Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security

Fang et al.

The Problem

- Poaching is a big problem!
- Tigers, elephants, rhinos, and many more
- Patrols can be used to combat poaching
- Limited resources
- How can we allocate patrols optimally?

Figure 1: A picture of a snare placed by poachers.

PAWS-Initial

- First attempt: PAWS-Initial
- Game theoretic approach to planning patrol routes
- Models poaching as repeated Stackelberg security game
- Poachers are attackers and patrollers are defenders
- Produces suggested areas to protect
- Uses data from previous patrols to optimize

PAWS-Initial: Stackelberg Security Game

- Conservation area modeled as a grid
    - 1km x 1km
- Each cell is a potential target
- Payoff of a cell is determined by animal density
- Defender chooses a strategy, can be mixed, of which cells to protect
- Attacker observes defender’s strategy, then attacks a target
- Attack: snare, poacher, poaching camp, etc
- Zero-sum game
- Attacker uses bounded rationality for decisions
    - SUQR model

PAWS-Initial: Gathering Data

- Patrollers gather data and photos from patrol route
- Human signs and animal signs
- This data is used to infer human activity and animal density
- Patrols improve over time

(a) Tiger sign (Nov. 2014)

(b) Human sign (lighter; Jul. 2015)

(c) Human sign (old poacher camp; Aug. 2015)

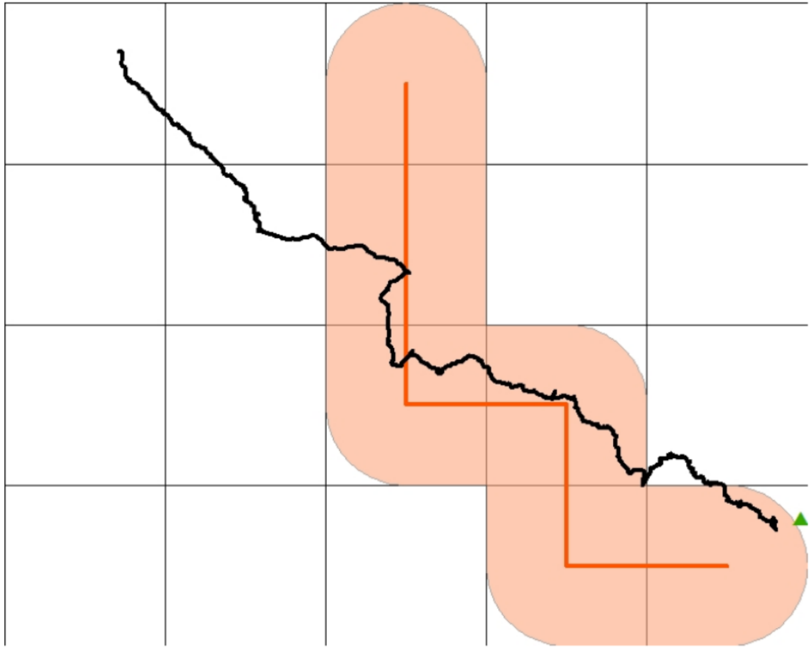
(d) Human sign (tree marking; Aug 2015)

PAWS-Initial: What it Did Well

- Information from previous patrols/rounds used in subsequent rounds
  - Effectiveness improves over time
- Bounded rationality better models real attackers
  - Attackers are not completely rational

PAWS-Initial: Issues

- Didn’t account for topography
  - “completely unanticipated” - Really?
  - Patrol routes crossed large bodies of water, extreme slopes, etc
- Didn’t account for uncertainty in animal locations
  - We don’t always know where they are
- Not scalable
  - Too slow for large areas
- Chooses targets but not routes
  - Set of targets not necessarily practical



(a) Deployed route

Accounting for Topography

- Divide each cell into 50m x 50 m “Raster Pieces”
- These record topographic information
  - Elevation, water, etc
  - Derived from topographical map input
- Account for elevation changes using standard hiking difficulty functions
- Account for extra difficulty as added distance, bound total distance
- Identify and prioritize “preferred terrain features”
  - Ridgelines, river banks
  - Easier to traverse, often followed by animals and poachers
- Route distance limits

Improving Scalability

- “Street map” approach
- Map area as set of nodes and edges
  - Nodes are small groups of raster pieces with significance
  - Edges are easiest topographical path between them
- Allows scalability while still providing 50m resolution

Accounting for Uncertainty

- Interval uncertainty used to model unknown animal locations
- Payoffs are known to lie within a certain interval
- Cells patrolled more frequently have less uncertainty
  - More information gathered

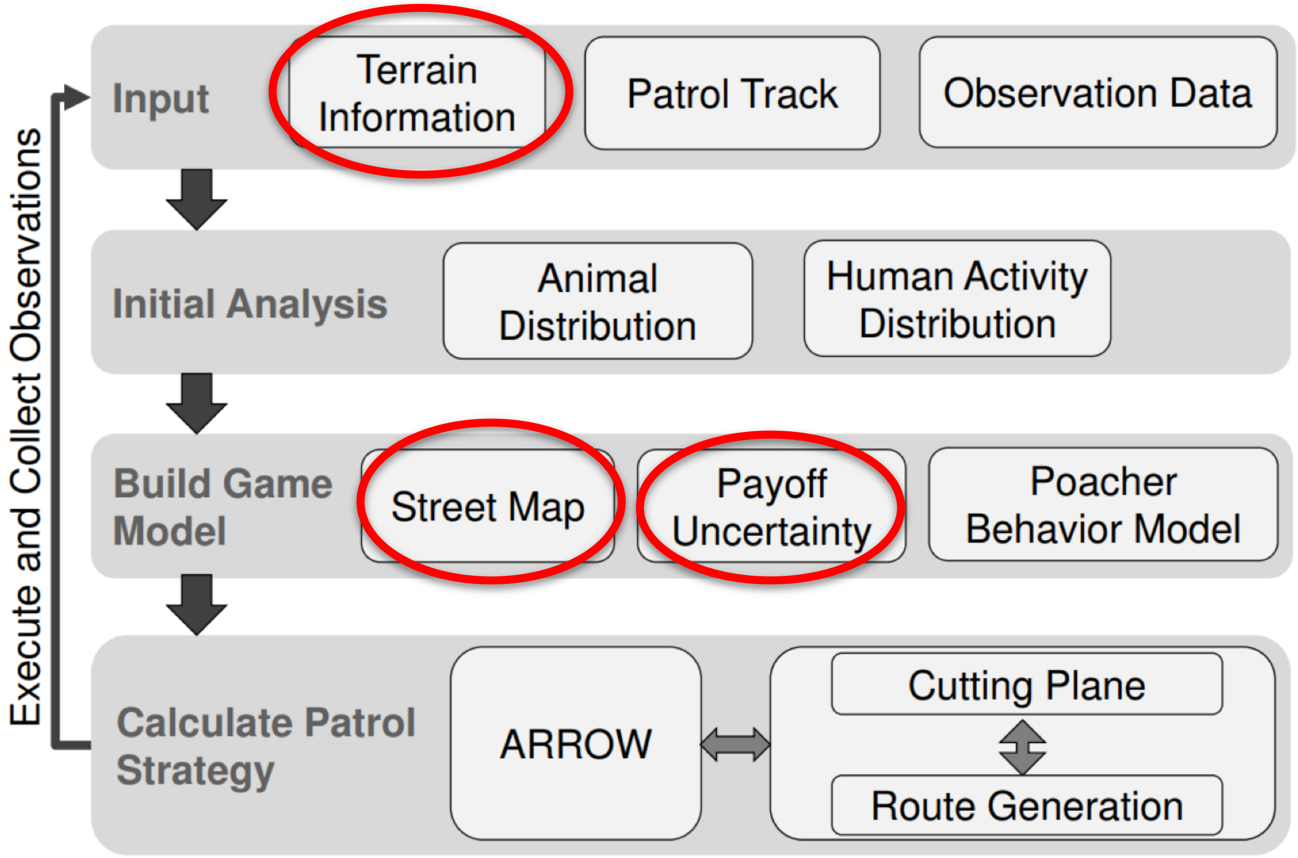


Figure 5: PAWS Overview

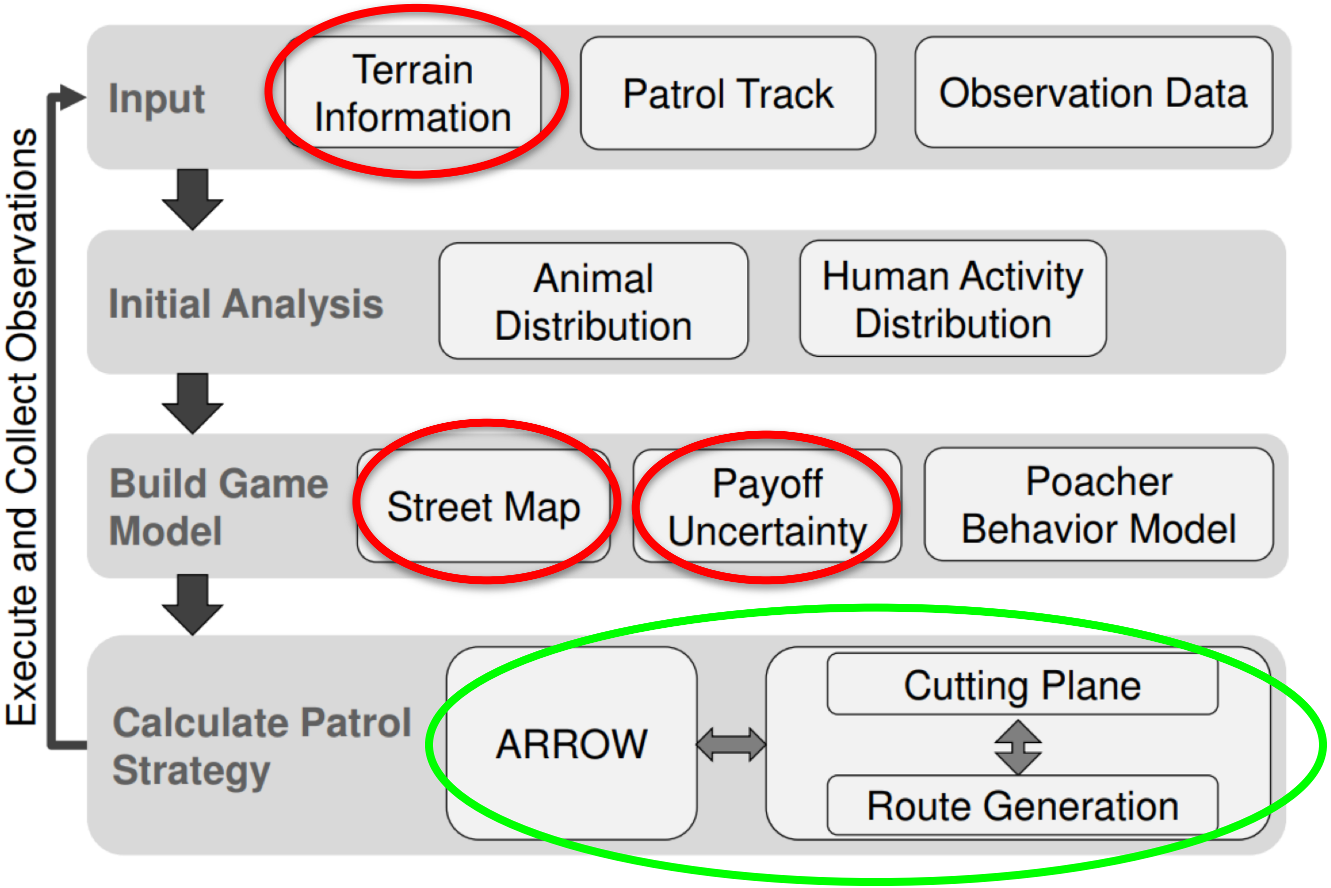


Figure 5: PAWS Overview

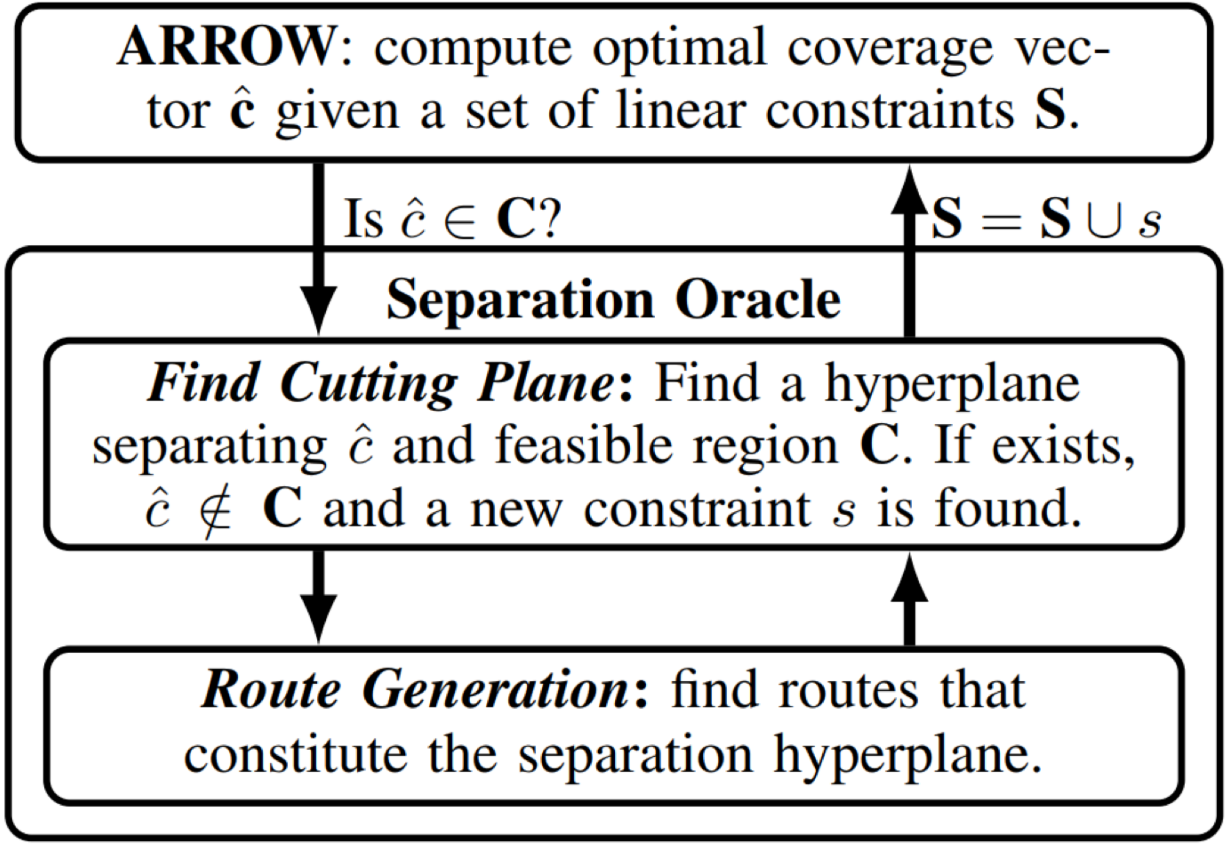


Figure 7: New integrated algorithm

Making it All Work Together: ARROW

- ARROW algorithm for payoff uncertainty and bounded rationality
  - Behavioral minimax regret
- Naive Solution: Find route that minimizes maximum regret for defenders
- But there are too many routes
- Instead, find optimal set of targets to protect WITHOUT considering route
- Coverage vector: List of targets and probability of defending them
- Then, check if coverage vector is satisfiable.
  - If so, done!
  - If not, refine solution.

Making it All Work Together: BLADE

- Given coverage vector, check if there is a valid route that satisfies it.
- Iteratively generate routes until:
  - We find a constraint verifying it's impossible
  - We have enough valid routes to match the coverage vector probabilities with a mixed strategy
- Avoids enumerating all routes
  - Scalable
- If not possible, return a constraint to ARROW
  - This is a cutting plane through
  - This allows ARROW to refine its solution
- Routes generated using S-algorithm for orienteering
  - Local search over large number of possible routes
  - Approximates optimal route

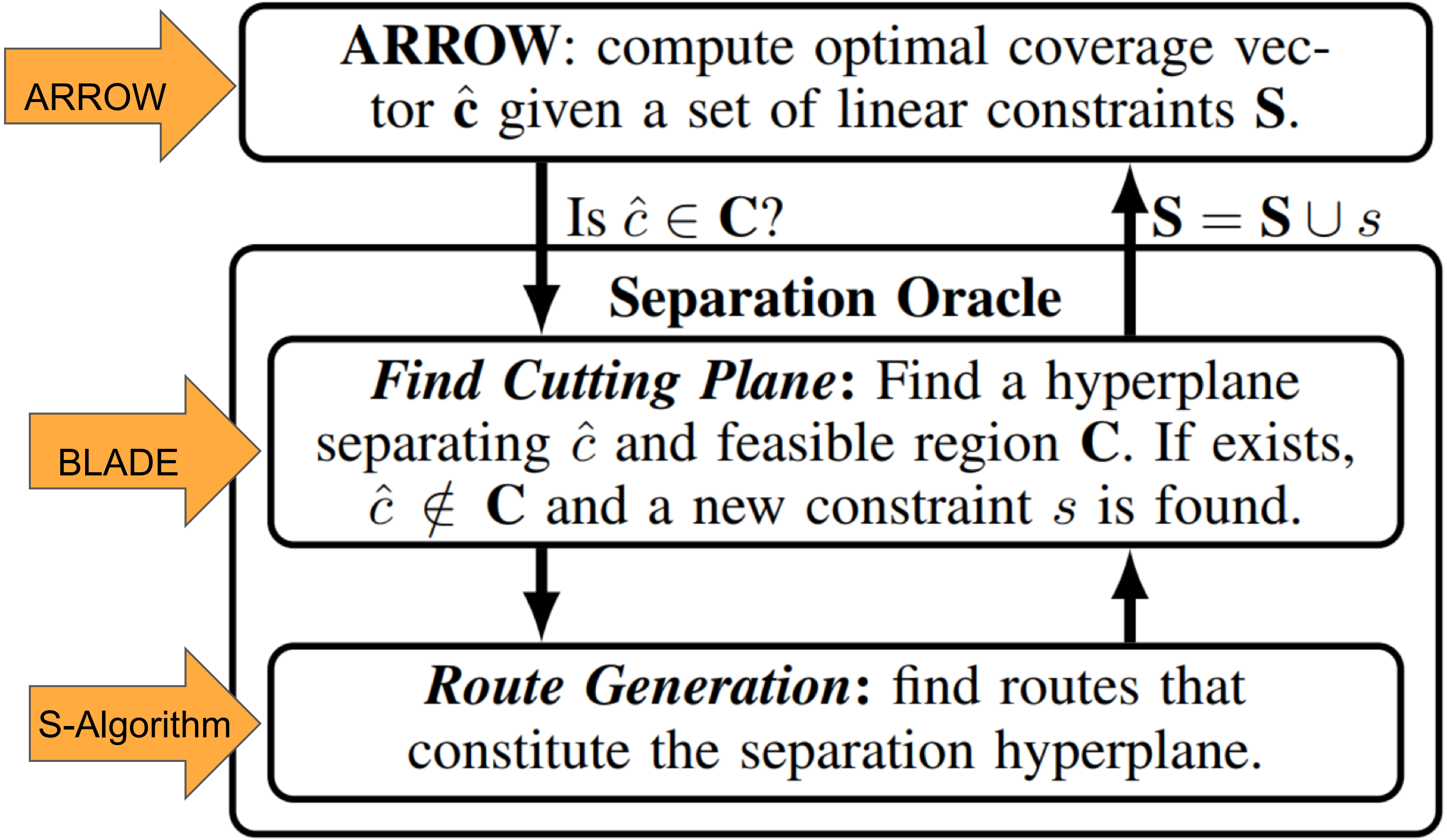


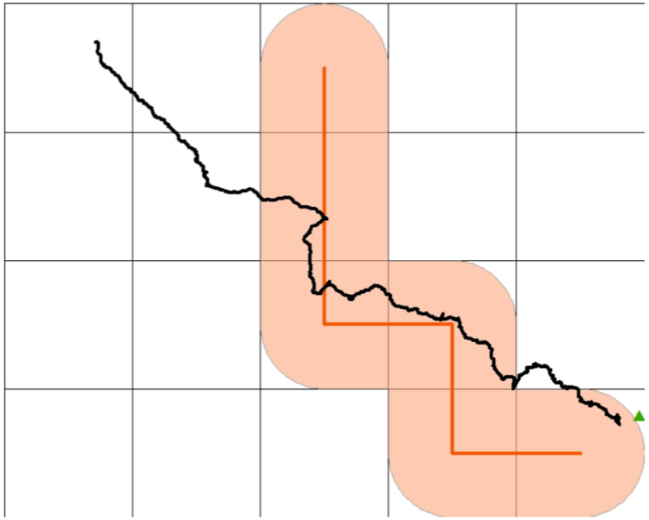
Figure 7: New integrated algorithm

PAWS: Real-World Results

- “PAWS patrols are now regularly deployed at a conservation area in Malaysia.”
- “The patrollers mostly followed PAWS’s suggested route, indicating that the route generated by PAWS is easy to follow.”
- “In addition, patrollers commented that PAWS is able to guide them towards poaching hotspots”

PAWS: Real-World Results

PAWS-Initial



(a) Deployed route

PAWS

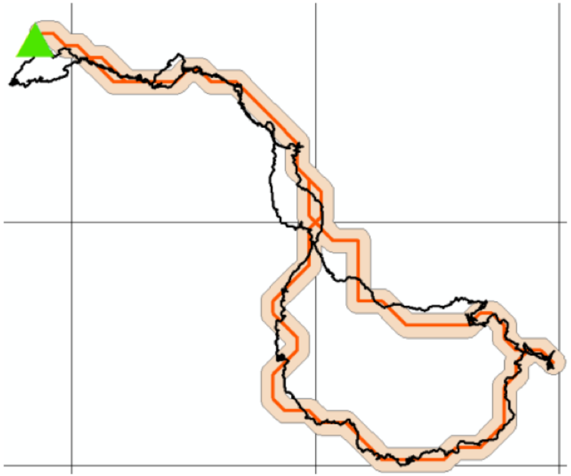


Figure 9: One daily PAWS Patrol route in Aug. 2015.

PAWS: Real-World Results

Patrol Type	All PAWS Patrol	Explorative PAWS Patrol	Previous Patrol for Tiger Survey
Total Distance (km)	130.11	20.1	624.75
Average # of Human Activity Signs per km	0.86	1.09	0.57
Average # of Animal Signs per km	0.41	0.44	0.18

Table 3: Summary of observations.

Conclusion

- A pure game theoretic approach was a good start.
- Real-world problems are often messy.
- Field experience was needed to make this tool useful.
- Now it is deployed and getting real results!
- Management of PAWS was turned over to ARMORWAY according to paper
    - Not sure if this actually happened