Comparing the Performance of Expert User Heuristics and an Integer Linear Program in Aircraft Carrier Deck Operations

Jason C. Ryan, *Student Member, IEEE,* Ashis Gopal Banerjee, *Member, IEEE,*
Mary L. Cummings, *Senior Member, IEEE,* and Nicholas Roy

***Abstract*—Planning operations across a number of domains can be considered as resource allocation problems with timing constraints. An unexplored instance of such a problem domain is the aircraft carrier flight deck, where, in current operations, replanning is done without the aid of any computerized decision support. Rather, veteran operators employ a set of experience-based heuristics to quickly generate new operating schedules. These expert user heuristics are neither codified nor evaluated by the United States Navy; they have grown solely from the convergent experiences of supervisory staff. As unmanned aerial vehicles (UAVs) are introduced in the aircraft carrier domain, these heuristics may require alterations due to differing capabilities. The inclusion of UAVs also allows for new opportunities for on-line planning and control, providing an alternative to the current heuristic-based replanning methodology. To investigate these issues formally, we have developed a decision support system for flight deck operations that utilizes a conventional integer linear program-based planning algorithm. In this system, a human operator sets both the goals and constraints for the algorithm, which then returns a proposed schedule for operator approval. As a part of validating this system, the performance of this collaborative human–automation planner was compared with that of the expert user heuristics over a set of test scenarios. The resulting analysis shows that human heuristics often outperform the plans produced by an optimization algorithm, but are also often more conservative.**

***Index Terms*—Decision support system, human supervisory control, human–automation interaction.**

I. Introduction

PLANNING operations regarded as resource allocation problems with timing and sequence constraints [1] and [2], in which a set of limited resources must be assigned to a group of agents in a manner that satisfies all temporal deadlines and required task sequences, can be found in a number of environments [3]–[9]. This type of problem is becoming increasingly common in environments utilizing unmanned vehicles, in which a centralized control server assigns tasks for the global system, routing vehicles through the environment so as to optimize any number of factors [10]–[15]. One example would be that of an automated warehouse, such as those operated by Kiva Systems [16] (now part of Amazon.com, Inc.), where unmanned vehicles locate and retrieve stocked merchandise for a human packer. The routing and control of aircraft (manned, unmanned, or both) within an airport runway and terminal system also falls within this class of problem, as does the allocation of hospital operating rooms under high-occupancy, time-critical cases (such as natural disasters).

An additional, though unexplored, domain is that of the aircraft carrier flight deck [17], [18]. Replanning operations in this domain require rapid action on the part of deck supervisors who must balance conflicting directives of maximizing safety and minimizing operational time. Additionally, these directives are confounded by the frequent failures that occur in the system, arising from vehicles, deck equipment, and crewmember actions. These failures are highly varied in both duration and severity, ranging from minor issues a few seconds in duration (a crewmember walking into a restricted area) to major, long-duration failures that severely limit the operational ability of the carrier flight deck (such as the complete disabling of a launch catapult). Replanning the operational schedule must not only address the specific failures within the system but must also create a schedule that provides maximum adherence to the prespecified launch windows for aircraft on the deck. These requirements, as well as the dynamic nature of the flight deck, require that the schedule also be computed in the minimum time possible.

The inclusion of unmanned aerial vehicles (UAVs) in the near future [19] provides a foothold for automating a variety of aircraft actions on the flight deck. Northrop Grumman's X-47B Pegasus UAV will demonstrate point-and-click control, in which the vehicle will receive abstract commands such as "taxi to this catapult and takeoff" [20]. The data links used to transmit information and commands from individual aircraft ground control stations could be modified to communicate directly with a central planning computer capable of managing the tasks of multiple vehicles. With sufficient geospatial data and understanding of the current state of the schedules of all aircraft in the system, an automated planning algorithm would be capable of replanning operations for all

Manuscript received August 23, 2012; revised December 28, 2012 and April 1, 2013; accepted June 13, 2013. This work was supported in part by the Office of Naval Research Science of Autonomy Program under Contract N000140910625. The opinions expressed in this paper are those of the authors and do not necessarily reflect the opinions of the sponsoring agency. Recommended by Associate Editor A. Tayebi.

J. C. Ryan, M. L. Cummings, and N. Roy are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jcryan13@mit.edu; missyc@mit.edu; nickroy@mit.edu).

A. G. Banerjee is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ashis@csail.mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2271694

vehicles simultaneously, perhaps also utilizing the input of an experienced human operator. However, before integration into aircraft carrier operations, a thorough performance assessment is required to ensure that the system performs adequately in terms of safety and schedule effectiveness. Comparing the performance of this futuristic system to current system performance is critical in establishing the value added by such an implementation. This performance comparison is a difficult task for four reasons. First, human supervisors in this domain receive no codified training in replanning deck operations, thus, there are no objective training performance metrics that could form the basis of comparison. Second, little to no data on flight deck operations is available in order to recreate daily operations; no performance assessments are conducted in real time, nor can any be reconstructed post hoc. Third, no simulated testing environments are available to outside researchers to test and compare possible planning strategies. Fourth, no standardized evaluation methods exist for judging the quality of operations on the carrier deck. This paper discusses both the development of an integer linear programming algorithm for planning flight deck operations and the definition of a testing methodology for assessing its performance against current planning operations.

A. *Related Research*

To this point, aircraft carrier flight deck operations have been relatively unexplored as a scheduling problem. However, the related domain of airport ground operations planning has been a prominent area of research. In principle, the two environments are quite similar. Both involve the distribution of limited, stationary resources to mobile vehicles (aircraft) to execute an operating schedule involving specific time windows for task execution. In the aircraft carrier flight deck, this means assigning aircraft to launch catapults, fueling stations, and other elements; for airport operations, these assignments are to runways and departure gates. In both cases, the schedule of tasks that vehicles must complete is known *a priori*; it is only the optimal assignment of tasks to resources and their ordering that must be decided.

There are many instances in which the airport problem has been addressed through the implementation of some form of integer programming [21]–[25]. In these cases, the objective functions commonly address some variants of the total operating time. For instance, Balakrishnan and Jung [21] formulate a cost objective related to the total time on the runway with an aircraft's engine active, while Roling and Visser [22] use an objective function minimizing both the total taxi time and the total holding time (stopped in place). Additionally, objective functions have typically included a minimization of delays [21], [23]–[25], which in these contexts entail completing the schedule in minimum time.

Constraints in these problems typically revolve around consistency issues, ensuring that capacity constraints are not violated, that paths and assignments are feasible, and that maximum queue lengths for runways are not violated. Only one paper considers speed constraints explicitly in their formulation [21], and none consider fuel issues. In flight deck operations, flight times are potentially a hard constraint, as are fuel concerns for approaching aircraft. Additionally, when implementing a system involving a human operator, the inputs and requests of that operator might also take the form of constraints in the plan.

Furthermore, a common tactic in these examples has been to define the system in terms of a networked graph, modeling runway segments as nodes with a certain capacity and including constraints that model the ability of aircraft to taxi through the runway. This is another important difference between airport and aircraft carrier operations—the carrier flight deck is a completely open space, and an advantage of its operations is that motion paths are unconstrained by the borders of roads or runways. In summary, while there has been substantial research in this area for airport ground operations, the explicit formulation of constraints and objectives are similar, but not identical to, those experienced in the aircraft carrier flight deck.

B. *Scope*

This paper describes the development and assessment of an integer linear programming (ILP) algorithm for planning aircraft carrier flight deck operations, comparing its performance to current operations. The performance during current operations was determined by the creation of a proxy decision making model based on interviews with subject matter experts. Through a cognitive task analysis (see [26] for an explanation of the methodology as applied to a different domain), we elicited a set of expert user heuristics employed by supervisors in replanning aircraft carrier operations. We then developed both a simulation of aircraft carrier operations and a set of automated decision support algorithms to generate data for comparison. The aforementioned simulation environment is part of the deck operations course of action planner (DCAP), which is designed to facilitate human-automation collaborative planning for the aircraft carrier deck (see [27] for details on the interface design).

The research presented in this paper compares the performance of the combined human–automation planning using this ILP planner to the set of expert human heuristics in generating schedules for a set of test scenarios. The remainder of the paper is organized as follows. Section II discusses the DCAP simulation environment in order to formulate the planning problem. Section III discusses the characteristics of the ILP planner, while Section IV explains the nine human planning heuristics elicited in the course of this research. Section V specifies the test methodology used to compare the performance of the two planners, while Section VI discusses the results from the test scenarios. Section VII presents the insights obtained from the results, and the concluding remarks and future works are provided in Section VIII.

II. Aircraft Carrier Deck Environment

The simulation environment is intended to replicate flight deck operations on the current fleet of United States *Nimitz*-class aircraft carriers. An image of the simulation environment is shown in Fig. 1. The DCAP simulation is agent-based, with each entity (crew members, aircraft, and stationary deck resources) acting independently based on individually prescribed rules and models. Four different generic aircraft classes are utilized, containing all possible combinations of fast/slow and manned/unmanned aircraft (Table I). The four types of deck resources are labeled in Fig. 1, each operating its own queue while servicing aircraft on the deck. Human crew and other

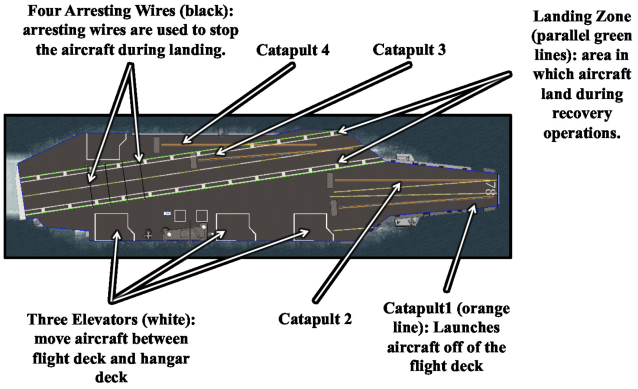
Four Arresting Wires (black): arresting wires are used to stop the aircraft during landing.
Catapult 4
Catapult 3
Landing Zone (parallel green lines): area in which aircraft land during recovery operations.
Three Elevators (white): move aircraft between flight deck and hangar deck
Catapult 2
Catapult1 (orange line): Launches aircraft off of the flight deck

Fig. 1. Screenshot of the aircraft carrier deck. Four types of deck resources are color-coded as follows: catapults (orange), elevators (white), arresting wires (black), and landing zone (green).

TABLE I
TYPES OF AIRCRAFT MODELED IN THE SIMULATION ENVIRONMENT

	Fast (higher flight speed; carries weapons)	Slow (lower flight speed; no weapons)
Manned	Fast Manned Aircraft (FMAC, based on F18 Hornet) Lowest endurance	Slow Manned Aircraft (SMAC, based on E2 Hawkeye) High endurance
Unmanned	Fast UAV (FUAV, based on X-47B Pegasus) Medium endurance	Slow UAV (SUAV, based on the MQ-1 Predator) Highest endurance.

support vehicles are also modeled, but primarily are used in supporting the actions of aircraft and resources.

Activity on the aircraft carrier deck is driven primarily by the tasks required for the active aircraft, such that all other entities (including crew) can be considered as resources to be utilized. When creating a schedule of operations on deck, the supervisor must define an ordered assignment of aircraft to launch catapults. When executing the schedule, aircraft and resources work simultaneously to execute tasks in the world. In general, aircraft begin at a parked state, where they must be fueled and loaded with weapons, and then taxi to a catapult and launch. Afterwards, they fly to a mission area away from the carrier deck, remain there for a duration of time, then return to the carrier. Once doing so, they enter a landing pattern (the Marshal Stack) to wait for landing clearance. In executing the schedule, aircraft, crew, support vehicles, and resources operate simultaneously and in the same shared space on the flight deck. This requires a variety of constraints, including constraints on movement across the deck (to avoid collisions) and on launches (adjacent catapults cannot launch simultaneously), which complicate schedule execution. Making precise *a priori* predictions as to when and how vehicles will interact and when delays may occur is nearly impossible, meaning that any estimates on the time to complete a task may be highly inaccurate.

Additionally, failures in the system affect both resources and aircraft, further complicating operations on the deck. An aircraft fuel or hydraulic leak may limit the operational time of the vehicle, requiring a new schedule of operations that ensures the survival of the aircraft. A deck resource failure may prevent the system from executing the current schedule of operations (e.g., it is unable to launch all aircraft from the deck under their current catapult assignments), requiring a complete reassignment of aircraft amongst the remaining operational resources. Failures like these are fatal to the current schedule and require a new operating schedule. In creating new schedules of operations, human planners have a primary goal of maximizing the rate at which aircraft are launched and recovered from the deck. In general, this is achieved by creating a schedule that requires the minimum absolute time to execute while also minimizing the required working time of crew, aircraft, and support resources. Human planners also must replan quickly, as these environments are time-critical and require fast reaction in order to ensure system performance and safety.

A simulation model of this environment should reflect the variability of the real world. As such, task completion times and failure durations (as well as a variety of other aspects of the environment) were defined as Gaussian distributions with a specified mean and standard deviation. When a task is executed in the simulation, the completion time is randomly sampled from the relevant distribution, and thus multiple replications of a task will have varying completion times. However, there is no published data on flight deck operations that can be used in generating these models. As such, task and failure models for DCAP were developed from a combination of other sources—interviews with flight deck crew, observations of training operations at a Naval training school for flight deck crew, reviews of videos of flight deck operations, and a visit to an aircraft carrier to observe live operations. In some cases, tasks were timed; in other cases, operators were asked for their estimates of completion times for various tasks, with similar actions taken for failure models.

Because of the absence of detailed published data, no formal simulation validation was performed. However, the simulation was reviewed by various Naval staff throughout its development and deemed an accurate representation of reality. After the development of the simulation environment, it was integrated with the ILP planner and with additional display elements to facilitate user replanning of the system. Two different replanning options were implemented: the first employs a manual planning interface, while the latter utilizes the ILP planner. The manual planning interface involves an expert user applying the internal heuristics used by operators in replanning; these heuristics will be discussed later in Section IV. The human–automation collaborative interface utilizes the ILP, in conjunction with a human operator specifying high-level goals for the planner. The next section addresses the ILP planning algorithm, which minimizes a cost function with explicit terms.

III. CHARACTERISTICS OF THE ILP PLANNER

The current automated planner in use with the DCAP system is an ILP that minimizes the overall weighted time taken by the fleet of aircraft to complete a given sequence of tasks. The planner satisfies a set of hard constraints imposed by the availability of resources to complete any task at a given time instant, as well as safety requirements and other physical restrictions as described in the previous section. The planner also accounts for all the soft constraints by including violation margins. These soft constraints are specified by the preferences of the human users to complete certain tasks for specific aircraft or all the tasks for airborne and deck aircraft within specified times (one example of a soft constraint would be where the human operator specifies that a vehicle's new schedule should delay its original launch time by 10 min). The planner is integrated with the simulation environment, such that when a user requests a new plan, state information for the environment is sent to the planner for computation. The computed plan is then sent to the operator for review before execution. While other planning methods, such as metaheuristics, temporal logic, or constraint programming can also be used, we choose ILP to obtain a feasible solution in real time that satisfies all types of hard and soft constraints without requiring appropriate selection of key algorithm parameters. Furthermore, the obtained ILP solution is empirically validated to be close to the global optimum.

We follow the integer programming formulation II given in [2] to deal with temporal dependencies of tasks and irregular task starting time costs. The binary decision variables $\mathbf{y_{ijk}}$ are equal to 1 if aircraft $i \in \{1, \dots, I\}$ begins task $j \in V$ at time step $k \in \{1, \dots, K\}$, and 0 otherwise. Each aircraft has a user-specified priority $\mathbf{p_i} \in \mathbb{R}_+$, and a set of tasks $\mathbf{V_i} = \{j_1, \dots, j_{J_i}\}$ that it needs to accomplish. A task can be of two types: one that requires available deck resources such as landing, launching, parking, and refueling, and another that does not need any resource such as taxiing, going to mission, or approaching the deck. Moreover, the first type of task can be either resource specific such as landing, launching, and refueling as it matters which catapult or refueling station is serving an aircraft, or resource agnostic such as parking as it is not important which parking spot is used. $\mathbf{V_{nr}}$, $\mathbf{V_{rs}}$, and $\mathbf{V_{ra}}$ represent the set of nonresource requiring, resource specific, and resource agnostic tasks respectively. Every task j also belongs to a task type set $\mathbf{E_j}$, which is identical to j if it is a nonresource requiring task, and includes all the resource requiring tasks associated with the same category such as launching or refueling otherwise. Note that only one resource specific task of a particular type is present in V_i. The initial choice is user defined, but then it may be altered by the optimizer during the course of the operation. $\mathbf{V_s}$ denotes the set of resource specific task pairs that cannot be performed simultaneously due to safety reasons and is independent of the aircraft type. These tasks include landing and launching from either catapult 3 or 4, and launching from both catapults 1 and 2, or both 3 and 4. The temporal dependencies of the tasks for each aircraft is represented by a weighted digraph $\mathbf{G_i} = (V_i, A_i)$ with $\mathbf{A_i} = \{(m_i, n_i) : d_{m_i n_i} > -\infty\}$, where $\mathbf{d_{m_i n_i}}$ is the time lag such that the starting time for task n_i, $\mathbf{S_{n_i}}$, is correlated to the starting time for task m_i, $\mathbf{S_{m_i}}$, by an inequality of the form $S_{n_i} \geq S_{m_i} + d_{m_i n_i}$. The objective function is the sum of the weighted completion times of all the aircraft tasks where each completion time is weighted by the aircraft priority. Note here that minimizing the completion time of the last task, i.e., the mission duration, would not have accounted for the varying importance of the different tasks.

The ILP is then written as

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{k=1}^{K} p_i(t_{ij} + k\Delta_t) y_{ijk} \tag{1}$$

subject to the constraints

$$\sum_{k=1}^{K} y_{ij_ik} = 1, \quad \forall i \quad \text{and} \quad j_i \in V_i \cap (V_{nr} \cup V_{ra}) \tag{2}$$

$$\sum_{j_i \in E_{j_i}} \sum_{k=1}^{K} y_{ij_ik} = 1, \quad \forall i \quad \text{and} \quad j_i \in V_i \cap V_{rs} \tag{3}$$

$$\sum_{s=k}^{K} y_{im_is} + \sum_{s=1}^{k+d_{m_in_i}-1} y_{in_is} \leq 1, \quad \forall i, k \quad \text{and} \quad (m_i, n_i) \in A_i \tag{4}$$

$$\sum_{i=1}^{I} y_{ijk} \leq N_{jk} \quad \forall k \quad \text{and} \quad j \in V_{rs} \cup V_{ra} \tag{5}$$

$$\sum_{i=1}^{I} (y_{ij_lk} + y_{ij_mk}) \leq 1 \quad \forall k \quad \text{and} \quad (j_l, j_m) \in V_s \tag{6}$$

$$\sum_{k=1}^{S_{ci'}} y_{i'j_lk} = 1, \quad \forall i' \in I_c \tag{7}$$

$$\sum_{k=1}^{S_{pi''j''}+T_m} y_{i''j''k} = 1, \quad \forall i'' \in I_p, \quad j'' \in V_{i''}. \tag{8}$$

Here, $\mathbf{t_{ij}}$ is the estimated time to complete task j by aircraft i and $\mathbf{N_{jk}}$ denotes the number of available deck resources to perform task j at time k. N_{jk} is always binary for a resource specific task and is less than the total number of resources for a resource agnostic task. $\mathbf{t_{ij}}$ is either taken as the deterministic value for task durations such as mission execution or the mean of the Gaussian distribution that models the processing time for any deck resource. The deck resource processing time also includes the expected arrival time for the required crew members. Thus, (2) and (3) together enforce that every task has to be started exactly once by each aircraft with the additional requirement of performing only one resource specific task of a particular type being imposed by (3). Equation (4) enforces temporal precedence conditions during task execution, (5) ensures that only the available number of deck resources are used to perform any resource requiring task, and (6) ensures that no two tasks that might lead to safety issues if performed concurrently are started together.

$\mathbf{I_c}$ and $\mathbf{I_p}$ denote the set of critical aircraft that have developed fuel or hydraulic leaks and the set of aircraft with user-specified preferences, respectively. $\mathbf{S_{ci'}}$ is the permissible time limit within which a critical aircraft i' should start landing task j_l, $\mathbf{S_{pi''j''}}$ is the user-stipulated time limit to complete task j'' for the aircraft $i'' \in I_p$, and $\mathbf{T_m}$ is the allowable time margin for violating any of the aircraft task preference constraints. Note that all the actual time values are conservatively converted to integral values by taking the interval between two successive time instants $(\mathbf{\Delta_t})$ as a constant. Also note that if our initial value of T_m, which is assigned at the onset of simulation before the set I_p is populated, results in an infeasible solution, we increase it until the problem has optimal solutions.

The ILP was implemented in C++ using g++ as the compiler. The aircraft priorities were assigned on an integer scale of [1–5], each with an initial priority of 3. While replanning, all the critical aircraft were reassigned a priority value of 5 (for more details on the replanning process, see [27]). The total planning time horizon was selected based on the worst-case estimate of completing all the tasks by assuming the sum of the mean and three sigma values for stochastic task durations and scenario that only one deck resource would be available to perform a task at any time instant. For a particular test scenario, all the constraints were initialized once in the planner code, except for the ones that would change dynamically during the course of any operation, namely, the aircraft that

TABLE II
TABLE OF EXPERT USER HEURISTICS

General	*Deck*	*Airborne*
(1) Minimize changes (2) Cycle aircraft quickly, but maintain safety (3) Halt operations if crew or pilot safety is compromised	(4) Maintain an even distribution of workload across the deck (5) Make available as many deck resources as possible (6) When moving aircraft on deck, maintain orderly traffic flow through the center of the deck	(7) Populate Marshal Stack according to fuel burn, fuel level, then miscellaneous factors (8) Park aircraft for maximum availability next cycle (9) "True" vs. "Urgent" Marshal Stack emergencies

would be considered as critical, tasks with user-preferred time limits, and deck resource failures. This initialization facilitated fast replanning by modifying, adding, or deleting only a few constraints.

We used heuristic-guided, depth-first branch and bound search to solve the ILP. The search heuristics were comprised of the following three rules. First, we assigned tasks to all the critical aircraft at the earliest possible time instants giving preference to the ones with more remaining tasks and breaking ties for equal number of tasks randomly. Second, if no critical aircraft were present, tasks were assigned at the earliest possible time instants to the aircraft with the maximum number of tasks, then to the one with the second most number of tasks, and so on, again breaking ties randomly. Third, deck resources were allocated such that all the available resources for a particular task type would have as equal of a workload as possible, where workload was defined as the total number of aircraft that would utilize a resource in the generated plan. All the rules were employed in the same order as described above to fix integer values to the decision variables during branching. While applying the first two rules were straightforward in terms of deciding which decision variables set to unity, implementing the third rule was more involved. It required removing the minimum number of tasks from every resource (i.e., setting the corresponding decision variables to zero) until all the resources for that task type had an equal number of assigned aircraft over the entire planning horizon. Bounding was done by simply comparing the obtained solution with the incumbent (current best) solution. The search was terminated when a feasible solution was obtained and the computation time exceeded half a second. The second termination criterion was imposed based on the need to generate plans in real time. While a theoretical analysis of the approach to provide performance guarantees is beyond the scope of this paper, empirical comparison with a widely used optimizer in Section VI establishes its utility.

IV. EXPERT USER HEURISTICS

Over two dozen experienced Naval personnel were consulted throughout the design process of the DCAP system, encompassing former Naval aviators, instructors at a Navy training base for deck crew, a former member of an Air Wing Commander's planning staff, and two commanders of a Navy training base for deck crewmen. In meetings that occurred in person, participants were presented with example scenarios that could occur in real-life operations and were asked what their responses to the situations would be. Despite the lack of standardized training for replanning carrier operations [17], these guided interviews allowed the DCAP research team to identify a relative consistency in solution generation. This consistency was captured in a set of heuristics: the rules of human decision-making shaped by experience and used to simplify the aircraft carrier planning problem and generate approximate solutions quickly [28]–[30]. The list of heuristics appears in Table II, grouped according to three general categories (general, deck, and airborne) but not necessarily in order of importance.

General heuristics are applicable to all replanning scenarios. The general heuristics include minimizing changes in the schedule (heuristic 1), minimizing time but without jeopardizing safety (heuristic 2), and ceasing operations if any human being is placed in immediate danger (heuristic 3). For deck heuristics, the concerns are to balance the workload on the deck (heuristic 4) due to concerns of crew fatigue and the maintainability of the deck equipment, to ensure maximum flexibility in operations by keeping as many resources available as possible (heuristic 5), and keeping orderly motion on the deck by funneling traffic movements to the interior of the deck (heuristic 6). Airborne heuristics deal with the ordering of aircraft in the landing order (heuristic 7), where they should be parked after landing (heuristic 8), and how to handle failures for airborne aircraft (heuristic 9). Applying heuristic 9 to an airborne aircraft requires understanding the nature of the failure and its criticality. True emergencies endanger the pilot and the aircraft and must be dealt with immediately. Urgent emergencies are of concern, but if compensating for these failures causes further schedule degradation or requires numerous changes on deck, operators may delay action until a more satisfactory time.

These expert heuristics were reviewed by the previously interviewed Naval personnel in the form of a teach-back interview [31]. In this form of interview, the interviewees were presented with a problem scenario, to which the interviewer applied the heuristics in question. The interviewer described the heuristics and what their resulting plan would be. The interviewee then validated the proposed action, possibly suggesting further details or a slight differentiation in the heuristic. The final set of heuristics allows a nonexpert user to generate approximately the same solutions as a more experienced subject matter expert. However, it must be remembered that these are simply rules of thumb for generating a schedule; how, exactly, these lead to better performance in the schedule are not necessarily straightforward, due to the complexity of interactions on the flight deck.

V. METHODOLOGY

In order to determine whether or not the DCAP planning system with the embedded ILP provided substantial benefit over the current human-only methods of replanning, we conducted a series of tests. The testing program involved three different testing scenarios of varying complexity, applied to three different planning scenarios. For two of the three planning conditions—the human heuristics (HH) and ILP conditions—a fatal failure necessitating a schedule replan occurred at the same prespecified time and affected the same aircraft or deck resource. Under the HH planning condition, a human operator utilized only the expert heuristics to manually create a new schedule to overcome the failures. In the ILP planning condition, the same human operator utilized the ILP planner within the DCAP environment to create new schedules. The third planning condition, the baseline (no failure) case, involved no failures and no replanning and serves as a reference point for performance under the original schedule.

Due to the randomized nature of task durations and the various constraints on task execution (as described in Section II), the simulation is nondeterministic—multiple replications of a single scenario will vary not only in the total time to execute the schedule, but also in the order in which aircraft arrive at catapults. As such, each of the three scenarios was tested thirty times for each of the three different planning conditions. For each scenario, the system was given the same initial state and schedule across all planning conditions and replications. Replanning actions occurred in real time: a single operator monitored the system from the outset, waiting for the failure to occur. Once it did, the operator engaged the replanning system (either the HH or ILP planner) to generate a new schedule of operations for the system. While replanning occurs, the simulation continues to execute—aircraft affected by the failure may not be able to execute tasks, but unaffected aircraft continue operating.

All the testing was performed on a Lenovo Thinkpad W500 laptop (2.80 GHz Intel Core 2 Duo T9600 CPU, 8 GB RAM, Ubuntu 9.10 64-bit operating system). The DCAP system (a Java$^{\mathrm{TM}}$ application) was run within the Eclipse Galileo Java$^{\mathrm{TM}}$ IDE. The DCAP system and the planning modules communicated via the LCM package, details of which are available in [32]. The LCM package allowed for real-time state information passing with negligible time delay, as demonstrated in the MIT DARPA Urban Challenge vehicle. Data were extracted through automated logging features embedded in the DCAP source code. Data were logged upon scenario termination, which was also automated to ensure no variation in end conditions.

A. Test Scenarios

Three test scenarios, designated simple, moderate, and complex, were used to represent different levels of complexity in the operational environment, as defined by the number of heuristics applied in replanning. The scenarios are detailed below.

1) *Simple Scenario:* The simple scenario models the occurrence of a catapult failure on deck during launch operations. Twenty aircraft (2 SMAC, 2 SUAV, 12 FMAC, and 4 FUAV) are fueled and have weapons loaded while parked on the deck. Aircraft then proceed to launch catapults, queuing in lines of no more than three at each launch catapult at any given time, similar to real operations. Aircraft launch assignments are initially distributed across catapults. Catapult 1 remains inaccessible for the entirety of the scenario due to several aircraft parked in the immediate vicinity. After launching from the catapult, aircraft proceed to a mission area. For the HH and ILP planning conditions, catapult 3 is disabled at a set time after simulation start (identical across both planning conditions and all replications). This failure is fatal to the schedule, as catapult 3 cannot be used for the remainder of the simulation.[1] Replanning after this failure should address the reassignment of aircraft to the remaining accessible and operational catapults. The scenario terminates when all aircraft have departed the carrier deck and reached their mission locations.

This scenario is identified as simple as this is a relatively common problem and replanning for the system requires the application of only four expert user heuristics: Heuristics 1, 2, 4, and 6. Application of the heuristics involves moving all aircraft from the failed catapult 3 forward to catapult 2 while also attempting to balance the number of aircraft at the two remaining functional catapults (catapults 2 and 4). As described in Section III, the ILP selects actions that minimize the overall mission duration for the system. It does not make considerations as to the number of changes in the schedule (heuristic 1), nor does it necessarily seek a balanced set of launch assignments (heuristic 4). The resulting plan may demonstrate these characteristics, but these are not the specified criteria in replanning. Additionally, the ILP makes no consideration of the locality of aircraft to launch catapults and, thus, may also act in opposition to heuristic 6. However, there are no guarantees that the HH plan is optimal in any sense—it simply follows the rules as specified, given the current state of the deck.

2) *Moderate Scenario:* The moderate scenario models a recovery (landing) task. In this scenario, all aircraft begin at their mission locations and immediately begin transiting to the landing pattern (known as the Marshal Stack). This scenario again uses twenty aircraft (2 SMAC, 2 SUAV, 12 FMAC, and 4 FUAV), timed to enter the landing pattern with a very tight spacing and an order based on heuristic 7. FMAC aircraft enter first, followed by SMAC aircraft, then FUAV aircraft, then SUAVs. Two failures are introduced just before aircraft enter the Marshal Stack—an FMAC has a hydraulic failure while an SMAC has a fuel leak. These aircraft were chosen based on the expected response of human operators. For a fuel leak, the HH plan would dictate that the aircraft should be moved forward in the Marshal Stack (so that it lands earlier). For a hydraulic leak, the aircraft should be moved back in the Marshal Stack to delay its landing. Typically, FMACs enter the landing pattern first and SMACs last; thus, an FMAC experiencing a hydraulic leak should lead to significant change in landing order, as should an SMAC with a fuel leak. In both the cases, however, actions should be selected that ensure that both aircraft land before encountering a limit violation on their hydraulic fluid and fuel, respectively.

Replanning for this scenario invokes five heuristics: 1, 2, 3, 7, and 9. In this case, applying the heuristics results in the movement of the SMAC (fuel leak) forward in the Marshal Stack to minimize the risk of this aircraft running out of fuel. However, a hydraulic failure increases the possibility that the landing strip (LZ) will be disabled after landing, in that the aircraft may be unable to vacate the LZ or may generate debris if it is damaged during a hard landing. In either case, a time penalty is incurred as crew must work to clear the LZ and prepare it for the next landing. Moving the FMAC backwards in the Marshal Stack allows for additional aircraft to land, and thus minimizes the potential for delays. These probabilities are not directly considered by the ILP; rather, the ILP will select actions that minimize mission duration while ensuring that the hard constraints on minimum fuel and hydraulic fluid levels are not violated. The ILP may defy heuristic 9 by not moving the

[1] Because this is a fatal failure for the schedule, the failure is not implemented for the baseline planning condition, allowing it to execute without requiring replanning.

TABLE III
METRICS USED IN THE TESTING PROGRAM

Metric	*Abbreviation*
Fuel Violations	FV
Landing Zone Foul Time	LZFT
SMAC #2 Recovery Time	SMAC 2 RT
FMAC #6 Recovery Time	FMAC 6 RT
Mission Duration	MD
Total Aircraft Active Time	TAAT
Total Crew Active Time	TCAT
Total Aircraft Taxi Time	TATT
Wait Time in Queue-Marshal Stack	WTQMS

fuel leak aircraft to the first available position, but it will still schedule the aircraft to land with a fuel level greater than the critical value. For the baseline case, no replanning occurs and all aircraft land as specified in the original order. Comparing the ILP and HH planners to the baseline thus reveals the differences in measures of fuel consumption and flight time for the true and urgent emergency cases in both the HH and ILP planner.

3) *Complex Scenario:* The two previous test scenarios focus only on one aspect of the launch or recovery of aircraft in the aircraft carrier environment. The Complex scenario focuses on both aspects, addressing a case where emergency launches are requested in the midst of landing operations. This scenario begins similarly to the moderate scenario, with eighteen (rather than twenty) aircraft (2 SMAC, 10 FMAC, and 6 FUAV) returning from mission. For this scenario, the aircraft are given much lower fuel values than in the moderate case, increasing the importance of accurate placement of aircraft in the landing order. As such, the initial order of entry into the landing pattern is slightly different from that of the moderate scenario—FUAVs enter the Marshal Stack first, followed by FMACs and SMACs. In the midst of return operations, a supervisor requests the launch of additional reconnaissance aircraft. To satisfy the supervisor's request, two additional SUAVs must launch from the flight deck. In launching these aircraft, only catapults 2, 3, and 4 are available (just as in the simple scenario, catapult 1 is inaccessible due to parked aircraft). Just as these two aircraft begin to launch, a fuel leak arises in an SMAC arriving in the landing pattern. This creates conflicting priorities for scheduling—a supervisor has requested that the SUAVs be launched immediately, but the SMAC leaking fuel must also be landed relatively quickly. Using catapults 3 and 4 to launch the requested reconnaissance aircraft may lead to conflicts with aircraft incoming to land.

Replanning for this scenario requires the application of seven heuristics: 1, 2, 3, 4, 6, 7, and 9. The HH solution requires moving the failed SMAC forward in the landing order (to minimize the chance of running out of fuel) and sending the launching aircraft to catapult 2 only. Utilizing only this catapult ensures that, regardless of the time required to launch, aircraft on approach do not experience any interference in the landing strip area. In this case, efficiency in launching is sacrificed to minimize the risk for the airborne aircraft. The ILP selects actions that provide for shortest time until launch for the two support aircraft while also balancing the constraints on the incoming aircraft. Based on the planner's estimates of how quickly aircraft can launch from the deck, the resulting assignments for the two launching aircraft may vary substantially. If the planner predicts that both may launch from catapults 3 and 4 before the first aircraft lands, this assignment may be given; if it believes this is impossible, both aircraft may be sent to the forward catapults. Just as in the moderate scenario, no replanning occurs for the baseline case and all aircraft land as specified in the original order. Again, comparing the ILP and HH planners to the baseline thus reveals differences in measures of fuel consumption and flight time for the true and urgent emergency cases for these planners.

B. Measurement Metrics

As noted earlier, the U.S. Navy does not perform any routine assessments of flight deck operations and provide no detailed metrics on assessing schedule performance. As such, we must create our own metrics for assessing operations. In terms of replanning strategy, both the human heuristics and the ILP have the same goals: simultaneously minimizing total mission duration and workload for crew and aircraft on the deck. Additionally, a major focus of replanning is to maximize the safety of the aircraft, primarily in that aircraft should be landed with fuel greater than their minimum emergency level. This is partially influenced by the amount of time the landing zone is occupied, as occupying the landing zone prevents airborne aircraft from landing, and thus increases the likelihood of an aircraft burning down to its emergency fuel level.

For this experiment, nine metrics were developed to address these issues, falling primarily into two classes: safety and time (Table III). In terms of safety, fuel violations (FV: incremented by one each time an aircraft drops below a certain fuel threshold) and landing zone foul time (LZFT: the time in which any entity is in the landing zone area, summed across all entities) relate to the safety of aircraft in the system. Two other metrics (SMAC 2 and FMAC 6 recovery times) address the safety of the two aircraft that experience fuel leak failures in the scenarios. Measures of time concern the effectiveness of the replanning system, including measures of the total mission duration (MD) and measures for workload of the crew [total crew active time (TCAT)] and aircraft [total aircraft active time (TAAT)]. Two additional measures were defined to examine the efficiency of the plans generated by the HH and ILP—total aircraft taxi time [the time aircraft spend taxiing on deck (TATT)], wait time in queue-Marshal Stack [the time aircraft spend waiting to land (WTQMS)].

In examining the performance of the three planning conditions against one another, results were compared for a single metric within a given scenario for each pairwise combination of planning conditions. The distribution of results for each metric from each scenario/planning condition combination were first checked for normality (using the Kolmogorov–Smirnov test) and heteroskedasticity (using the Levene test). If both distributions in the pair were normal, a two-sided ANOVA was used; otherwise, a two-sided nonparametric Mann–Whitney U test was employed. These statistical methods test the same null hypothesis (H_0: the distributions are identical) to determine whether or not the distributions are identical. To preserve a family wise error rate of 0.05 for

TABLE IV

COMPARISON BETWEEN HEURISTIC AND OPTIMAL ILP SOLVERS

Scenario	*Heuristic ILP computation time*	*COIN-OR Cbc computation time*	*% sub-optimal soln. by heuristic ILP*
Simple	0.5 ± 0	1.58 ± 0.07	1.67 ± 0.04
Moderate	0.31 ± 0.03	0.34 ± 0.02	0 ± 0
Complex	0.5 ± 0	3.47 ± 0.15	2.08 ± 0.05

TABLE V

RESULTS OF STATISTICAL TESTING FOR THE SIMPLE SCENARIO. BOLDED RESULTS SIGNIFY SIGNIFICANCE AT $\alpha = 0.002$, NONPARAMETRIC MANN–WHITNEY U TESTS WERE USED FOR EACH COMPARISON

Metric	*B vs. HH*		*B vs. ILP*		*HH vs. ILP*	
	p-value	Superior	p-value	Superior	p-value	Superior
FV	-	-	-	-	-	-
LZFT	**p=0.000**	B	p=0.046	=	**p=0.002**	ILP
SMAC 2 RT	-	-	-	-	-	-
FMAC 6 RT	-	-	-	-	-	-
MD	**p=0.000**	B	**p=0.000**	B	p=0.745	=
TAAT	**p=0.000**	B	**p=0.000**	B	**p=0.000**	HH
TCAT	**p=0.000**	B	**p=0.000**	B	**p=0.000**	HH
TATT	**p=0.000**	B	**p=0.000**	B	p=0.081	=
WTQMS	-	-	-	-	-	-

statistical tests in each test scenario, p values of 0.002 were used for each ANOVA and Mann–Whitney U test.

VI. RESULTS OF EMPIRICAL TESTING

We first present a comparison of our heuristic ILP solver with COIN-OR Cbc [33] v2.0, a widely used off-the-shelf optimizer, to demonstrate the utility of our solver in providing good quality solutions in significantly less time. Based on 30 runs, Table IV shows that the heuristic solver is always able to compute the optimal solutions for the moderate scenario (where fewer number of tasks are assigned to the aircraft), and generates solutions that only deviate from the true optimal by less than 1.71% and 2.13% for the simple and complex scenarios, respectively. Furthermore, the heuristic solver is always terminated within 0.5 s, whereas the optimizer takes more than 1.5 and 3.3 s for the simple and complex scenarios, respectively. Keeping in mind that real-time performance is the key requirement in our problem domain[2], we adopt the heuristic solver for the remaining tests.

We now discuss the results of statistical testing within each scenario to compare the performance of all the three planning conditions [baseline (B), HH-only, and ILP-supported] using the metrics presented in Section V-B.

A. Simple Scenario

The primary differences in planner performance for the simple scenario are highlighted by five of the nine metrics described in Section V-B (the four unused metrics are appropriate only for recovery operations and are thus not applicable to this testing scenario). Figs. 2 and 3 show the resulting distributions of values for each of the test cases. The boxplots in these (and all subsequent figures) show the median value as a horizontal red line, with the diagonal lines emanating from the median denote the 95% confidence interval of the median. Horizontal blue lines above and below the median signify the 25th and 75th quartiles, respectively. Outliers (> 90th percentile, < 10th percentile) appear as red asterisks, with the horizontal black lines (connected by the dashed line whiskers) signifying largest nonoutlier values. Generally, if the 95% confidence intervals (the diagonal lines) between two distributions in any figure overlap, these are likely to be shown as having no statistical difference. Nonoverlap (for instance, the LZFT values for the B and HH conditions in Fig. 2) typically, but not always, signifies a significant statistical difference. Table V presents a compilation of data for the statistical tests on the remaining metrics.

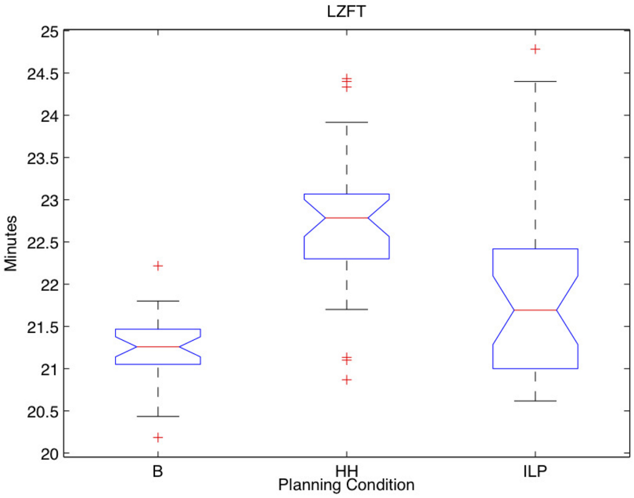

Fig. 2. Boxplot of landing zone foul time (LZFT) values for the simple scenario. Red lines indicate the median value, center blue box indicates interquartile range, and whiskers extend to largest nonoutlier points.

Analysis of the data revealed that, qualitatively, the HH and ILP planners implemented similar plans. For both planning conditions, the returned plan divided the aircraft evenly between the two remaining operational catapults. This explains the similarity in values for LZFT (Fig. 2), TATT, and MD (Fig. 3). The difference in the workload measures, however, is an interesting result. Even though the total MD value was not statistically different between the HH and ILP planners (Table V), the ILP planner showed significant increases in both TAAT and TCAT (measures of aircraft activity on deck). This is likely due to the fact that the ILP planner does not consider the actual locations of the different crew members and instead uses expected values, which significantly affects the processing times at the deck resources. The human heuristics, developed through years of experience, have incorporated an understanding of crew movement and requirements into its rule base. While beneficial to the HH planner, the use of crew members as active agents in the combinatorial optimization process in the ILP planner would render the problem computationally intractable for real-time scheduling. An alternative machine learning-based approach to alleviate this issue is discussed as a part of future work in Section VIII.

The data in Table V also shows that there was only one instance where the ILP or HH planner showed performance equivalent to the baseline: the ILP versus B LZFT comparison.

[2] In future, the planner would also need to account for deck crew members, cyclic or repeated operations, and movement of aircraft between the surface and hangar decks, which would further increase the ILP problem size, and, consequently, the computation time of exact optimizers, rendering them practically even less useful.

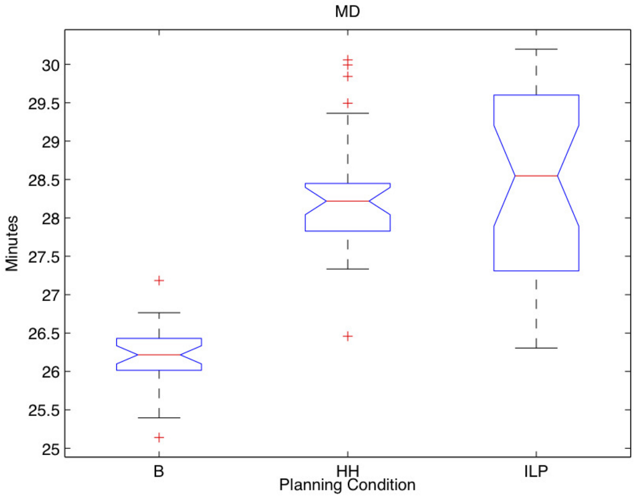

Fig. 3. Boxplots of mission duration (MD) values for the simple scenario. Red lines indicate the median value, center blue box indicates interquartile range, and whiskers extend to largest nonoutlier points.

The fact that the baseline performs well against the HH and ILP planners is not unexpected, as the failure of catapult 3 was fatal to the initial schedule and significantly impairs flight deck capabilities. The baseline should thus be able to launch aircraft more quickly and achieve lower values of LZFT than either the ILP or HH case. It is interesting that the ILP planner was able to reach performance equivalent to the baseline, through the development of plans that moved aircraft away from the failed catapults quickly, even though this is not an explicit goal in its optimization.

The results from this scenario also imply two other important facts: the system itself may be resilient and the human heuristics perform quite well in replanning for the environment. Regarding the former, despite the increases in TCAT and TAAT values, the ILP plans were statistically equivalent to the baseline in terms of LZFT and to the HH planner in terms of the overall mission duration. The increases in aircraft activity were not meaningful in terms of overall system efficiency. Additionally, the actual cost of increases in TAAT and TCAT is minimal. Increased TAAT results in more fuel consumption, but this is not of great concern to the stakeholders. Large increases in TCAT are problematic as they increase the fatigue of the crew, which increases the chances of an error or an accident occurring. However, the average increase in the time of activity for 100+ person crew is minimal (≈ 1 min per person). Regarding the performance of the human heuristics, the HH plans were never shown to be inferior to the ILP plans. In the five metrics reviewed here, the ILP was, at best, statistically equivalent to the HH planner. This suggests that, despite the relative simplicity of these rules, their effect is powerful.

B. Moderate Scenario

In examining the performance of the planners in the moderate scenario, mixed results were seen between planning conditions. The ILP planner outperformed the HH planner in measures that addressed global performance (such as mission duration and total aircraft active time), while the HH planner maintained superior performance in measures addressing the high priority fuel leak aircraft (measures for SMAC #2).

TABLE VI
RESULTS OF STATISTICAL TESTING FOR THE MODERATE SCENARIO

Metric	*B vs. HH*		*B vs. ILP*		*HH vs. ILP*	
	p-value	Superior	p-value	Superior	p-value	Superior
FV	-	-	-	-	-	-
SMAC 2 RT	**p=0.000**	HH	**p=0.000**	ILP	**p=0.000**	HH
FMAC 6 RT	**p=0.000**	B	**p=0.000**	B	**p=0.000**	ILP
MD	**p=0.000**	B	**p=0.000**	B	**p=0.000**	ILP
TAAT	†p=0.132	=	**p=0.000**	B	**p=0.000**	ILP
TCAT	**p=0.000**	=	†p=0.021	=	†**p=0.000**	HH
TATT	**p=0.000**	HH	†p=0.032	=	†**p=0.000**	HH
WTQMS	**p=0.000**	B	**p=0.000**	B	**p=0.000**	ILP

Bolded results signify significance at α = 0.002. All statistical tests nonparametric Mann–Whitney U tests, except where indicated by an †(where a parametric ANOVA was used).

The following paragraphs present the results of the statistical testing and a discussion of the results.

For the moderate scenario, seven metrics were useful in revealing interesting characteristics about planner performance (Table VI). The two remaining metrics (fuel violations and landing zone foul time) showed no variations between planning conditions. Testing revealed a mixture of performance in all three planner pairings; there were no cases where one planner dominated the other in terms of performance. Unlike in the simple scenario, the baseline case was not expected to dominate the majority of measures due to the nature of the failure. In the simple scenario, the failure of the catapult was immediately fatal to the initial schedule in the system; without replanning, aircraft at catapult 3 would never launch. For the moderate scenario (as well as the complex scenario), the failures do not impair the ability of the schedule to complete, instead affecting two individual aircraft (SMAC 2 and FMAC 6) that must land before hitting their critical fuel and hydraulic fluid measures, respectively. These failures were allowed to occur to provide a comparison point for the recovery time (RT) of the failed aircraft. A comparatively smaller RT value implies that the failed aircraft landed earlier.

The first important finding in this scenario was that the HH and ILP planners again developed qualitatively similar schedules. Both the HH and ILP planners moved SMAC #2 (fuel leak) forward in the landing order, as demonstrated by lower values of SMAC 2 RT as compared to the baseline (Fig. 4). Also, both the HH and ILP planners moved FMAC #6 (hydraulic leak) backwards in the landing order, signified by larger values of FMAC 6 RT as compared to the baseline (Fig. 5). The difference lies in the magnitude of movement instituted by the planners. Comparing FMAC 6 and SMAC 2 RT between the HH and ILP planners reveals that the HH planner moved each aircraft to a greater degree than the ILP planner (the ILP moved the SMAC forward 1 slot in the landing order as opposed to 11 slots in the HH plan, while moving the FMAC backwards six slots as opposed to seven in the HH plan). This is to be expected; the HH planner will act to minimize the time the fuel leak-stricken SMAC spends in the air, while maximizing the time for the hydraulic leak-stricken FMAC without breaching its emergency threshold. The second important result concerns total mission duration, where the ILP planner (54.4 ± 0.4 min) completed the mission in less time than the HH planner (55.5 ± 0.4 min). The diagnostic measure WTQMS supports this view, showing that the ILP planner required aircraft to be in the landing pattern for less

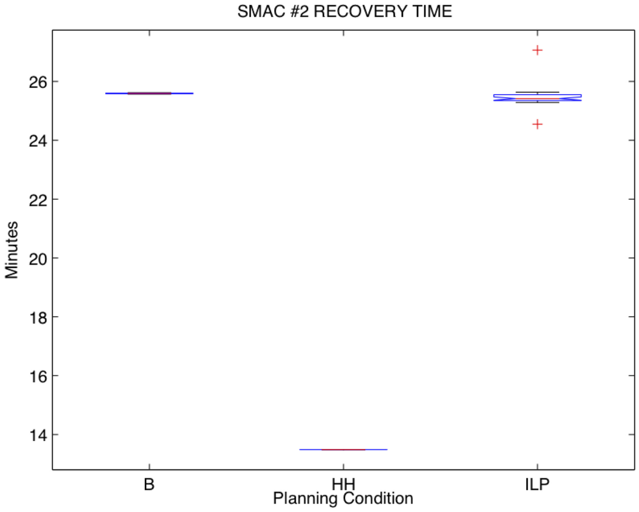

Fig. 4. Boxplot of emergency recovery time for SMAC #2. Red lines indicate the median value, center blue box indicates interquartile range, and whiskers extend to largest nonoutlier points.

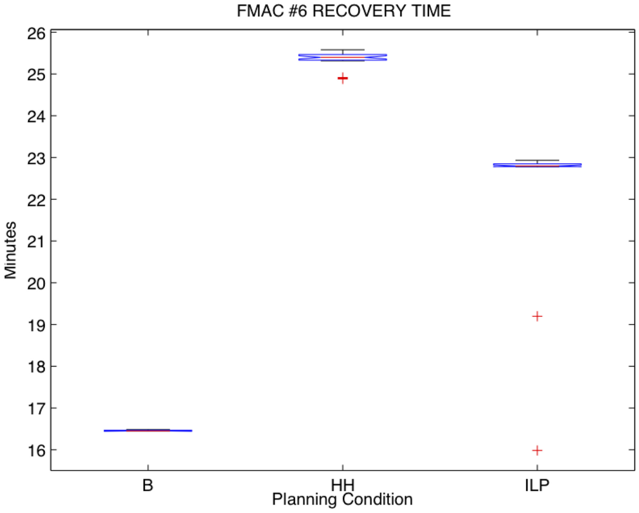

Fig. 5. Boxplot of emergency recovery time for FMAC #6. Red lines indicate the median value, enter blue box indicates interquartile range, and whiskers extend to largest nonoutlier points.

time (209.38 ± 3.03 min) than the HH planner (223.42 ± 2.16 min). Despite the rigid architecture of the landing patterns, the ILP planner was still better able to optimize the landing order so as to minimize total mission duration. The planner thus demonstrated the ability to optimize for the entire system at the cost of a wider safety margin for the failed aircraft, primarily SMAC #2. While the ILP may be better able to incorporate known future probabilities into its calculations, the conservative bias on the part of the HH planner also insulates the system against both known and unknown possible events. In this case, the ILP removed this conservative buffer to the benefit of the schedule; in turn, however, removing this buffer may reduce the trust operators place in the ILP planner.

C. Complex Scenario

The results from the complex scenario showed that both the HH and ILP planners were able to address the failures and the additional launches effectively, although quantitatively they performed quite differently with respect to the metrics of interest. This is primarily revealed through an analysis of the six metrics from Table III: the recovery time for the emergency aircraft (SMAC #2 RT), TAAT, TCAT, TATT, WTQMS, and the total MD. The remaining three metrics either showed no differences between planning conditions (FV, LZFT) or were not applicable (FMAC 6 RT: no FMAC experienced a fuel leak) and are not shown here.

TABLE VII
Results of Statistical Testing for the Complex Scenario

Metric	*B vs. HH*		*B vs. ILP*		*HH vs. ILP*	
	p-value	Superior	p-value	Superior	p-value	Superior
FV	-	-	-	-	-	-
SMAC 2 RT	**p=0.000**	HH	**p=0.000**	B	**p=0.000**	HH
FMAC 6 RT	-	-	-	-	-	-
MD	p=0.647	=	**p=0.000**	B	**p=0.000**	HH
TAAT	**p=0.000**	B	**p=0.000**	B	**p=0.000**	HH
TCAT	†**p=0.000**	HH	†**p=0.000**	B	†**p=0.000**	HH
TATT	†p=0.734	=	**p=0.001**	B	**p=0.000**	HH
WTQMS	**p=0.000**	B	**p=0.000**	B	**p=0.000**	HH

Bolded results signify significance at α = 0.002. All statistical tests nonparametric Mann–Whitney U tests, except where indicated by an †(where a parametric ANOVA was used)

Table VII presents a compilation of the statistical testing data for the remaining six metrics. This table again lists the metric name and details the statistical testing between pairs of planning conditions. The overall trend is that the performance of the ILP planner was significantly inferior to both the baseline and HH planners; in fact, at no time did the ILP planner even show equivalent performance to either of the other planning conditions in any metric. However, the difference in mission duration (Fig. 6) for the ILP planner (47.18 ± 0.84 min) was only about 1 min with regards to the B (45.91 ± 0.19 min) and HH (45.90 ± 0.23 min) planners.

Despite its inferior performance, the plans generated by the ILP were again qualitatively similar to the HH planner—both SUAV aircraft were sent to catapult 2 to launch, ensuring that they do not conflict with the incoming aircraft (the other accessible catapults, numbers 3 and 4, conflict with the landing zone). However, changes in the landing order were quite different. The HH planner only changed the landing order of a small number of aircraft: SMAC #2 was moved to the beginning of the landing order, bumping back all aircraft ahead of it by one slot. The ILP planner made landing order changes for almost every aircraft, but the emergency aircraft was only moved forward one position. Each change in landing order adds a slight switching cost to that aircraft, which, given the dynamics of the system, may be passed on to other aircraft. This switching cost was not purposefully designed into the system, and thus was unknown to the ILP planner. The accumulation of these switching costs negated the gain from advancing SMAC #2's landing position, resulting in a performance that was almost identical to not having made any change at all (the baseline case). However, the ILP planner still landed the emergency aircraft (SMAC #2) with sufficient fuel, caused no other aircraft to crash, and generated only one additional fuel violation. The main goals of the replanning were achieved, with only one additional minute of mission time required. Even though the planner may not have reached equivalent performance in these defined metrics, from a goal-oriented view, the ILP planner executed its task effectively.

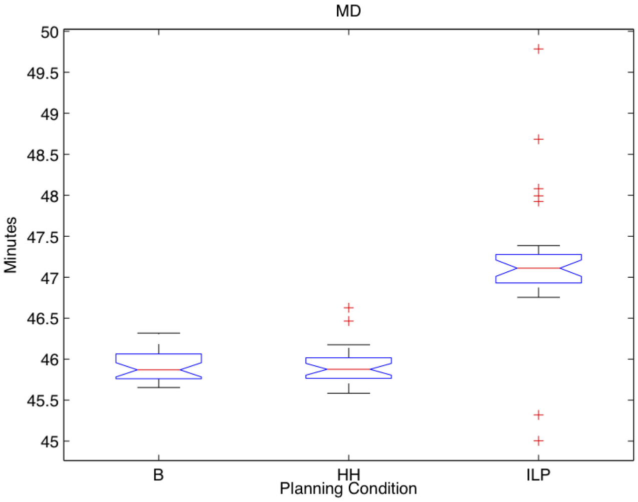

Fig. 6. Boxplot of mission duration values for the complex scenario. Red lines indicate the median value, center blue box indicates interquartile range, and whiskers extend to largest nonoutlier points.

From the opposite perspective, however, the HH plans were once again seen to be proficient in addressing the failures in the system, producing plans that not only accomplished all tasks but did so without seriously affecting the time of operation or incurring further penalties. In this case (as opposed to the moderate scenario), the HH planner still achieved a total mission duration slightly better than that of the ILP planner, while also minimizing the time required to land the failed SMAC 2 aircraft.

VII. Discussion

The main significance of the results presented in the previous section is that the performance of experienced human operators, who use a set of relatively simple, yet flexible, heuristics, is mostly as good and sometimes better than an integer linear program in a challenging planning environment. Statistically speaking, the HH planner outperformed in 11 out of the 17 measures across the three test scenarios. In the remaining six measures, the ILP planner showed better performance in three, with equivalent performance in the other three. However, performance was shown to be comparable in our primary measure of total mission duration. Each planner outperformed the other on one scenario, with no statistical difference in the third scenario. Even for the two cases showing differences in performance, the difference in mission duration was only a single minute, a minor difference for scenarios lasting more than half an hour. The majority of the other measures show similarly small, but statistically significant, differences. Thus, while statistical tests suggest some variations in performance, in terms of the actual values, the differences are not all that large.

Interestingly though, the planners differed qualitatively in their level of risk aversion. The HH planner tried to minimize the likelihood of potentially catastrophic accidents that, while highly unlikely, were still possible. For instance, in the complex scenario, the HH planner launched aircraft only from the forward catapults in order to ensure complete de-confliction with the landing strip; this ensures that, in the case of any true fuel emergencies, the landing strip remains available. In the moderate scenario, the HH planner acted to land the true emergency aircraft as soon as possible regardless of its current fuel state, minimizing the risk of losing the aircraft due to lack of fuel. The HH planner also acted to move the urgent emergency aircraft to the end of the landing order. Doing so minimizes the repercussions of any crash during landing, which would require a halt to landing operations for half an hour or more. This, in turn, increases the risk of losing other aircraft due to lack of fuel. The ILP planner does not contain this level of risk aversion; it still moves the true emergency aircraft forward in the order and the urgent aircraft backward, but not drastically so. This lack of risk aversion on behalf of the ILP planner seemingly led to a slight performance gain in the moderate scenario while having no significant detriments on safety in any of the scenarios, as no major increases in fuel violations, hydraulic violations, or other safety measures were observed. This suggests that there may be greater flexibility in scheduling operations than what is assumed by the subject matter experts.

However, a limiting factor in the current system is the ability of the ILP and HH planners to modify the behavior of actors (humans and vehicles) in the system. In other words, there is no provision to command vehicles to move faster, adjust the time between landings, or command complete reconfigurations of the deck (for instance, rearranging aircraft to free up a blocked catapult). This structural rigidity may explain why neither planner ever greatly outperformed the other; it may also explain why the lack of risk aversion in the ILP planner did not significantly decrease safety in the system. Enabling further modifications to actions and activities, which in turn requires increasing the complexity of the ILP algorithm, may provide boosts to system performance. It may also lead to a decrease in safety, which may require implementing additional measures to ensure that activities maintain the desired level of safety.

VIII. Conclusion

In this paper, we formally evaluate the performance of alternative task planning approaches within a simulation environment that replicates operations on aircraft carrier decks involving real-time resource allocation with timing and sequence constraints. Our results indicate both the power of simple human planning heuristics in complex environments, as well as the potential for an automated optimization-based planning algorithm to support coordinated human–vehicle operations, either by offloading cognitive workload to the planner or by providing alternative plans with varying benefits.

In the future, considering that optimization algorithms encounter difficulties in real-time adaptation to unknown scenarios and human heuristics are designed to work in uncertain situations by accumulating several years of experience, incorporating these heuristics as constraints or alternate objective functions might benefit the ILP. We would also like to take advantage of the fundamental difference in the nature of the plans generated by the human heuristics and the ILP, and have the flexibility of switching between safe and risky behavior based on the operating conditions. This would involve learning which heuristics to turn on or activate based on prior knowledge of what works better. We additionally plan to conduct user studies to identify which type of aircraft and tasks are prioritized depending upon the timing and safety constraints. This would enable us to model the soft constraints either as hard constraints or ignore them altogether, thereby, eliminating the need for obtaining a suitable value for the violation margin and enhancing the usability of the system in terms of ready acceptance of the generated schedules.

The simulation environment embedded in the DCAP system and the methods by which the planners are allowed to schedule operations are based on the current observations of the actual aircraft carrier environment. However, in the future, additional levels of autonomy in aircraft or supervisory control systems may allow structural modifications to operations, which may also provide performance benefits as noted earlier. One such allowable modification is flexible or on-demand availability of crew members to perform tasks on the deck. This allowance will significantly increase the dimensionality of the planning problem for the ILP, thereby, rendering it computationally intractable to support real-time operations using any standard solver. We have already developed a novel regression-based approach [34] to address the issue by exploiting the similarity in the structure (in terms of the decision variables and the constraints) of the relaxed LP problems that are solved at the nodes of the branch and bound trees of ILP problems. We plan to make use of this regression approach in futuristic DCAP environments with enhanced autonomy levels to solve large-scale planning problems efficiently with performance guarantees on the estimated solution values.

Acknowledgment

This work was performed as a part of a larger research effort by the members of the Aerospace Controls Laboratory, Computer Science and Artificial Intelligence Laboratory, the Humans and Automations Laboratory, and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. The authors would like to thank the numerous students and professors who contributed in the development of the system. They would also like to thank the reviewers for their helpful comments in the editing of this paper.

References

[1] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problem with time window constraints," *Operations Res.*, vol. 35, no. 2, pp. 254–265, Mar.–Apr. 1987.

[2] R. H. Möhring, A. S. Schulz, F. Stork, and M. Uetz, "On project scheduling with irregular starting time costs," *Operations Res. Lett.*, vol. 28, no. 4, pp. 149–154, 2001.

[3] C.-F. Liaw, C. C. White, and J. Bander, "A decision support system for the bimodal dial-a-ride problem," *IEEE Trans. Syst., Man, Cybern., Part A, Syst., Humans*, vol. 26, no. 5, pp. 552–565, Sep. 1996.

[4] I. Benyahia and J.-Y. Potvin, "Decision support for vehicle dispatching using genetic programming," *IEEE Trans. Syst., Man, Cybern., Part A, Syst., Humans*, vol. 28, no. 3, pp. 306–314, May 1998.

[5] J. Malasky, L. M. Forest, A. C. Khan, and J. R. Key, "Experimental evaluation of human–machine collaborative algorithms in planning for multiple UAVs," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 3, pp. 2469–2475, 2005.

[6] T. Shima and C. Schumacher, "Assignment of cooperating UAVs to simultaneous tasks using genetic algorithms," in *Proc. AIAA Guidance, Navigation Control Conf. Exhibit*, San Francisco, CA, USA, 2005.

[7] F. Xue, A. C. Sanderson, and R. J. Graves, "Multiobjective evolutionary decision support for design-supplier-manufacturing planning," *IEEE Trans. Syst., Man, Cybern., Part A, Syst., Humans*, vol. 39, no. 2, pp. 309–320, Feb. 2009.

[8] H. Bui, X. Han, S. Mandal, K. R. Pattipati, and D. L. Kleinman, "Optimization-based decision support algorithms for a team-in-the-loop planning experiment," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2009, pp. 4684–4689.

[9] M. Gudadhe, K. Wankhade, and S. Dongre, "Decision support system for heart disease based on support vector machine and artificial neural networks," in *Proc. Int. Conf. Comput. Commun. Technol.*, Sep. 2010, pp. 741–745.

[10] Y. Jin, Y. Liao, and M. M. Polycarpou, "Balancing search and target response in cooperative unmanned aerial vehicle (UAV) teams," *IEEE Trans. Syst., Man, Cybern., Part A, Syst., Humans*, vol. 36, no. 3, pp. 571–587, Jun. 2006.

[11] M. Cummings and P. Mitchell, "Automated scheduling decision support for supervisory control of multiple UAVs," *AIAA J. Aerospace Comput., Inform., Commun.*, vol. 3, no. 6, pp. 294–308, 2006.

[12] V. Giordano, P. Ballal, F. Lewis, B. Turchiano, and J. B. Zhang, "Supervisory control of mobile sensor networks: Math formulation, simulation, and implementation," *IEEE Trans. Syst., Man, Cybern., Part B, Cybern.*, vol. 36, no. 4, pp. 806–819, Aug. 2006.

[13] C. Galindo, J.-A. Fernandez-Madrigal, and J. Gonzalez, "Multihierarchical interactive task planning: Application to mobile robotics," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 38, no. 3, pp. 785–798, Jun. 2008.

[14] O. Arslan and G. Inalhan, "An event driven decision support algorithm for command and control of UAV fleets," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 5198–5203 .

[15] M. L. Cummings and A. S. Brzezinski, "Global versus local decision support for multiple independent UAV schedule management," *Int. J. Appl. Decision Sci.*, vol. 3, no. 3, pp. 188–205, 2010.

[16] J. Scanlon. (2009). How KIVA Robots Help Zappos and Walgreens [Online]. Available: http://www.businessweek.com/innovate/content/apr2009/id20090415$_$8764%20.htm

[17] G. I. Rochlin, T. R. La Porte, and K. H. Roberts, "The self-designing high-reliability organization: Aircraft carrier flight operations at sea," *Naval War College Rev.*, vol. 40, no. 4, pp. 76–90, 1987.

[18] K. H. Roberts, "Some characteristics of one type of high-reliability organization," *Organization Sci.*, vol. 1, no. 2, pp. 160–176, 1990.

[19] "Autonomous Vehicles in Support of Naval Operations," Naval Studies Board, Washington, DC, USA, Tech. Rep., 2005.

[20] S. Ackerman. (2011). Navy Wants Its Drone to Land on a Carrier with a Mouse Click. Wired.com Danger Room [Online]. Available: http://www.wired.com/dangerroom/2011/04/navy-wants-mouse-click-flying-f%or-its-carrier-based-drone/

[21] H. Balakrishnan and Y. Jung, "A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport," in *Proc. AIAA GNC*, Hilton Head, SC, USA, 2007.

[22] P. C. Roling and H. G. Visser, "Optimal airport surface traffic planning using mixed-integer linear programming," *Int. J. Aerospace Eng.*, vol. 2008, pp. 1:1–1:11, Jan. 2008.

[23] G. Gupta, W. Malik, and Y. C. Jung, "A mixed integer linear program for airport departure scheduling," in *Proc. 9th AIAA ATIO Conf.*, Hilton Head, SC, USA, 2009.

[24] A. M. Churchill and D. J. Lovell, "Coordinated aviation network resource allocation under uncertainty," *Transportation Res. Part E*, vol. 48, no. 1, pp. 19–33, Jan. 2012.

[25] C. N. Glover and M. O. Ball, "Stochastic optimization models for ground delay program planning with equity–efficiency tradeoffs," *Transportation Res. Part C*, pp. 196–202, 2013.

[26] J. M. Tappan, D. J. Pitman, M. L. Cummings, and D. Miglianico, "Display requirements for an interactive rail scheduling display," in *Engineering Psychology and Cognitive Ergonomics*, D. Harris, Ed., vol. 6781 of Lecture Notes in Computer Science, Berlin/Heidelberg, Germany: Springer, 2011, pp. 352–361.

[27] J. C. Ryan, M. L. Cummings, N. Roy, A. Banerjee, and A. S. Schulte, "Designing an interactive local and global decision support system for aircraft carrier deck scheduling," in *Proc. AIAA Infotech@Aerospace Conf.*, St. Louis, MO, USA, 2011.

[28] G. Gigerenzer and P. M. Todd, *Simple Heuristics That Make Us Smart*, S. Stich, Ed. New York, NY, USA: Oxford Univ. Press, 1999.

[29] H. A. Simon, G. B. Dantzig, R. Hogarth, C. R. Plott, H. Raiffa, T. C. Schelling, K. A. Shepsle, R. Thaler, A. Tversky, and S. Winter, "Decision making and problem solving," *Interfaces*, vol. 17, no. 5, pp. 11–31, 1987.

[30] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," in *Judgment Under Uncertainty: Heuristics and Biases*, D. Kahneman, P. Slovic, and A. Tversky, Eds. Cambridge, U.K.: Cambridge Univ. Press, 1982, ch. 1.

[31] J. P. Spradley, *The Ethnographic Interview*. New York, NY, USA: Holt, Reinhart, and Winston, 1979.

[32] A. S. Huang, E. Olson, and D. Moore, "LCM: Lightweight communications and marshalling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 4057–4062.

[33] R. Lougee-Heimer, "The Common Optimization INterface for Operations Research," *IBM J. Res. Develop.*, vol. 47, no. 1, pp. 149–154, 2003.

[34] A. G. Banerjee, M. Ono, N. Roy, and B. Williams, "Regression-based LP solver for chance constrained finite horizon optimal control with nonconvex constraints," in *Proc. Amer. Control Conf.*, pp. 131–138, 2011.

Jason C. Ryan (M'12) received the Bachelor of Science degree in aerospace engineering from the University of Alabama, Tuscaloosa, AL, USA, in 2007, and the Master of Science degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2011. He is currently pursuing the Ph.D. degree at the Engineering Systems Division, MIT. His dissertation work focuses on building agent-based models of human-unmanned vehicles systems, with an emphasis on examining the tradeoffs between safety and mission effectiveness for different implementations of unmanned vehicles.

His current research interests include human supervisory control, human-unmanned vehicle interaction, and collaborative human–computer decision making.

Ashis Gopal Banerjee (S'08–M'09) received the B.Tech. degree in manufacturing science and engineering from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 2004, the M.S. degree in mechanical engineering from the University of Maryland (UMD), College Park, MD, USA, in 2006, and the Ph.D. degree in mechanical engineering from UMD in 2009.

He is currently with the GE Global Research Center, Niskayuna, NY, USA. Previously, he was a Research Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. His current research interests include multi-robot planning and control under uncertainty, sequential decision making, and cyber-physical systems.

Mary L. Cummings (SM'03) received the B.S. degree in mathematics from the United States Naval Academy, Annapolis, MD, USA, in 1988, the M.S. degree in space systems engineering from the Naval Postgraduate School, Monterey, CA, USA, in 1994, and the Ph.D. degree in systems engineering from the University of Virginia, Charlottesville, VA, USA, in 2004.

Being a naval officer and military pilot from 1988 to 1999, she was one of the Navy's first female fighter pilots. She is currently an Associate Professor with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. Her current research interests include human supervisory control, human-unmanned vehicle interaction, collaborative human–computer decision making, decision support, human performance modeling and the ethical, and social impact of technology.

Nicholas Roy (M'96) received the Ph.D. degree in robotics from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

He is currently an Associate Professor with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, and a member of the Computer Science and Artificial Intelligence Laboratory, MIT. His current research interests include autonomous robots, decision-making under uncertainty, machine learning, and human–computer interaction.