.17
22	220	**54**	2099.09	**74**	34.37	**102**	1311
26	260	**58**	4547.47	**78**	2086.2	108	t/o
32	320	78	t/o	109	t/o	123	t/o
34	340	64	t/o	98	t/o	115	t/o

Table 1. The *MinC* results obtained by CPLEX for different topology sizes and number of node failures.

For each experiment, we distribute *q* nodes in a circular topology of radius 100 units. The various testing and communication configurations are generated as follows. We first compute D_{max} and D_{min}, the maximum and minimum inter-node distances, respectively, within the topology. Each testing configuration now has a range given by $\max(D_{min}, (1-\alpha) \cdot D_{max})$ where $0 \le \alpha \le 1$ is a user-defined variable. For our experiments, five different testing configurations having costs 5, 4, 3, 2, and 1 were obtained with α assuming values of 0, 0.5, 0.25, and 1, respectively. (The configuration with the greatest testing range has the maximum cost of 5, and so on.) The above process is repeated to obtain five different communication configurations.

A. Analysis of *MinC* Results

Table 1 summarizes the results obtained by CPLEX for different topology sizes under one, two, and three node failures. The results show the achieved costs as well as the corresponding processing times. Column two shows the worst-case diagnosis cost c_{max} incurred by the overall system if both the testing and communication configurations selected for each node cost 5 apiece. The time-out period for the solver was set to 10,000 seconds and the optimal solutions are shown in boldface. In case of a timeout, indicated by *t/o* in the table, the best available solution is shown. The results indicate that CPLEX succeeds in finding optimal solutions in many cases, and even those solutions obtained after a solver time-out incur substantially lower cost than their corresponding c_{max} values; for example, the cost incurred for diagnosing a node when $f = 1, 2, 3$ is, on average, 68%, 55%, and 45% less than the corresponding c_{max}. For the smallest topology with $q = 4$, the optimization problem is unsatisfiable and denoted by *uns* in the table, for $f = 2, 3$. (Note that to diagnose a node under a two (three) fault model, a minimum of five (seven) tester nodes are required.)

Finally, the results obtained by PBS for *MinC* are not shown here since CPLEX consistently provided better solutions. However, it must be noted that though PBS timed-out in most cases (the time-out was set to 3,000 seconds.), the sub-optimal solutions obtained were also substantially better than c_{max}.

B. Analysis of *MaxD* Results

Both CPLEX and PBS were used to solve the *MaxD* model for different topology sizes. For each experiment, we generated a circular topology of radius 100 units 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 and 3,000 seconds, respectively.

Nodes(q)	*f = 1*		*f = 2*	
	PBS	CPLEX	PBS	CPLEX
4	uns	uns	uns	uns
8	uns	uns	uns	uns
12	**9**	**9**	uns	uns
16	12	12	8	8
20	16	16	16	15
22	17	17	17	13
26	20	20	20	16
32	25	32	14	32
34	27	34	15	24

Table 2. Number of nodes diagnosed under the different fault models; five testing (communication) configurations corresponding to $\alpha = 0$, .25, .5, .75, 1 are assumed

Nodes(q)	*f = 1*		*f = 2*	
	PBS	CPLEX	PBS	CPLEX
4	uns	uns	uns	uns
8	uns	uns	uns	uns
12	**9**	**9**	uns	uns
16	12	12	uns	uns
20	15	15	14	**14**
22	16	16	14	13
26	19	19	19	19
32	23	23	7	14
34	25	25	6	23

Table 3. Number of nodes diagnosed when only four configurations corresponding to $\alpha = .25, .5, .75, 1$ are assumed

Table 2 summarizes the results obtained by CPLEX and PBS, in terms of the number of diagnosable nodes, for $f = 1, 2$. Again, optimal results are shown in boldface in the figures. We assume five testing (communication) configurations corresponding to α values of 0, .25, .5, .75, and 1. The results indicate that *MaxD* is a substantially harder problem to solve than *MinC*. Though the solutions are obtained quickly, both solvers time-out trying to prove their optimality, and so Table 2 does not show the corresponding processing times. To obtain the results in Table 3, only four configurations were assumed, corresponding to α values of .25, .5, .75, and 1. Since the testing and communication ranges on nodes are now somewhat limited, the results show a slight decrease in system diagnosability.

To summarize, the ILP models appear tractable for medium-size topologies up to 40 nodes and *MinC* was easier to solve than the *MaxD* problem. Also, in the case of *MinC*, even those solutions not proved as optimal still result in substantial cost savings when compared to the worst-case bounds. The next section briefly discusses how to apply the proposed models to larger network topologies.

V. Discussion

This paper has addressed the problems of sensor selection and placement for distributed failure diagnosis in UAV networks. We developed two ILP models, namely *MinC* and *MaxD* to solve the problems of interest. The *MinC* model allows designers to systematically explore available low-cost system diagnosis alternatives without having to duplicate sensors and processors on individual nodes. The *MaxD* model, on the other hand, allows designers to specify the placement of nodes within a given topology to maximize system diagnosability while incurring no additional testing costs. The ILP models were solved using the CPLEX and PBS solvers, 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. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah, "Generic ILP Versus Specialized 0-1 ILP: An Update," *Proc. IEEE/ACM Conf. Computer Aided Design* (ICCAD), pp. 450-457, 2002.

[2] M. Barborak, M. Malek, and A. Dahbura, "The Consensus Problem in Fault-Tolerant Computing," *ACM Computing Surveys*, vol. 25, no. 2, pp. 171-219, June 1993.

[3] D. M. Blough and H. W. Brown, "The Broadcast Comparison Model for On-line Fault Diagnosis in Multicomputer Systems: Theory and Implementation," *IEEE Trans. Comp.*, vol. 48, no. 5, pp. 470-493, May 1999.

[4] D. M. Blough, G. F. Sullivan, and G. M. Masson, "Efficient Diagnosis of Multiprocessor Systems under Probabilistic Models," *IEEE Trans. Computers*, vol. 41, no. 9, Sept. pp. 1126-1136, September 1992.

[5] A. Caruso, S. Chessa, P. Maestrini, and P. Santi, "Evaluation of a Diagnosis Algorithm for Regular Structures," *IEEE Trans. Computers*, vol. 51, no. 7, pp. 850-865, July 2002.

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

[7] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Transactions on Automatic Control*, vol 49, no. 9, pp. 1465-1476, September 2004.

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

[9] J. J. Gertler, "Fault Detection and Diagnosis in Engineering Systems," *Marcel Dekker*, New York, 1998.

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

[11] W. Kozlowski and H. Krawczyk, "A Comparison-Based Approach to Multi-Computer System Diagnosis in Hybrid Fault Situations," *IEEE Trans. Computers*, vol. 40, no. 11, pp. 1283-1287, November 1991.

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

[13] S. Malella and G. Masson, "Diagnosable Systems for Intermittent Faults," *IEEE Trans. Computers*, vol. 6, no. 6, pp. 560-566, June 1978.

[14] A. Pelc, "Optimal Fault Diagnosis in Comparison Models," *IEEE Transactions on Computers*, vol. 41, no. 6, pp. 779-786, June 1992.

[15] F. Preparata, G. Metze, and R. Chien, "On the Connection Assignment Problem of Diagnosable Systems," *IEEE Transactions on Computers*, vol. 16, no. 6, pp. 848-854, December 1967.

[16] N. S. V. Rao, "Computational Complexity Issues in Operative Diagnosis of Graph-Based Systems," *IEEE Trans. Computers*, vol. 42, no. 4, pp. 447-457, April 1993.

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

[18] A. Sengupta and A. T. Dahbura, "On Self-Diagnosable Multiprocessor Systems: Diagnosis by the Comparison Approach," *Proc. IEEE Symposium on Fault-Tolerant Computing*, pp. 54-61, 1989.

[19] C. J. Walter, P. Lincoln, and N. Suri, "Formally Verified On-Line Diagnosis," *IEEE Trans. Software Eng.*, vol. 23, no. 11, pp. 684-721, Nov. 1997.

[20] S. Zelinski, T. J. Koo, and S. Sastry, "Hybrid System Design For Formations of Autonomous Vehicles," *Proc. IEEE Conference on Decision & Control*, pp. 1-6, 2003.

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

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

[23] J. P. Desai, J. P. Ostrowski, V. Kumar. "Control of Changes in Formation for a Team of Mobile Robots," *Proc. IEEE Conf. Robotics & Automation*, pp.1556-61, May 1999.