Planning and execution of UAV surveillance missions

Bjørnar Luteberget

2024-11-11

SINTEF

Teknologi for et bedre samfunn

Vehicle fixed-rate production missions

Problem setting:

- Vehicles with limited battery capacity
- A base station with an unlimited supply of fresh batteries
- Some locations where an agent can collect reward for spending time on a task.

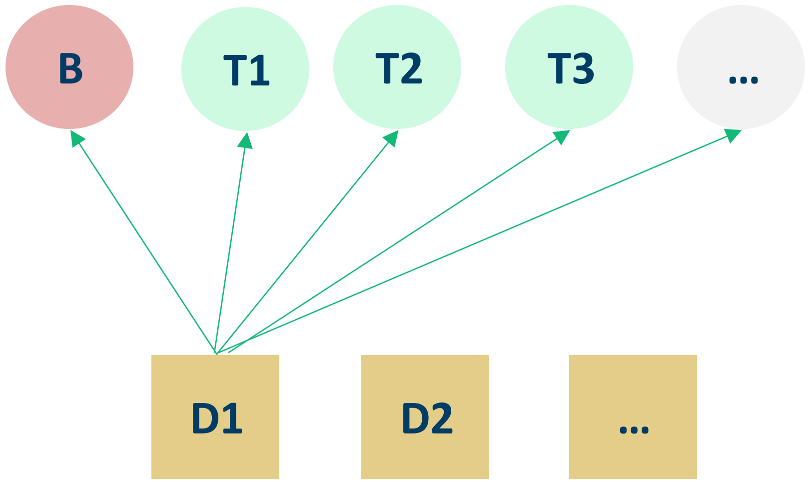
Applications: “endurance” type missions, e.g., surveillance of industry facilities, emergency response,

Photo: FFI

Surveillance mission action space

Execution model: simple action model:

- Go back to base and get a fresh battery
  - Two UAVs should not take off or land at the same time.
- Approach and follow a target (fixed or movable)
  - Available targets change when movable targets appear within sight.



Surveillance mission vehicle transition graph

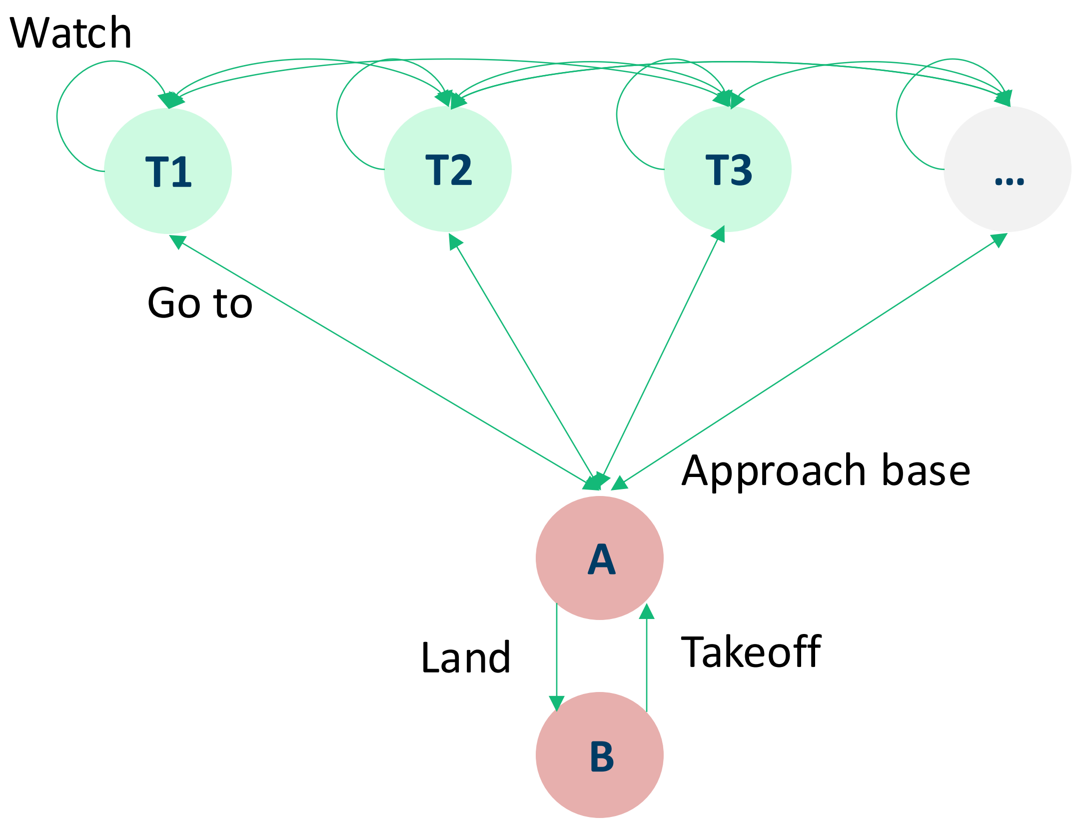
Planning model actions:

- Takeoff
- Go to target (distance matrix)
- Watch target (receive **reward** per time unit)
- Approach base
- Land

State:

- UAVs: location and battery level

Online-problem: targets and distance change



Surveillance missions

Planning problem challenges:

- Multiple vehicles, multiple locations
- Limited battery, multiple recharge cycles.
- Temporal and numerical characteristics.

Weakness of using domain-independent planning solvers:

- Only optimized solutions make sense (distance matrix, long-duration actions, etc.)

Weakness of using optimization solvers (integer programming / constraints):

- Unknown number of actions

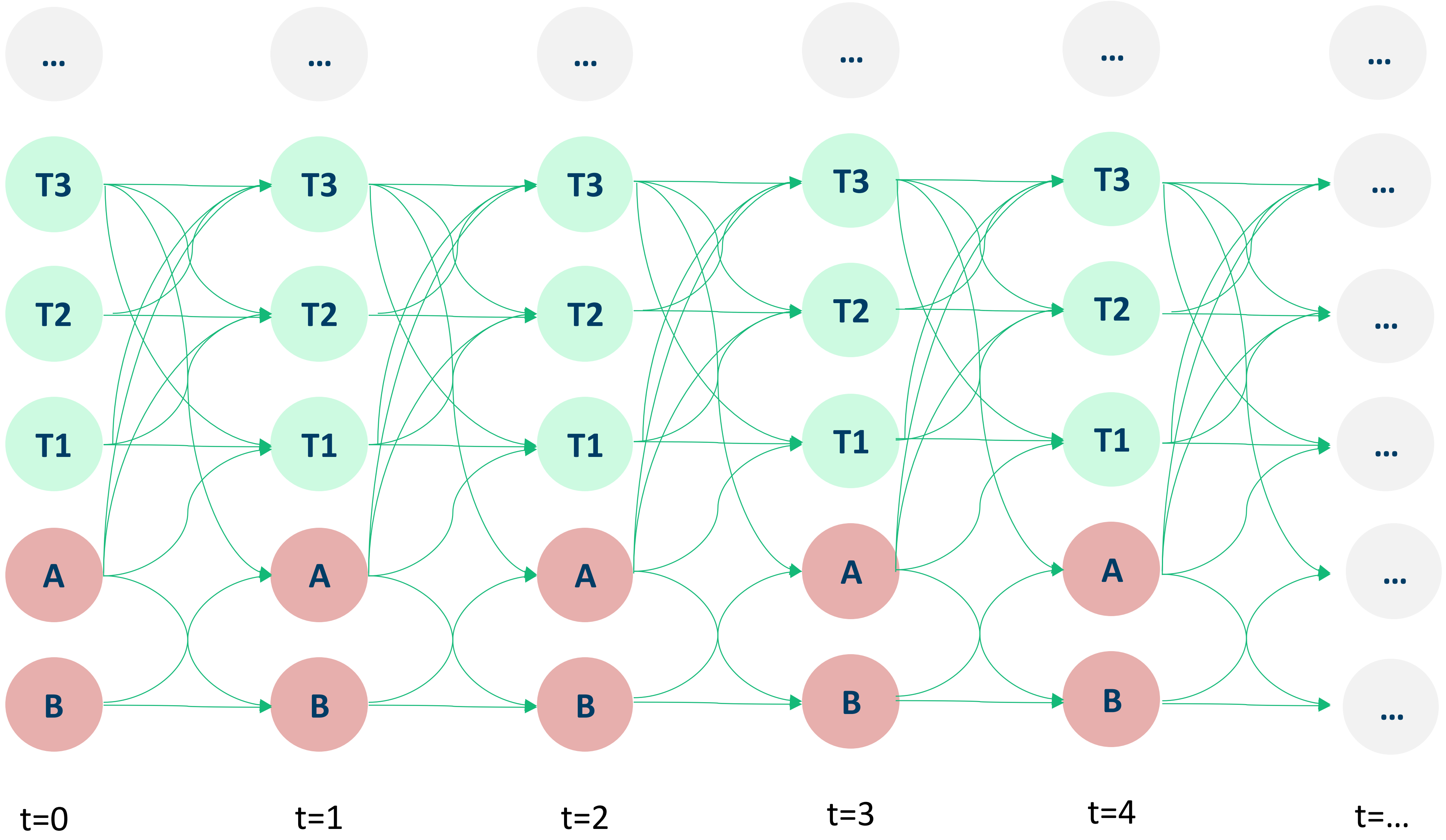
Two ideas and an algorithm

- **1. The time-expanded graph** removes the unkown number of actions and unknown durations

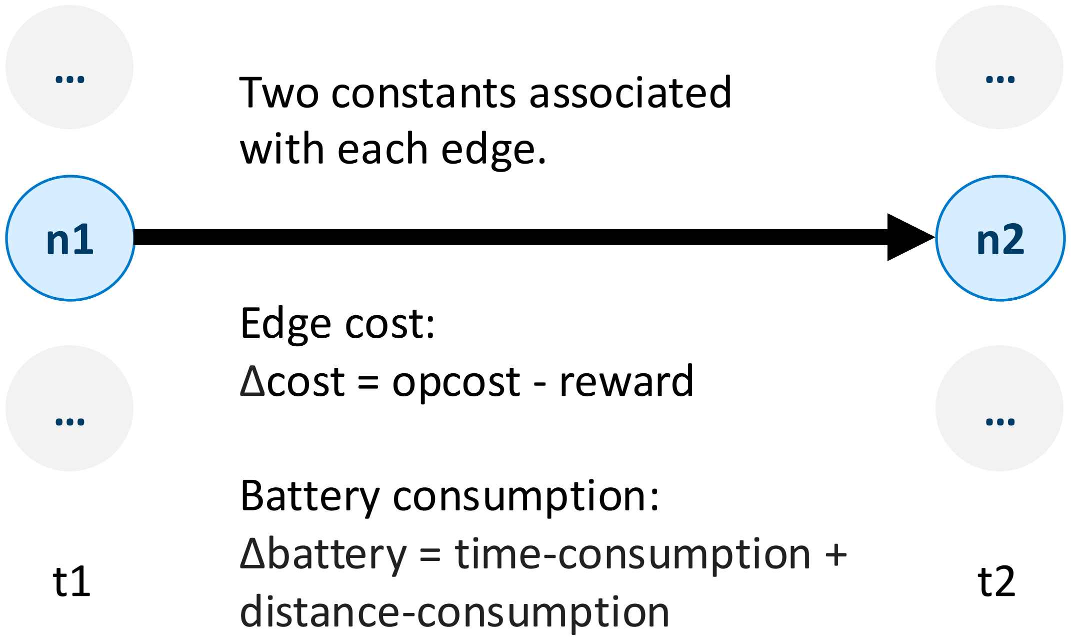
- **2. A problem decomposition** lets us solve easier planning problems and use them as building blocks in the overall planning

A greedy algorithm using these ideas lets us plan
- reasonably well
- very fast.

The time-expanded graph



The time-expanded graph

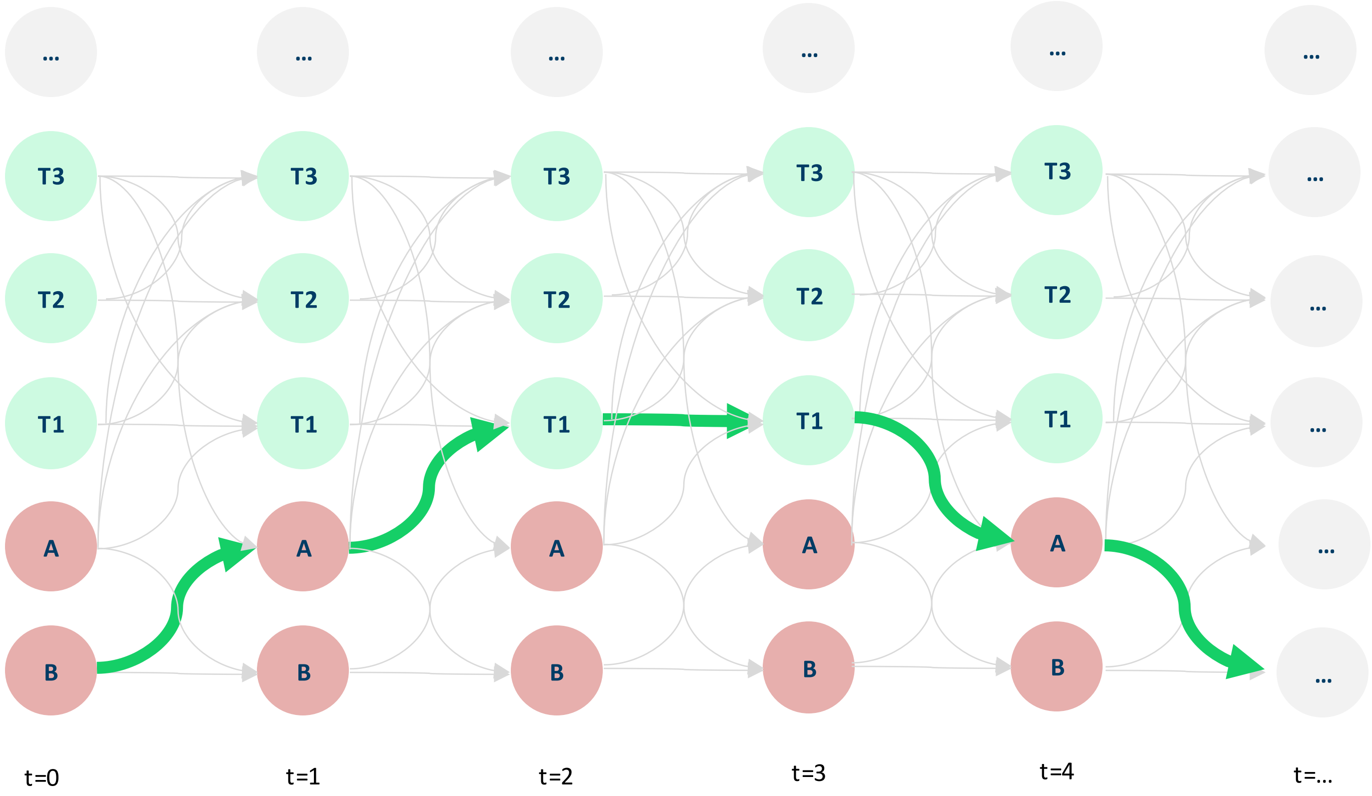


Decomposition ideas

- Disregard battery consumption:
  => ***min-cost flow*** *problem on the time-expanded graph*
- Planning for a single agent only (1 UAV):
  => ***resource-constrained shortest path*** *problem on the time-expanded graph*
- **Planning for a single agent (1 UAV) and a single battery cycle only**
  => *resource-constrained shortest path problem on the time-expanded graph*

Battery cycles decomposition

- A *battery cycle* is a path in the time-expanded graph,
  - starting and ending at the base,
  - being battery-feasible



Battery cycles decomposition

- Assume we have *all possible* battery cycles at hand:
- The planning problem is to select a subset of these, such that:
  - For a specific time step, only N (number of UAV) plans overlapping this timestep are selected
  - For a reward edge, only 1 plan using this edge is selected.

- Could be solved **exactly** using
  linear programming, column generation, branch-and-price.
- Even faster: greedily select battery cycles.

Greedy algorithm

1. Set initial agent capacities and reward edge capacities.
2. Find the best battery cycles with these capacities.
3. If the best battery cycle is trivial (do nothing), terminate.
4. Fix the selection of the best battery cycle and update capacities.

Demo videos

Conclusions

- Finding the right decomposition can be crucial for performance of greedy algorithms.
- Frequent replanning can cause practical problems with plan stability.
  - Symmetry (rounding causes equal values)
  - Variations in estimates
  - (can use explicit stabilizing terms)

Teknologi for et bedre samfunn