Dynamic Vehicle Routing with Complex Mission Specifications

Cooperative Control of Unmanned Air Vehicles (C^2UAV) Concentration

Emilio Frazzoli

R. Sanfelice (Post-doc) S. Karaman (PhD) C. Oravetz (MS)

Laboratory for Information and Decision Systems
Aeronautics and Astronautics
Massachusetts Institute of Technology

MAX Kick-Off Meeting
Ann Arbor, MI
August 29, 2007

Outline

1. Overview
2. Background
   - Dynamic Vehicle Routing
   - Vehicle Routing with Differential Constraints
3. Preliminary Results
   - Perimeter Defense
   - Vehicle Routing with Complex Mission Specifications
4. Proposed research
   - Research objectives
   - Experimental facilities

Challenges

Mission Profiles

- Complex Task Specifications.
- Heterogeneous vehicles.
- UAV Dynamics.
- Uncertain, Stochastic or Adversarial Environment.
- Persistence.
- Limited Sensing and Communication.

Challenges

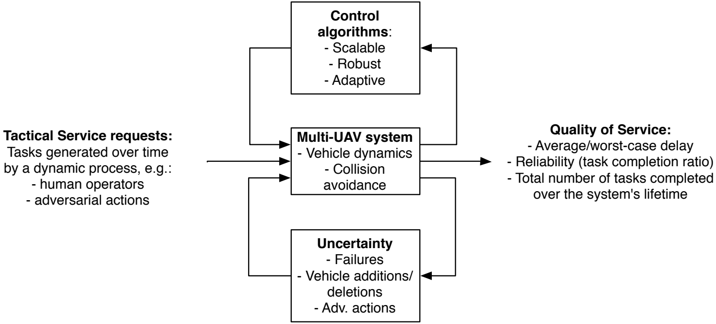
Mission Profiles

- Complex Task Specifications.
- Heterogeneous vehicles.
- UAV Dynamics.
- Uncertain, Stochastic or Adversarial Environment.
- Persistence.
- Limited Sensing and Communication.

Objectives

- Correctness/performance guarantees.
- Scaling effects (performance, complexity).
- Robustness, Learning, Adaptation.

An "input/output" view



- UAV network as shared, persistent infrastructure.
- Given a parameter describing the "input" from a certain class, compute the system's achievable "performance."
- Human as a "user" or "customer:" provide judgment in choosing mission objectives, as opposed to mission plans.

Dynamic Vehicle Routing: a (not so) basic problem

"User" model

- Exogenous process generating "service requests" located at points in a region of interest ("targets")
- Service requests are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and a spatial pdf φ. (Assume $\int_{\mathcal{Q}} \varphi \, dq = 1$).
- Service requests fulfilled when visited by a vehicle.

Dynamic Vehicle Routing: a (not so) basic problem

"User" model

- Exogenous process generating "service requests" located at points in a region of interest ("targets")
- Service requests are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and a spatial pdf φ. (Assume $\int_{\mathcal{Q}} \varphi \, dq = 1$).
- Service requests fulfilled when visited by a vehicle.

System model

- A team of m identical planar vehicles, moving with speed bounded by 1.

Dynamic Vehicle Routing: a (not so) basic problem

"User" model

- Exogenous process generating "service requests" located at points in a region of interest ("targets")
- Service requests are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and a spatial pdf φ. (Assume $\int_{\mathcal{Q}} \varphi \, dq = 1$).
- Service requests fulfilled when visited by a vehicle.

System model

- A team of m identical planar vehicles, moving with speed bounded by 1.

Performance Criterion

- QoS: Average time between issuance of service requests and their fulfillment.

Algorithm Design and Analysis

A spatially-decentralized algorithm

- Associate to each agent a weighted virtual generator $(g, w)_i$, and partition the workspace with a Power Diagram (generalizes Voronoi diagrams).
- Each agent updates its virtual generator according to the (negative) gradient of $J(g, w) = \sum_{i=1}^{m} \left[\left(\int_{V_i(g,w)} \sqrt{\varphi(q)}\, dq \right)^2 + \frac{1}{\lambda} \int_{V_i(g,w)} \|q - g_i\| \varphi(q)\, dq \right]$
- Within own region, each agent repeats
  1. Find the densest cluster with at least a fraction $\eta \in (0, 1]$ of the outstanding targets;
  2. Visit these targets efficiently (TSP-like).

Algorithm Design and Analysis

A spatially-decentralized algorithm

- Associate to each agent a weighted virtual generator $(g, w)_i$, and partition the workspace with a Power Diagram (generalizes Voronoi diagrams).
- Each agent updates its virtual generator according to the (negative) gradient of
$$J(g, w) = \sum_{i=1}^{m} \left[\left(\int_{V_i(g,w)} \sqrt{\varphi(q)}\, dq \right)^2 + \frac{1}{\lambda} \int_{V_i(g,w)} \|q - g_i\| \varphi(q)\, dq \right]$$
- Within own region, each agent repeats
  1. Find the densest cluster with at least a fraction $\eta \in (0, 1]$ of the outstanding targets;
  2. Visit these targets efficiently (TSP-like).

Theorem

The average system time of service requests is (locally) optimal for $\lambda \to 0^+$, and satisfies

$$\bar{T} \leq \frac{\beta^2 \lambda}{m^2 (2 - \eta)} \left(\int_{\mathcal{Q}} \sqrt{\varphi(q)}\, dq \right)^2 \leq 1.8\, \bar{T}^*, \qquad \text{for } \lambda \to +\infty.$$

Vehicle Routing with Differential Constraints

- What happens if the vehicles are subject to non-integrable differential constraints on their motion?
  - Minimum turn radius, constant speed (UAVs, Dubins cars)
  - Minimum turn radius, able to reverse (Reeds-Shepps cars)
  - Differential drive robots (e.g., tanks).
  - Bounded acceleration vehicles (e.g., helicopters, spacecraft).

- Fundamentally different problems, combining combinatorial task specifications with differential geometry and optimal control.

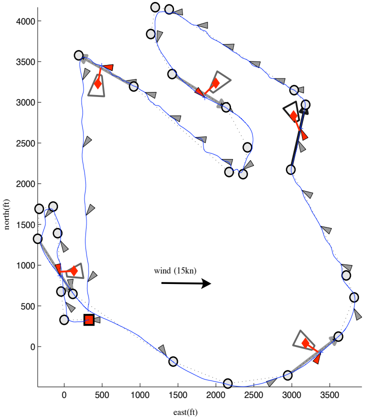
- Decompose the problem, study the asymptotic cases:
  - Heavy load: the "Dubins Traveling Salesperson Problem."
  - Light load: optimal loitering patterns.

Results (heavy- and light-load cases)

- System time $\bar{T} = \Theta(\lambda^2/m^3)$.

- Turning radius results in an additive approximation to the system time.
- Additive penalty may be reduced by ad-hoc teaming.

Vehicle Routing with Differential Constraints and Wind



Airborne camera TSP in wind

- "Fly the camera" through given targets.
- No access to the autopilot, only waypoint commands.
- Reduction to an Asymmetric TSP, approximate solution.

Joint work with J. Enright (UCLA), N. Ceccarelli, S. Rasmussen, and C. Schumacher (AFRL/VACA)

Recent extensions

Vehicle Routing with Target Impatience

- What if targets may disappear after some “impatience” time, itself a random variable?
- Quality of Service criterion: probability of “missing” a target due to impatience.
- Result: constructive characterization of the minimum number of vehicles needed to ensure a given QoS.

Recent extensions

Vehicle Routing with Target Impatience

- What if targets may disappear after some "impatience" time, itself a random variable?
- Quality of Service criterion: probability of "missing" a target due to impatience.
- Result: constructive characterization of the minimum number of vehicles needed to ensure a given QoS.

Vehicle Routing with no Communications

- What if agents cannot communicate or even see one another?
- Designed an algorithm that achieves exactly the same performance as the full-communication algorithm $\Rightarrow$ Communications do not improve performance (but improve the transient).

Recent extensions

Vehicle Routing with Target Impatience

- What if targets may disappear after some "impatience" time, itself a random variable?
- Quality of Service criterion: probability of "missing" a target due to impatience.
- Result: constructive characterization of the minimum number of vehicles needed to ensure a given QoS.

Vehicle Routing with no Communications

- What if agents cannot communicate or even see one another?
- Designed an algorithm that achieves exactly the same performance as the full-communication algorithm ⇒ Communications do not improve performance (but improve the transient).

Sensor-based Vehicle Routing

- What if agents can only sense targets within a given sensor range?
- In the light-load case, system time dominated by search. Optimality through optimal search patterns.
- In the heavy-load case, sensor limitations do not impact system's performance.

Perimeter Defense

"User" model

- Exogenous process generating "intruders" entering the workspace boundary.
- Intruders are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and an angular pdf φ. (Assume $\int_{\mathcal{S}^\infty} \varphi(\theta)\, d\theta = 1$).
- Intruders eliminated when "tagged" by a vehicle.

Perimeter Defense

"User" model

- Exogenous process generating "intruders" entering the workspace boundary.
- Intruders are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and an angular pdf φ. (Assume $\int_{\mathcal{S}^\infty} \varphi(\theta)\, d\theta = 1$).
- Intruders eliminated when "tagged" by a vehicle.

System model

- A team of m identical vehicles, moving on the plane with bounded speed.

Perimeter Defense

"User" model

- Exogenous process generating "intruders" entering the workspace boundary.
- Intruders are generated by a spatio-temporal Poisson process with time intensity $\lambda > 0$, and an angular pdf φ. (Assume $\int_{\mathcal{S}^\infty} \varphi(\theta)\, d\theta = 1$).
- Intruders eliminated when "tagged" by a vehicle.

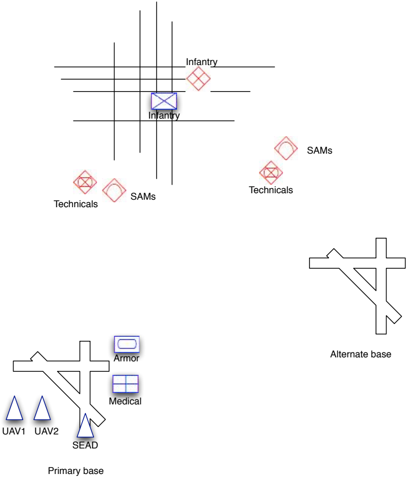
System model

- A team of m identical vehicles, moving on the plane with bounded speed.

Performance Criterion

- QoS: Radius of the protected area.
  (Prob. of trespassing less than a given $\varepsilon > 0$).

A Complex multi-UAV Mission



Mission specs

- Infantry unit pinned down by insurgents in an urban area.
- Egress routes blocked by technicals, protected by SAM units.
- Help infantry unit to reach a base with a medic in minimum time/minimum total flight time.

Friendly units

- Two UAVs capable of taking out ground targets, but vulnerable to SAMs.
- One SEAD UAV.
- One armored unit.
- One medical unit.

Linear Temporal Logic as a Mission Optimization Language

LTL_{-X} basics

Operators:

- Boolean operators: NOT ($\neg$), AND ($\wedge$), OR ($\vee$).
- Additional operators: ALWAYS ($\Box$), EVENTUALLY ($\diamond$), UNTIL ($\mathcal{U}$), WEAK UNTIL ($\mathcal{W}$).

LTL can be used to write complex multi-UAV mission specs

- LTL (or its version LTL_{-X}) is very expressive, and remarkably close to natural language.
- For example, the condition that *"UAV1 and UAV2 cannot engage Technical1 until SAM1 is destroyed by either SEAD or Armor"* can be written as:

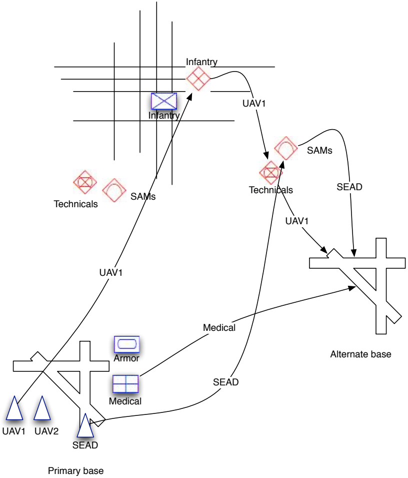
$$(\neg(UAV1@Technical1 \vee UAV2@Technical1))\mathcal{W}(SEAD@SAM1 \vee Armor@SAM1).$$

- LTL widely used for verification of embedded systems and software.

Is LTL amenable to mathematical programming?

- The model checking community has developed feasibility analysis tools, i.e., tools that find an "execution" that satisfies (or, better, falsifies) a certain condition.
See, e.g., Pappas' and Belta's recent work for applications to robotics.
- Can we find, among all executions that satisfy our mission specification, one with minimum cost?

Optimal solution



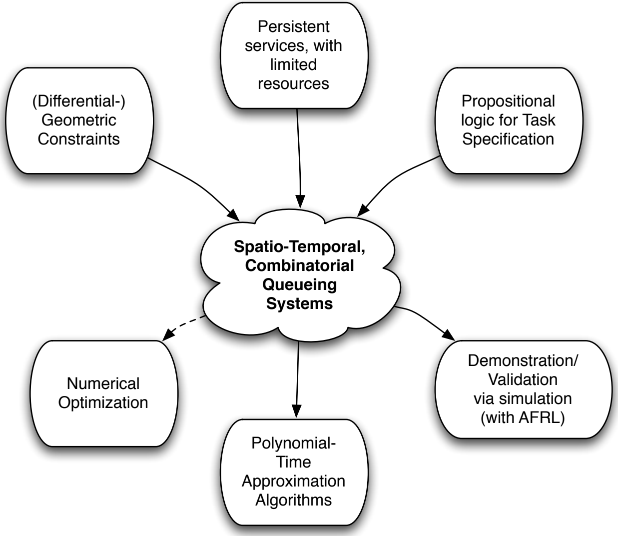
Automatic Reduction of LTL-Optimization problems into MILPs

- Novel systematic approach to write LTL specifications exactly as mixed-integer linear constraints.
- Automated tools under development

Numerical Experiments

- Qualitative aspects of the solution change considerably depending on small changes of, e.g., the vehicles' speeds or target locations (i.e., the solution is not trivial).
- ILOG CPLEX solves this problem in under 2 seconds in all our experiments..

Research Objectives



Cyber FlightCage

Movie courtesy of Prof. Jonathan How and the MIT Aerospace Controls Laboratory

- Funded by a 2007 AFOSR DURIP award.
- Faculty: J. How (PI), E. Frazzoli, N. Roy, R. Tedrake.
- Objective: Extend the existing MIT facilities to larger spaces and a broader array of vehicles/sensors.
- *No funding through MAX.*