2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Complex Mission Optimization for Multiple-UAVs using Linear Temporal Logic

Sertac Karaman Emilio Frazzoli

***Abstract*— This paper discusses a class of mission planning problems in which mission objectives and relative timing constraints are specified using the Linear Temporal Logic language LTL_{-X}. Among all mission plans that satisfy the LTL_{-X} specifications, it is desired to find those minimizing a given cost functional. We show that such an optimization problem can be formulated as a Mixed-Integer Linear Program, and present an algorithm for this purpose. This algorithm mainly relies on a novel systematic procedure which converts a given LTL_{-X} formula into a set of mixed-integer linear constraints. The approach presented here can be used for Multiple-UAV Mission Planning purposes, allowing the operator to specify complex mission objectives in LTL_{-X} in a very natural manner; the proposed algorithm constructs the optimal mission plan satisfying the given LTL_{-X} specification. Examples for practical problem sizes are presented and discussed in the paper.**

I. Introduction

Uninhabited Air Vehicles (UAVs) are evolving to have more computational power and better sensors. These improvements have been increasing their potential capability for executing more complex tasks with reduced human supervision. In order to realize such potential, there has been a need for algorithms for high-level coordination of multiple UAVs [1], [2]. Especially the importance of relative timing constraints—such as the requirement of servicing sets of targets in a given order—was emphasized in [1]. Also, complex scenarios with heterogeneous targets and UAVs often result in combinatorial mission specifications, in which not all objectives must be accomplished, and not all vehicles must be used; for example, an objective might be to cover a subset of the targets, assuming that some other criteria are satisfied, e.g. another partial task has been completed. Then the mission planning algorithm must be able to decide which targets to service in the end using perhaps a carefully selected subset of the UAVs in order not to risk all of them. An important problem is then the specification of these type of constraints in a natural manner using a high-level language. Given the specification, solution methods are also required that will plan the mission in an optimal way considering the given constraints.

In this paper, we propose a solution strategy that relies on the Linear Temporal Logic language LTL_{-X} for specification of the aforementioned constraints. We then present a Mixed Integer Linear Programming (MILP) based algorithm that yields the exact solution for a broad subset of problems. Essentially, we employ a slightly generalized version of VRP with Time Windows (VRPTW), to which we add the Linear Temporal Logic specifications, encoded as a set of linear constraints in integer variables. We also present examples of practical sizes with complex mission objectives and several temporal constraints in the paper. We should note that the VRPTW and several other variants of VRPs have already been applied to UAV scheduling problems [3]–[5]; however, Temporal Logic based specifications have never been considered on top of the VRP formulations, to the authors' knowledge.

Temporal Logic is a form of modal logic first proposed by philosophers and then was used for several applications in different disciplines [6]. The seminal work of Pnueli [7] first discussed its applications in Computer Science for temporal reasoning about computer programs [8]. Using temporal logic as a specification language, one can employ well-developed model-checking techniques [9] in order to decide whether a given program satisfies a set of temporal specifications. Temporal logic has also been used for planning in Artificial Intelligence [10] and in Control Theory as early as eighties [11]. Some of the most recent applications in control theory include [12], [13]. It should be noted that simple propositional logic has already been considered in an integer programming framework in [14], [15] and employed in control theory together with integer programming to represent a broad class of systems in a unifying way in [16]. But none of those references consider temporal logic.

The contributions of this paper are as follows. First, under a finite horizon assumption, we present a novel systematic procedure which constructs a set of linear constraints for any given LTL_{-X} formula, such that this formula is satisfied by a set of atomic propositions if and only if a corresponding set of binary variables satisfy the constructed constraints. Second, we introduce the problem of mission planning with linear temporal logic specifications. We argue with examples that it is quite natural to model the mission objectives and several complicated constraints using LTL_{-X} language. Finally, we present an exact algorithm to solve this problem under some technical assumptions. We show in the examples that several practical cases already satisfy the given assumptions. It should also be noted that although the presentation in the paper focuses on UAV mission planning, the framework can be employed for several other logistics problems modeled using a variant of VRP.

The rest of the paper is organized as follows. We provide some preliminary definitions and present the LTL_{-X} language in Section II. Section III introduces the formulation of a given LTL_{-X} formula as linear constraints in mixed integer variables. The optimal mission planning problem

S. Karaman and E Frazzoli are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA {sertac,frazzoli}@mit.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

under temporal constraints is defined in Section IV. A brief discussion on the solution approach of this paper is also included in the same section. Section V is devoted to introduction of the MILP based exact algorithm which solves the problem for a class of instances. This section also presents simulation results for an example mission. The paper ends with conclusions and ideas for future work.

II. Preliminary Definitions and LTL$_{-X}$ Language

A. Preliminary Definitions

We will use logics to reason about temporal properties of a *system* which is composed of a set of *system variables*. Let us denote the set of all system variables with X, which takes values in a domain $\mathcal{X}$. Let Π be a set of atomic propositions, i.e., abstract statements on the system variables X. Every proposition takes a value either `True` (1 or $\top$) or `False` (0 or $\perp$) at any given time instant.

A *state* s is defined as a mapping which assigns certain values to all of the system variables as well as the atomic propositions. If a state s_i assigns a proposition p value `True` then it is denoted as $s_i \Vdash p$. If the value assigned to proposition p is `False` then we denote it by $s_i \nVdash p$.

Definition A *transition system* is a tuple $\mathcal{TS} = (Q, Q_0, \rightarrow, \Pi, \vDash)$ where Q is a set of states, $Q_0 \subseteq Q$ is a set of initial states, $\rightarrow \subseteq Q \times Q$ is a transition relation, Π is a set of atomic propositions, and $\vDash \subseteq Q \times \Pi$ is a satisfaction relation.

Let $\mathcal{T} = \{1, 2, \ldots\}$ be a set of time instances and $\sigma = (s^1, s^2, \ldots)$ be an infinite sequence of states. (Superscripts will used to indicate time instances throughout the paper.)

Definition A *run* σ on a given transition system $\mathcal{TS}$ is an infinite set of sequences $\sigma = (s^1, s^2, \ldots)$ such that $s^i \in Q$ for all $i \in \mathcal{T}$, $s^1 \in Q_0$, and $(s^i, s^{i+1}) \in \rightarrow$ for all $i \in \mathcal{T}$, where $\mathcal{T} = \{1, 2, \ldots\}$ is the set of time instances.

As a run is being executed on a transition system $\mathcal{TS}$, all the atomic propositions in Π evolve with time. The time evolution of atomic propositions is defined as follows.

Definition The *time evolution of an atomic proposition* $p \in \Pi$ in a given run $\sigma = (s^1, s^2, \ldots)$ on an extended transition system $\mathcal{TS}$ is defined as the infinite sequence $\pi = (p^1, p^2, \ldots)$ which a takes value p^i for each state s^i for $\forall i \in \mathcal{T}$.

Informally speaking, a temporal specification is a constraint on the time evolution of the atomic propositions of a transition system.

Definition A *temporal specification on* Π states that a formula ϕ constructed using the atomic propositions $p \in \Pi$ of a given extended transition system $\mathcal{TS}$ must be `True` at the initial time. This formula ϕ satisfies the syntax of a temporal language.

The underlying time structure in a mission planning problem or a VRP is continuous in the sense that the events take place at some point t in time where t is a continuous variable. In order to address this situation we define the satisfaction functions and continuous valuations of propositions.

Definition A *satisfaction function* $S : [0, \infty) \rightarrow 2^{\Pi}$ defined on the set of propositions Π gives the set $S(t) \in \Pi$ of propositions that are `True` at time t.

S can be considered as a run in a continuous time setting. We define $S[t]$ as the same function but with domain $[t, \infty)$, i.e. $S[t] : [t, \infty) \rightarrow 2^{\Pi}$, for which $S[t](\tau) = S(\tau)$ for $\forall \tau \in [t, \infty)$.

The continuous valuation of a proposition is a function which assigns a value in $\{0, 1\}$ to a proposition for every given time point $t \in [0, \infty)$. A formal definition is as follows,

Definition The *continuous valuation of a proposition* is a function $\pi_C : [0, \infty) \rightarrow \{0, 1\}$ for which $\pi_C(t) = 1$ if the proposition it addresses is `True` at time t, and 0 otherwise.

B. Syntax of LTL$_{-X}$

The syntax of the LTL$_{-X}$ language can be defined recursively as follows. Every atomic proposition $p \in \Pi$ is an LTL$_{-X}$ formula, and if ϕ and ψ are formulas then so are $\neg\phi$, $\phi \wedge \psi$, and $\phi \mathcal{U} \psi$. The grammar of LTL$_{-X}$ is then

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \phi \mathcal{U} \phi$$

where ϕ is a formula, $\neg$ is the *negation* operator, $\vee$ is the *disjunction* operator, and $\mathcal{U}$ is the *until* operator. Informally speaking, the temporal operator *until* when used as $p\mathcal{U}q$ implies that p will keep being `True` until q becomes `True`.

It is convenient to define some additional operators other than the ones that are used for building up the grammar. Given the operators *negation* and *disjunction*, the operators *conjunction* ($\wedge$), *implication* ($\Rightarrow$), and *equivalency* ($\Leftrightarrow$) can be defined, respectively, as $\phi_1 \wedge \phi_2 = \neg(\neg\phi_1 \vee \neg\phi_2)$, $\phi_1 \Rightarrow \phi_2 = \neg\phi_1 \vee \phi_2$, and $\phi_1 \Leftrightarrow \phi_2 = (\phi_1 \Rightarrow \phi_2) \wedge (\phi_1 \Rightarrow \phi_2)$ respectively. Finally, using all these operators with *until*, we can define *eventually* ($\Diamond$) and *always* ($\Box$) as $\Diamond\phi = \top\mathcal{U}\phi$ and $\Box\phi = \neg\Diamond\neg\phi$. We also define the operator *unless* as $p\mathcal{W}q = (\Box p) \vee (p\mathcal{U}q)$. *Unless* is almost the same as *until* but it is slightly weaker in the sense that q does not have to be `True` eventually. This operator is also known as *weak until*. All these new operators do not add any expressive power to the language, they are just used to make the notation easier.

C. LTL$_{-X}$ Discrete Semantics

Given an extended transition system $\mathcal{TS}$, a run σ on $\mathcal{TS}$ is said to satisfy the atomic proposition p at an instance j (denoted by $(\sigma, j) \vDash p$) if and only if $(s_j, p) \in \vDash$ (denoted by $s_j \Vdash p$). We write $s_j \nVdash p$ if and only if $(s_j, p) \notin \vDash$.

Let p be an atomic proposition, and ϕ and ψ be any two formulas in LTL$_{-X}$, then the discrete semantics of LTL$_{-X}$ are defined as

$$(\sigma, j) \vDash p \quad \text{iff} \quad s_j \Vdash p;$$

$$(\sigma, j) \vDash \neg\phi \quad \text{iff} \quad (\sigma, j) \nvDash \phi; \tag{1}$$

$$(\sigma, j) \vDash \phi \vee \psi \quad \text{iff} \quad (\sigma, j) \vDash \phi \text{ or } (\sigma, j) \vDash \psi; \tag{2}$$

$$(\sigma, j) \vDash \phi\mathcal{U}\psi \quad \text{iff} \quad \exists k \geq j \text{ such that } (\sigma, k) \vDash \psi, \text{ and for } \forall i, j \leq i < k : (\sigma, i) \vDash \phi. \tag{3}$$

Even though the semantics given above are complete we also define other operators as well for later use,

$$(\sigma, j) \vDash \phi \wedge \psi \quad \text{iff} \quad (\sigma, j) \vDash \phi \text{ and } (\sigma, j) \vDash \psi; \tag{4}$$

$$(\sigma, j) \vDash \Box\phi \quad \text{iff} \quad (\sigma, k) \vDash \phi \text{ for } \forall k \geq j; \tag{5}$$

$$(\sigma, j) \vDash \Diamond\phi \quad \text{iff} \quad \exists k \geq j \text{ such that } (\sigma, k) \vDash \phi. \tag{6}$$

III. MILP Formulation of Operators

In this section we describe a novel systematic procedure to obtain a set of linear inequalities in the time evolution of atomic propositions used in the formula ϕ. These inequalities are satisfied if and only if the time evolution of the atomic propositions satisfy the given temporal formula ϕ for the discrete semantics of LTL$_{-X}$. The procedure relies on the following assumption.

Assumption 1.a *All the propositions have continuous valuation functions that make only a finite number of transitions from 0 to 1 within the time interval* $[0,\infty)$.

This assumption means that there are no events in the system that occur infinitely many times. So the system in the end reaches to a final state and stays there forever. Such an example in a mission planning setting is a case in which some of the targets have been serviced and all the vehicles have landed. This assumption is quite practical and this type of mission planning problems have been widely studied in the literature using MILP approaches [1], [3]–[5].

To obtain the linear inequalities, we define a slack variable for each occurrence of an operator in the formula ϕ. Then we form the inequalities for this slack variable and substitute this variable instead of the operator in the formula ϕ. This substitution process is continued until there is no operator left in the formula. At the end, the formula ϕ becomes just one slack variable and we will have several inequalities. Finally the last constraint will be to force the last slack variable to be equal to 1, i.e., to be `True` for the first time instance, which is defined as the present time in our case.

Relying on Assumption 1.a, let the time indices be denoted as a finite set $\Xi = \{1,\ldots,T\}$. In order to relate Assumption 1.a with the definitions of Section II, we assume that the last state repeats itself for all times after T for any given run.

The negation as defined by (1) can be represented as

$$\delta^t = (1-p^t) \quad t=1,\ldots,T. \tag{7}$$

where δ is the slack variable that will be substituted in the formula ϕ instead of $\neg p$. Note that if the value of p is `True` at any time instance then the value of δ is `False` and vice versa, which is exactly what we desire. The continuous variable δ should satisfy $0 \le \delta^t \le 1, \delta^t \in \mathbb{R}$. These constraints hold for all the slack variables defined in this section.

Conjunctive constraints were defined in (4). Now let us consider a more general case where conjunction of several propositions each denoted by p_i, i.e., $\bigwedge_{i=1}^k p_i$. For this general case, only one slack variable is defined, which satisfies the following constraints:

$$\delta^t \le p_i^t \qquad i=1,\ldots,k, \quad t=1,\ldots,T; \tag{8}$$
$$\delta^t \ge \textstyle\sum_{i=1}^k p_i^t - (k-1) \quad t=1,\ldots,T. \tag{9}$$

Disjunctive constraints defined by (2) can also be generalized in a similar manner. The constraints for $\delta = \bigvee_{i=1}^k p_i$ are

$$\delta^t \le \textstyle\sum_{i=1}^k p_i^t \quad t=1,\ldots,T; \tag{10}$$
$$\delta^t \ge p_i^t \qquad i=1,\ldots,k, \quad t=1,\ldots,T. \tag{11}$$

In the case of the *eventually* operator defined by (6), the slack variable must satisfy the following constraints

$$\delta^t \le \textstyle\sum_{\tau=t}^T p^\tau \quad t=1,\ldots,T; \tag{12}$$
$$\delta^t \ge p^\tau \qquad \tau=t,\ldots,T, \quad t=1,\ldots,T. \tag{13}$$

The slack variable used to represent the *always* operator as in (5) must satisfy the following constraints

$$\delta^t \le p^\tau \qquad \tau=t,\ldots,T,\ t=1,\ldots,T; \tag{14}$$
$$\delta^t \ge \textstyle\sum_{\tau=t}^T p^\tau - (T-t) \quad t=1,\ldots,T. \tag{15}$$

For the *until* operator, one has to define more variables in order to make the constraints linear. Namely, the slack variable δ must satisfy

$$\alpha_{tj} \ge q^j + \textstyle\sum_{\tau=t}^j p^\tau - (j-t+2) \quad j=t+1,\ldots,T,\ t=1,\ldots,T-1; \tag{16}$$
$$\alpha_{tj} \le q^j \qquad j=t+1,\ldots,T,\ t=1,\ldots,T-1; \tag{17}$$
$$\alpha_{tj} \le p^\tau \qquad \tau=t,\ldots,j, \quad j=t+1,\ldots,T,\ t=1,\ldots,T-1; \tag{18}$$
$$\alpha_{tt} = q^t \qquad t=1,\ldots,T; \tag{19}$$
$$\delta^t \le \textstyle\sum_{j=t}^T \alpha_{tj} \quad t=1,\ldots,T; \tag{20}$$
$$\delta^t \ge \alpha_{tj} \qquad t=1,\ldots,T, \quad j=t,\ldots,T. \tag{21}$$

where the extra variables are denoted by $\alpha_{tj} \in \mathbb{R}$ with $0 \le \alpha_{tj} \le 1$. These extra variables have been utilized in order to make the constraints linear in the time evolution of the propositions. Note that this formulation of *until* is slightly conservative since it requires p and q to be `True` together during the transition from p to q if it occurs. This condition is introduced in order to make the *until* constraint hold for the continuous-time valuation of the two propositions when it holds for the discrete-time evolution of the two propositions. This conservatism is arbitrary small since it only makes p to be `True` for only an extra amount of time of zero measure. (16) can be modified to $\alpha_{tj} \ge q^j + \sum_{\tau=t}^j p^\tau - (j-t+1)$ in order to satisfy the same discrete semantics given by (3).

IV. Mission Planning under Temporal Constraints

A. LTL$_{-X}$ Continuous Semantics

Given a satisfaction function S, the LTL$_{-X}$ continuous semantics are defined as follows,

$$\begin{aligned}
S[t] &\vDash_C p && \text{iff} \quad p \in S(t);\\
S[t] &\vDash_C \neg\phi && \text{iff} \quad S[t] \nvDash_C \phi;\\
S[t] &\vDash_C \phi \vee \psi && \text{iff} \quad S[t] \vDash_C \phi \text{ or } S[t] \vDash_C \psi;\\
S[t] &\vDash_C \phi\,\mathcal{U}\psi && \text{iff} \quad \exists\tau \ge t \text{ such that } S[\tau] \vDash_C q \text{ and}\\
& && \qquad \text{for } \forall\bar\tau \text{ with } t \le \bar\tau \le \tau, S[\bar\tau] \vDash_C \phi.
\end{aligned}$$

where ϕ and ψ are formulas in LTL$_{-X}$, p is an atomic proposition and $S[t] \nvDash_C \phi$ refers to $S[t] \vDash_C \phi$ not being `True`. We denote the set of all satisfaction functions for which a given LTL$_{-X}$ formula ϕ holds with $\mathcal{S}(\phi)$, i.e., $\mathcal{S}(\phi) = \{S : S[0] \vDash_C \phi\}$.

B. Problem Formulation

Given a set of targets and a set of UAVs, the Multiple-UAV Mission Planning problem is to assign the targets to UAVs in an optimal manner, and generate the optimal paths to be followed by each UAV. Generally the main mission objective is to service all the targets employing all the UAVs. We further generalize this problem such that the mission objective is also an LTL_{-X} specification, and not all the UAVs have to be used nor all the targets have to be serviced in the resulting mission plan.

To give some examples, let p_i be the proposition which will be true if target i has been serviced. In this case a mission objective can be to eventually service target 1 or target 2. This objective can be denoted in LTL_{-X} as $\Diamond(p_1 \vee p_2)$. Then the resulting plan can either service target 1 or target 2 but not necessarily both. We can also state that target 2 will not be serviced until target 1 is serviced which can be denoted as$((\neg p_2)\,\mathcal{U} p_1)$. This specification merely means that if target 2 is serviced, it will be serviced after target 1. If targets 1, 2, and 3 need to be visited in the given order, then the following specification can be used: $(\Diamond p_3) \wedge ((\neg p_3)\,\mathcal{U} p_2) \wedge ((\neg p_2)\,\mathcal{U} p_1)$. In order to demonstrate a further example assume that q_{ik} is true whenever UAV k services target i. Then to avoid UAV 1 servicing target 1, one can state that always UAV 1 will not service target 1 which can be denoted as $\Box \neg q_{11}$. Note that negations, conjunctions and disjunctions of these examples can be used for specifying even more complex mission objectives.

The cost function of the multiple-UAV Mission Planing problem is a linear function $f : \mathcal{X} \to \mathbb{R}$ from the set of values of system variables to the real numbers.

Problem Given the set of system variables X for a multiple-UAV mission planning problem instance, the set propositions Π defined on X, a temporal specification ϕ on Π, and a cost function $f(x)$; *Multiple-UAV mission planing problem with temporal constraints* is to find a value $x \in \mathcal{X}$ for the system variables such that $S_x \in \mathcal{S}(\phi)$ holds and $f(x)$ is minimized*. In a more formal way the problem is to

$$\text{minimize} \quad f(x), \tag{22}$$

$$\text{subject to} \quad x \in \mathcal{X}, \tag{23}$$

$$S_x \in \mathcal{S}(\phi). \tag{24}$$

C. Solution Methodology

In order the solve this problem, we utilize the method presented in the previous section for MILP formulation of LTL_{-X} specifications. We first model the underlying mission planning problem using MILP formulations of VRPs. We relax these formulations to some extent, slightly generalizing the problem so that the conditions for which not all the UAVs have to be used, or not every target has to be serviced, can be modeled. Even though the underlying time is continuous in the problem, in order to utilize the methodology presented in the previous section, we define a finite set of time instances $\Xi = \{1, \dots, T\}$. Each of these time instances occur at some point in time denoted by $\theta^t \in \mathbb{R}^+$ defined for $\forall t \in \Xi$. Given a set of propositions Π, we define binary variables P^t for each proposition in $p \in \Pi$. For several practical propositions, we present linear inequalities such that P^t is 1 if p is `True` at time t. We finally use the formulation in the previous section to constrain the variables P^t such that the given LTL_{-X} specification is satisfied in the discrete level. Solving the resulting MILP with several constraints yields the optimal mission plan that satisfies the given temporal constraints.

*Note that the satisfaction function changes with the values of system variables. We denote the satisfaction function corresponding to a value $x \in \mathcal{X}$ by S_x

In this framework one has to show that if the propositions satisfy a given specification at the discrete level, then the specification is also satisfied by the continuous valuations. Although being slightly conservative, using the following the assumptions on the atomic propositions and the specification, it is easy to show that if the formula is satisfied in the discrete level, i.e., by time evaluations of the propositions, then the continuous valuations also satisfy the formula.

The first assumption replaces Assumption 1.a as follows.

Assumption 1.b *All propositions are initially* `False`*, and they make only one transition to* `True` *or stay* `False` *forever.*

This assumption seems very constraining, but in fact, as will be shown in the examples several practical examples fall into this category, e.g., a UAV being launched or landed, a target being classified by a UAV, a communication link being established etc. The second assumption is the following.

Assumption 2 *In the specification, temporal operators are only applied to the atomic propositions or their negations.*

With this assumption, everything becomes a combination of conjunctions, disjunctions and negations of several temporal specifications that are only applied to atomic propositions and their negations. It will be seen in examples that several very practical cases can be modeled under these assumptions. Note that the following equivalencies can also be used: $[(\Diamond\phi) \vee (\Diamond\psi)] \Leftrightarrow [\Diamond(\phi \vee \psi)]$, $[(\Box\phi) \wedge (\Box\psi)] \Leftrightarrow [\Box(\phi \wedge \psi)]$, $[(\phi\,\mathcal{U}\psi) \vee (\phi\,\mathcal{U}\varphi)] \Leftrightarrow [\phi\,\mathcal{U}(\psi \vee \varphi)]$, and $[(\phi\,\mathcal{U}r) \wedge (\psi\,\mathcal{U}\varphi)] \Leftrightarrow [(\phi \wedge \psi)\,\mathcal{U}\psi]$, where ϕ, ψ and φ are any LTL_{-X} formula. The left sides of the equivalences can be substituted for the right sides in the formula in order to satisfy Assumption 2.

Assumption 2 can be relaxed by allowing only one proposition to have a transition in its value between every two consequent time instances. This can be realized by defining more continuous variables and constraints. However it requires the number of instances to be at least as many as the number of atomic propositions. This indeed causes a considerable growth in the computation time. The method is as follows. Let $\mathcal{P} = \{i : p_i \in \Pi\}$ denote the set of indices of the propositions. For every proposition $p_i \in \Pi$ and each time instance $t \in \Xi$ let us define a continuous variable $\beta_i^t \in \mathbb{R}$ with $0 \le \beta_i^t \le 1$. This variable will be constrained to be 1 if p_i made a transition from `False` to `True` between time instances t and $t+1$, and 0 if no such transition occurs. The following constraints ensure this property: $\beta_i^t \ge P_k^{t+1} - P_k^t$, $\beta_i^t \ge P_k^t - P_k^t$, $\beta_i^t \le P_k^t + P_k^{t+1}$ and $\beta_i^t \le 2 - (P_k^t + P_k^{t+1})$ for $\forall i \in \mathcal{P}, \forall t \in \Xi$. Finally we allow at most one transition between two consequent time instances: $\sum_{i \in \mathcal{P}} \beta_i^t \le 1, \forall t \in \Xi$.

V. Exact Algorithm based on MILP

A. Introduction and Model

In the mission planning problem considered here, similar to [5], there are L launch sites where the UAVs are launched from and C landing sites where the UAVs can land on. There are also N targets which has to be serviced by the UAVs. We will denote the set of launch sites with $\mathcal{L} = \{\mathrm{L}_1, \ldots, \mathrm{L}_L\}$, the set of landing sites with $\mathcal{C} = \{\mathrm{C}_1, \ldots, \mathrm{C}_C\}$ and the set of targets with $\mathcal{N} = \{\mathrm{N}_1, \ldots, \mathrm{N}_N\}$. Let us define the set of departing nodes as $I = \mathcal{L} \cup \mathcal{N}$ and the set of approaching nodes as $\mathcal{J} = \mathcal{N} \cup \mathcal{C}$ for later use. We also define the set of indices indicating different vehicles as $\mathcal{K} = \{1, \ldots, K\}$.

We are given the time vehicle k needs to fly from a departing node i to an approaching node j, and denote this time with t_{ijk} for $\forall i \in I$, $\forall j \in \mathcal{J}$ and $\forall k \in \mathcal{K}$. To formulate the problem the binary decision variables x_{ijk} are defined which are 1 if UAV k has traveled from node i to j and is 0 otherwise for $i \in I$, $j \in \mathcal{J}$ and $k \in \mathcal{K}$. Also the continuous variables t_i defined for $\forall i \in I$ indicate the time that approaching node i is serviced. An other set of continuous variables defined is t_{jk} for $\forall j \in \mathcal{C}$ and $\forall k \in \mathcal{K}$. t_{jk} indicate the time for which UAV k lands on the landing site j. If UAV k does not land on landing site j then this variable is zero. A final set of continuous variables is s_{ik} defined for $i \in I$ and $k \in \mathcal{K}$. These variables indicate the time that UAV k spent on loitering around node i before leaving it. In [5] several different objective functions were utilized. We will be using the total time of the travel as the objective function in the examples, although the others can also be used as well.

Given the above definitions, MILP problem for multiple-UAV mission planning without the temporal specifications can be written as follows,

$$\min \quad f := \textstyle\sum_{j\in\mathcal{C}} \sum_{k=1}^{K} t_{jk} \tag{25}$$

$$\text{s. t.} \quad \textstyle\sum_{k=1}^{K} \sum_{j\in\mathcal{J}, j\neq i} x_{ijk} \leq 1 \qquad \forall i \in \mathcal{N} \tag{26}$$

$$\textstyle\sum_{i\in I, i\neq h} x_{ihk} - \sum_{j\in\mathcal{J}, j\neq h} x_{hjk} = 0 \qquad \forall h \in \mathcal{N},\ \forall k \in \mathcal{K} \tag{27}$$

$$\textstyle\sum_{i\in\mathcal{L}} \sum_{j\in\mathcal{N}} x_{ijk} \leq 1 \qquad \forall k \in \mathcal{K} \tag{28}$$

$$\textstyle\sum_{i\in\mathcal{L}} \sum_{j\in\mathcal{N}} x_{ijk} - \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{C}} x_{ijk} = 0 \qquad \forall k \in \mathcal{K} \tag{29}$$

$$t_i + t_{ijk} + s_{ik} - M(1 - x_{ijk}) \leq t_j \qquad \forall i \in I,\ \forall j \in \mathcal{N}, j \neq i,\ \forall k \in \mathcal{K} \tag{30}$$

$$t_i = 0 \qquad \forall i \in \mathcal{L} \tag{31}$$

$$t_i + t_{ijk} + s_{ik} - M(1 - x_{ijk}) \leq t_{jk} \qquad \forall i \in \mathcal{N},\ \forall j \in \mathcal{C},\ \forall k \in \mathcal{K} \tag{32}$$

$$t_{jk} \leq r_k \qquad \forall j \in \mathcal{C},\ \forall k \in \mathcal{K} \tag{33}$$

where M is a big enough constant. In the problem given above, (25) is the objective function. Constraints (26) ensure that every target will be serviced at most once. (27) is the flow constraint stating that every visited node must also be left. The constraints (28) ensure that if a UAV is launched it is launched from a launch site, whereas the constraints (29) are to make sure that each UAV that is launched lands at a landing site. (30,31,32) are the timing related constraints. These constraints guarantee a feasible flow of time. They also prevent the loop solutions and act as sub-tour elimination constraints. Finally, (33) ensures that each UAV lands before it is out of fuel.

B. Formulation of Propositions

Given the variables and parameters of the problem, we are now ready to present some possible propositions. Note that one can define several different atomic propositions on the problem. Our intention here is to present some examples instead of giving a complete list.

Assume that $\bar{T}$ is a big enough number. Continuous variables θ^t indicate the time of occurrence for these instances for $\forall t \in \{1, \ldots, T\}$. A feasible time flow is obtained with the constraints $\theta^1 = 0$, $\theta^i \leq \theta^{i+1}$ for $\forall i \in \{1, \ldots, T-1\}$ and $\theta^T = M_T$ where M_T is a large constant.

A possible proposition is the one which states that *Target i has been serviced*. This proposition will be denoted by P_i and is defined for $i \in \mathcal{N}$. Let us denote the value of this proposition at time instance t with P_i^t. Then the linear inequalities characterizing $P_i^t \in \{0, 1\}$ can be formulated as follows:

$$P_i^t \geq \tfrac{1}{\bar{T}}(\theta^t - t_i) \qquad \forall t \in \Xi, \tag{34}$$

$$P_i^t \leq 1 - \tfrac{1}{\bar{T}}(t_i - \theta^t) \qquad \forall t \in \Xi. \tag{35}$$

Another useful proposition is Q_k which states that *UAV k has landed* and which takes the value Q_k^t at time instance t.Then $Q_k^t \in \{0, 1\}$ can be characterized by the following inequalities.

$$Q_k^t \geq \tfrac{1}{\bar{T}}(\theta^t - \textstyle\sum_{j\in\mathcal{C}} t_{jk}) \qquad \forall t \in \Xi, \tag{36}$$

$$Q_k^t \leq 1 - \tfrac{1}{\bar{T}}(\textstyle\sum_{j\in\mathcal{C}} t_{jk} - \theta^t) \qquad \forall t \in \Xi. \tag{37}$$

A similar proposition is defined as R_{jk} stating *UAV k has landed on landing site j*. If UAV k does not land on landing site j then this proposition will always be zero. Denoting the time evolution of this proposition with $R_{jk}^t \in \{0, 1\}$, the linear inequalities characterizing this proposition can be stated as follows,

$$R_{jk}^t \geq \tfrac{1}{\bar{T}}(\theta^t - t_{jk}) - (1 - \textstyle\sum_{i\in\mathcal{N}} x_{ijk}) \qquad \forall t \in \Xi, \tag{38}$$

$$R_{jk}^t \leq 1 - \tfrac{1}{\bar{T}}(t_{jk} - \theta^t) \qquad \forall t \in \Xi, \tag{39}$$

$$R_{jk}^t \leq \textstyle\sum_{i\in\mathcal{N}} x_{ijk} \qquad \forall t \in \Xi. \tag{40}$$

A final proposition can be S_{ki} which states that *Target i has been serviced by UAV k*. This proposition can be further generalized to state *target i has been serviced by one of the UAVs in* κ where κ is a set of not necessarily identical UAVs. Then the constraint on $S_{i\kappa}^t \in \{0, 1\}$ can be expressed as follows,

$$S_{i\kappa}^t \geq \tfrac{1}{\bar{T}}(\theta_t - t_i) - (1 - \textstyle\sum_{k\in\kappa} \sum_{j\in\mathcal{J}} x_{ijk}) \qquad \forall t \in \Xi, \tag{41}$$

$$S_{i\kappa}^t \leq \tfrac{1}{\bar{T}}(t_i - \theta_t) \qquad \forall t \in \Xi, \tag{42}$$

$$S_{i\kappa}^t \leq \textstyle\sum_{k\in\kappa} \sum_{j\in\mathcal{J}} x_{ijk} \qquad \forall t \in \Xi. \tag{43}$$

C. Simulations

Let us consider a single mission as an example and observe the changes in the resulting mission plan for slight changes in the scenario. The example we consider in this section is motivated by a complex military operation, loosely inspired by the events narrated in [17]. In this specific scenario, taking place in an urban setting, a friendly unit is pinned down by enemy units, and needs to be rescued. There are three groups of enemy infantry in the scenario, denoted as T1, T2 and T3. Two of them are protected by Surface-to-Air Missile (SAM) units S1 and S2. In this setting, S1 protects only T1, and S2 protects only T2. There are four UAVs with different capabilities which are to be outlined. The UAVs are launched from one launch site L1 and can land on one of the two landing sites C1 and C2 if they are launched. See Figure 1 for a map.

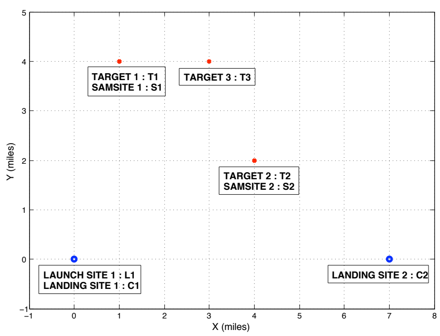

Fig. 1. Map of the "Black-Hawk-Down" scenario

Even though our model is general enough to allow using many different distance metrics and vehicle capabilities—by computing the traversal times t_{ijk} and the cost function f accordingly, in the example we used the Manhattan distance, as in [5], as a good measure of distance in urban settings. Each target also has a servicing time of 0.25 hours. The four UAVs V1,V2, V3, and V4 travel with 25mph, 25mph, 40mph and 12mph respectively. V1 and V2 can defeat the targets, but they cannot engage the SAM sites. It is also given that V1 is vulnerable to S1 whereas V2 is vulnerable to S1 and S2. V3 can destroy only the SAM sites, but it can not engage any of the other targets. V4 represents a ground unit (autonomous or otherwise) in this scenario. It travels with relatively low speed, but it can destroy any of the sam sites or targets. In addition, it carries medical supplies that can be used to treat wounded personnel in the unit to be rescued.

The mission in this case is to destroy either T1 and T3, or T2 and T3. Thus making a way through for the infantry from T1 side or T2 side. There is also another specification which is that if the rescued unit escapes from T2 side and reaches the friendly base C2 then V4 must meet the rescued unit there with necessary health supplies.

These constraints can be formalized as follows. The mission objective is to eventually destroy T3 and also destroy either T1 or T2 which can be denoted in LTL$_{-X}$ as $(\Diamond P_{T1} \vee \Diamond P_{T2}) \wedge (\Diamond P_{T3})$. We have also noted that if T2 is destroyed then V4 has to land on C2 which can be expressed in LTL$_{-X}$ as $(\Diamond P_{T2}) \rightarrow (\Diamond Q_{C2,V4})$. The mission constraints state that if T1 is serviced by V1 or V2 then S1 has to be destroyed beforehand. This specification can be expressed in LTL$_{-X}$ as $(\neg S_{T1,\{V1,V2\}})\mathcal{W}(P_{S1})$ which states that V1 or V2 can not visit T1 unless S1 is destroyed. This specification can be written as $(\Box\neg S_{T1,\{V1,V2\}}) \vee (\neg S_{T1,\{V1,V2\}})\mathcal{U}(P_{S1})$. Similarly we have that $(\neg S_{T2,V2})\mathcal{W}(P_{S2})$. Finally it was specified that V1 and V2 can not destroy the sam sites. This specification can be expressed in LTL$_{-X}$ as $(\Box\neg S_{V1,S1}) \wedge (\Box\neg S_{V1,S2}) \wedge (\Box\neg S_{V2,S2}) \wedge (\Box\neg S_{V2,S2})$. Also note that V3 can not destroy any of the targets which results in $(\Box\neg S_{V3,T1}) \wedge (\Box\neg S_{V3,T3}) \wedge (\Box\neg S_{V3,T3})$. Then the overall LTL specification can be obtained by considering the conjunction of all these constraints which is,

$$\begin{aligned}\phi = [(\Diamond P_{T1} \vee \Diamond P_{T2}) \wedge (\Diamond P_{T3})] \wedge [(\Diamond P_{T2}) \rightarrow (\Diamond Q_{C2,V4})] \\ \wedge[(\Box\neg S_{T1,\{V1,V2\}}) \vee (\neg S_{T1,\{V1,V2\}})\mathcal{U}(P_{S1})] \\ \wedge[(\Box\neg S_{T2,V2}) \vee (\neg S_{T2,V2})\mathcal{U}(P_{S2})] \\ \wedge[(\Box\neg S_{V1,S1}) \wedge (\Box\neg S_{V1,S2}) \wedge (\Box\neg S_{V2,S2}) \wedge (\Box\neg S_{V2,S2})] \\ \wedge[(\Box\neg S_{V3,T1}) \wedge (\Box\neg S_{V3,T3}) \wedge (\Box\neg S_{V3,T3})]. \quad (44)\end{aligned}$$

The objective function was selected to be the total time that UAVs travel. This objective function can be used for representing a kind of risk since it represent the total time the UAVs were used. Several other objective functions as given in [5] or a mixture of those can be used as well.

The solution of this mission is given in Figure 2. The resulting mission plan is to use only V4 and destroy T3 and T1 respectively. Notice that since V4 is not vulnerable to any of the sam sites, S1 was not destroyed. Considering the risk factor, the mission plan does not utilize all the UAVs.

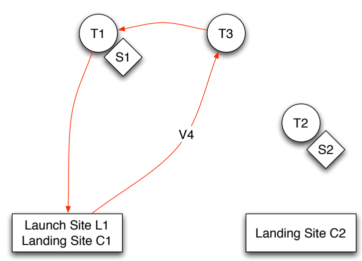

Fig. 2. Simulation 1 for the military scenario

Now we look at a case in which V4 is 2mph slower, i.e. it travels with 10mph. This time the solution is given in Figure 3. The solution in this case is to destroy T1 and T3 respectively by V1. Since V1 can be hit by S1, S1 is destroyed by V3 before V1 services T1.

In an other example, consider a case in which T1 and S1 are 4 miles to the north. This time the solution is given in Figure 4. This time it is more advantageous to destroy T2 and T3 for which V4 was used for this purpose. Notice that V4 finally lands on C2 in order to deliver the necessary cargo.

A final example is that V4 travels with 8mph this time, i.e. 2mph even more slower. In this case the resulting optimal mission plan is given in Figure 5. Similarly T2 and T3 are destroyed, this time by V1. Note that V1 is not vulnerable

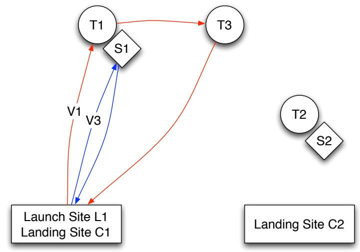

Fig. 3. Simulation 2 for the military scenario

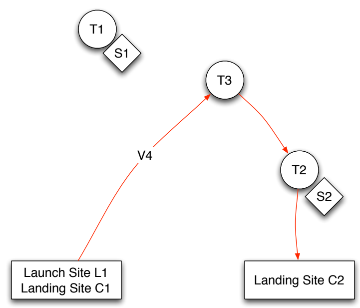

Fig. 4. Simulation 3 for the military scenario

to S2. Notice also that V4 lands on C2 just to deliver the necessary cargo without destroying any of the targets.

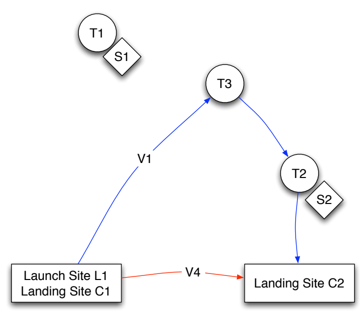

Fig. 5. Simulation 4 for the military scenario

We should note that none of the simulations take more than two seconds on a computer with two 2.66 GHz processors and 4GB RAM running AMPL with CPLEX 10.2.0 parallel.

There are certainly other capabilities of the method that were not addressed in this example. Assume for instance that a target has to be first destroyed and then destruction must be verified. One can easily put two targets at the same place denoting these tasks. Then a first constraint is that destruction should not be verified unless the target is destroyed. One can even state that if the target was destroyed by a specific vehicle then verification is necessary otherwise it is not. These kind of even more complex objectives are straightforward to implement within the given framework.

VI. Conclusions and Future Work

In this paper we have addressed relative timing constraints for Multiple-UAV Mission Planning Problems, emphasizing the need for a specification language to specify complex constraints naturally. We have developed a general class of algorithms for solving complex Multiple-UAV Mission Planning Problems with complex constraints, expressed using the LTL_{-X} Temporal Logic specification language. These algorithms are based on standard techniques used to model Vehicle Routing Problems using MILP, integrated with a novel procedure to formulate LTL_{-X} specifications as mixed-integer linear constraints. The expressive power of LTL_{-X} have been shown with a mission planning example.

Our future research will mostly focus on search for efficient algorithms with provable performance guarantees. Such algorithms would provide solutions for large-scale problems including several targets and UAVs.

Acknowledgments

This work was supported in part by the Michigan/AFRL Collaborative Center on Control Sciences, Air Force grant no. FA 8650-07-2-3744. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the supporting organizations.

References

[1] M. Alighanbari, Y. Kuwata, and J.P. How. Coordination and control of multiple uavs with timing contraints and loitering. In *American Control Conference*. IEEE, 2003.

[2] P.R. Chandler, M. Pachter, D. Swaroop, J.M. Fowler, J.K. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard. Complexity in uav cooperative control. In *American Control Conference*, 2002.

[3] J. Bellingham, M. Tillerson, A. Richards, and J.P. How. *Coopertive Control: Models, Applications and Algorithms*, chapter Multi-Task Allocation and Path Planning for Cooperating UAVs. Kluwer Academic Publishers, 2001.

[4] C. Schumacher, P.R. Chandler, M. Pachter, and L.S. Patcher. Optimization of air vehicles operations using mixed-integer linear programming. *Journal of the Operational Research Society*, 58:516–527, 2007.

[5] A.L. Weinstein and C. Schumacher. Uav scheduling via the vehicle routing problem with time windows. Technical Report AFRL-VA-WP-TP-2007-306, Air Force Research Laboratory, January 2007.

[6] E.A. Emerson. *Handbook of Theoretical Computer Science*, chapter Temporal and Modal Logic. Elsevier, 1995.

[7] A. Pnueli. The temporal logic of programs. In *18th annual IEEE-CS Symposium on Foundations of Computer Science*, pages 46–57, 1977.

[8] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, 1992.

[9] E.M. Clarke, O. Grumberg, and D.A. Pled. *Model Checking*. MIT Press, Cambridge, MA, 1999.

[10] D.M. Gabbay, I. Hodkinson, and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects*, volume 2 of *Oxford Logic Guides*, chapter Intervals and Planning. Oxford Science Publications, 1994.

[11] A. Fusaoka, H. Seki, and K. Takahashi. A description and reasoning of plant controllers in temporal logic. In *Proceedings of the 8th International Joint Conference in Artificial Intelligence*, pages 405–408, August 1983.

[12] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control (accepted)*, 2007.

[13] P. Tabuada and G.J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.

[14] R.G. Jeroslow. *Logic Based Decision Support: Mixed Integer Model Formulation*. Elsevier Science, 1989.

[15] H.P. Williams. Logic applied to integer programming and integer programming applied to logic. *European Journal of Operational Research*, 81:605–616, 1995.

[16] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[17] M. Bowden. *Black Hawk Down: A Story of Modern War*. Atlantic Monthly Press, Berkeley, CA, 1999.