Team Orienteering with Fixed-Wing Drones

Kaarthik Sundar[1] **Sujeevraja Sanjeevi**[2]

[1]Staff Scientist
Information Systems and Modeling (A-1)
Los Alamos National Laboratory, NM, USA
`kaarthik@lanl.gov`

[2]Team Lead Operations Research
Sabre, TX, USA
`sujeev.sanjeevi@gmail.com`

Arizona State University, November 1 2019

Contents

Table of Contents

Motivation

Drone applications of interest:

- Package delivery, Healthcare
- Monitoring, Sensing, Mapping, Surveillance

Drone-routing problem complexity:

- Generic VRP issues (dense graphs, cycles)
- Fuel constraints
- Rigid body dynamics (if considering kinematic constraints)

Type of drones

Rotary drones:

- Lighter payloads
- Lower altitudes
- On-the-spot turns

Applications:

- Package delivery
- Healthcare

Figure: DHL's drone package copter used in package delivery

Types of drones

Fixed-Wing drones:

- Heavier payloads
- Higher altitudes
- Longer flights
- Resistance to wind
- Minimum turn radius

Applications:

- Surveillance
- Sensing
- Mapping

Figure: AeroTerrascan's drone Ai450 mapping a field in Indonesia

Table of Contents

Kinematics

Path requirements:

- Start from (x_i, y_i) at angle θ_i.
- End at (x_f, y_f) at angle θ_f.
- Travel at constant speed v.

Constraints:

- $\dot{x} = v \cos\theta$ (x component)
- $\dot{y} = v \sin\theta$ (y component)
- $|\dot{\theta}| \leqslant \alpha$ (yaw rate limit)

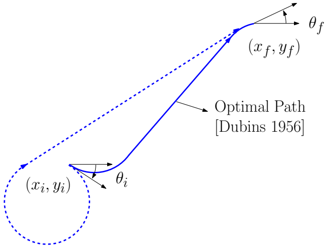


Figure: Dubins' path

Dubins Path

Shortest path segments [Dub57]:

- maximum curvature turns
- straight lines

Computing shortest path:

- Possible path sequences: RSR, RSL, LSR, LSL, RLR, LRL.
- Compute lengths of all sequences (easy), pick minimum!

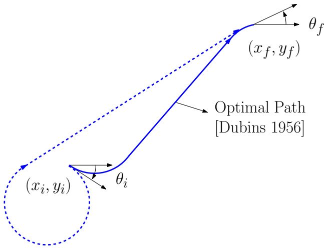


Figure: Candidate (dotted, RSR) and optimal (solid, LSR) Dubins paths

Drone Routing

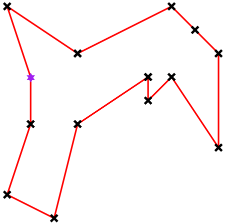


Figure: Euclidean TSP

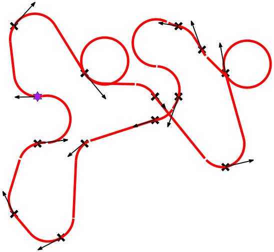


Figure: Dubins TSP with fixed heading angles

- Euclidean solution may not be Dubins feasible.
- Finding optimal heading angles for a sequence is NP-hard.

Table of Contents

Problem Statement

Orienteering Problem:

- Decision: find subset of targets to visit
- Objective: maximize visit reward
- Constraints: route length, costs, time windows
- Complexity: NP-hard (reduction from TSP)
- Team Orienteering: multiple vehicles

Team Orienteering with Fixed-Wing Drones

- Paths are Dubins paths.
- Only route length constraints.

Research Contributions

- First known exact algorithm for TOP with fixed-wing drones
- Interleaved DSSR - a novel path generation procedure
- Acceleration schemes for pricing problems
- Concurrent branch-and-bound with new branching scheme
- Highly successful computational results

Discretization

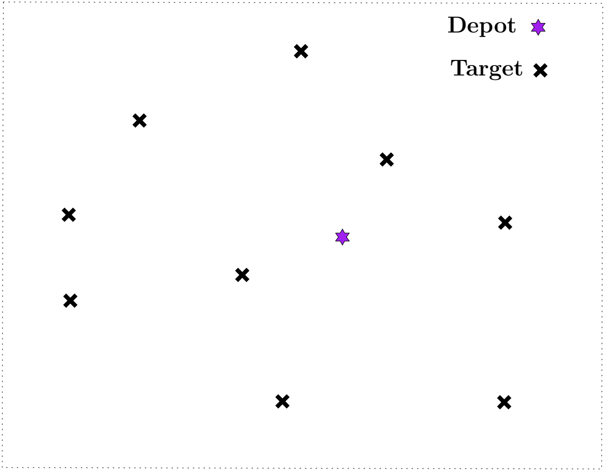


Figure: Target map

Discretization

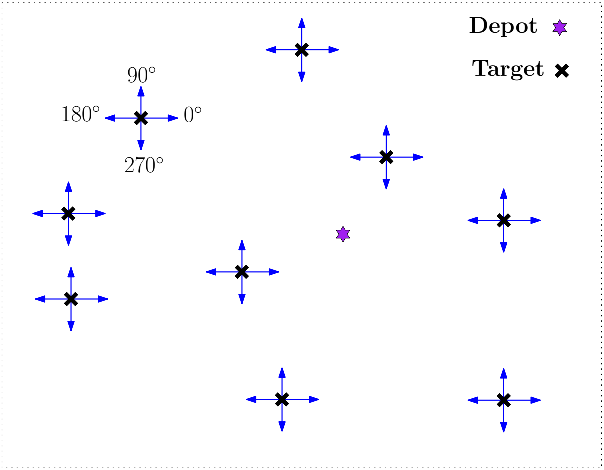


Figure: Discretized heading angles

Discretization

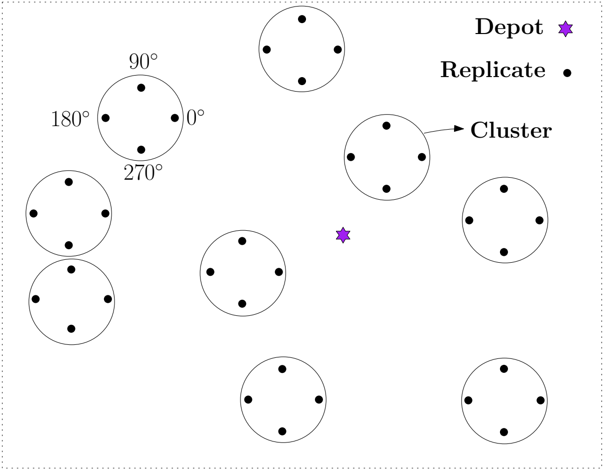


Figure: Graph with clusters

Discretization

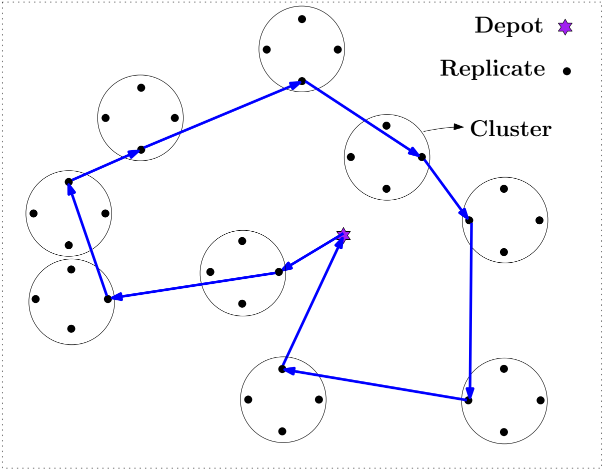


Figure: Dubins path on modified graph

Formulation

$$
\begin{array}{lll}
\textit{maximize} & p^T z & \\
\textit{subject to} & \mathbf{1}^T z \leqslant m, & \\
 & a_t^T z \leqslant 1, & t \in T, \\
 & z \in \{0,1\}^{|R|}. &
\end{array}
$$

- p: route scores
- m: number of vehicles
- a_t: route incidence for target t
- z_r: 1 if route $r \in R$ selected, 0 otherwise.

Table of Contents

Pricing

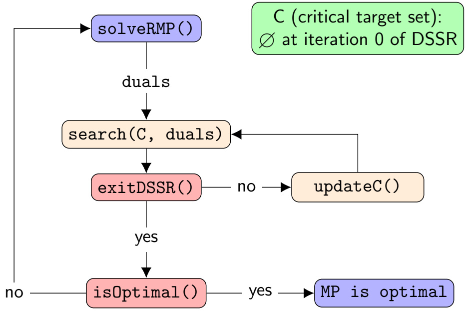


Figure: pricing with Decremental State Space Relaxation [RS08]

Pricing

Search:

- Extend all forward labels.
- Extend all backward labels.
- Join labels to build paths.

Limitations:

- How to select label?
- Early exits difficult.

Interleaved search

- Select label by bang-for-buck ($rc/length$).
- Switch between forward and backward labels (balancing).
- Join and extend each label (interleaving).
- Use early exits.

Early Exits

Conditions that worked for us:

- During search: stop if 500 elementary paths are found.
- After search: stop if optimal path is elementary [RS08].
- After search: stop if 10 elementary paths are found.
- Use weaker dominance checks in early iterations.

Dominance: a label l_1 dominates l_2 if

- Reduced cost, path length of l_1 are lower than l_2.
- Critical targets visits of l_1 are a subset of l_2 visits (computationally expensive, relax!).

Branching

Branching for Team Orienteering [BFG07]:

- If a vertex has fractional flow, branch on visit to it.
- Otherwise, find edge (v_1, v_2) with fractional flow.
- If v_1 or v_2 visit enforced, branch on edge visit.
- Otherwise, branch by
  - not visiting v_1,
  - visiting v_1 and (v_1, v_2),
  - visiting v_1 and not visiting (v_1, v_2).

Branching

Target Branching for Team Orienteering:

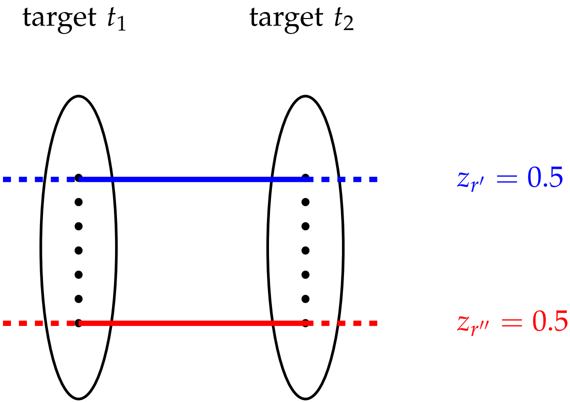
- If a target has fractional flow, branch on visit to it.
- Otherwise, find target connection (t_1, t_2) with fractional flow.
- If t_1 or t_2 visit enforced, branch on edge visit.
- Otherwise, branch by
  - not visiting t_1,
  - visiting t_1 and (t_1, t_2),
  - visiting t_1 and not visiting (t_1, t_2).

Branching Model

$$
\begin{array}{lll}
\textit{maximize} & p^T z - My & \\
\textit{subject to} & \mathbf{1}^T z \leqslant m, & \\
& a_t^T z \leqslant 1, & t \in \tilde{T}, \\
& a_t^T z \geqslant 1, & t \in ET, \\
& b_c^T z \geqslant 1, & c \in EC, \\
& y \geqslant 0, z \in \{0,1\}^{|R|}. &
\end{array}
$$

- M: large positive number
- $\tilde{T}$: visitable targets
- $ET \subseteq \tilde{T}$: enforced targets
- b_c: route incidence for enforced connection $c \in EC$

Branching Sufficiency



- Is this solution unique?

Branching Sufficiency

Can a LP solution have integer flows between all target pairs and a vertex-edge with fractional flow?

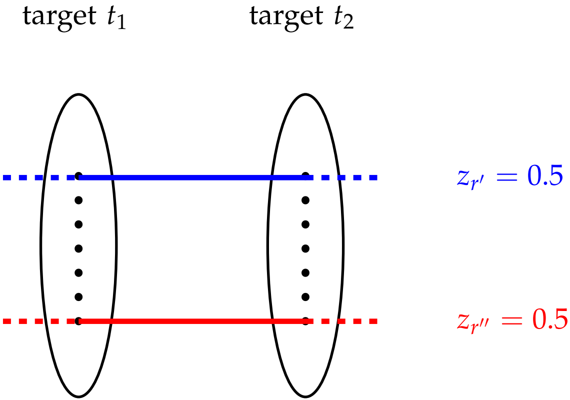
Theorem

If LP solution has

- *integer flow into every target,*
- *fractional flow into some vertex,*

an alternate optimal integer solution can be constructed.

Branching Sufficiency



- Is this solution unique?
- No! Alternate solution has $z_{r'} = 1$, $z_{r''} = 0$.

Table of Contents

Test Setup

Software:

- Implementation: Kotlin (JVM)
- Concurrency: Kotlin coroutines
- Solver: CPLEX 12.9
- Data: SQLite, Python

Hardware:

- HPE ProLiant XL170r servers
- Two Intel 2.10 GHz CPUs
- 128 GB of memory

Overall Stats

- Time limit: 1 hour
- Maximum concurrent solves: 8
- Number of data-sets: 267 [CGW96], [BFG07]
- Number of targets: 21, 32, 33, 64, 66
- Number of discretizations: 2, 3, 4
- Number of instances: 801
- Number of infeasible instances: 152
- Number solved to optimality: 502 out of 649 (77.34%)

Discretization Effects

Compare run-times and solutions of 2, 4, 6 discretizations.

Selection criteria:

- Number of targets: 66
- Solution time: > 1 second
- Solution is feasible
- The 6 discretization problem reaches optimality.

Discretization Effects

Table: Solutions and run-times for 66-target instances

	Disc 2		*Disc 4*		Disc 6	
Name	*Obj*	*Time*	*Obj*	*Time*	*Obj*	*Time*
5.2.g	270	2.30	300	27.32	320	415.29
5.2.h	300	5.44	330	103.82	340	1610.49
5.3.j	240	1.14	300	12.28	380	209.20
5.3.k	420	2.74	460	52.19	470	696.54
5.3.l	450	5.49	495	78.20	510	1846.28
5.4.m	320	1.21	380	10.39	420	133.14
5.4.n	450	2.35	570	23.64	590	468.53
5.4.o	600	4.73	620	54.86	660	1772.39
5.4.p	600	5.60	660	77.66	680	1778.87

Search and Concurrency

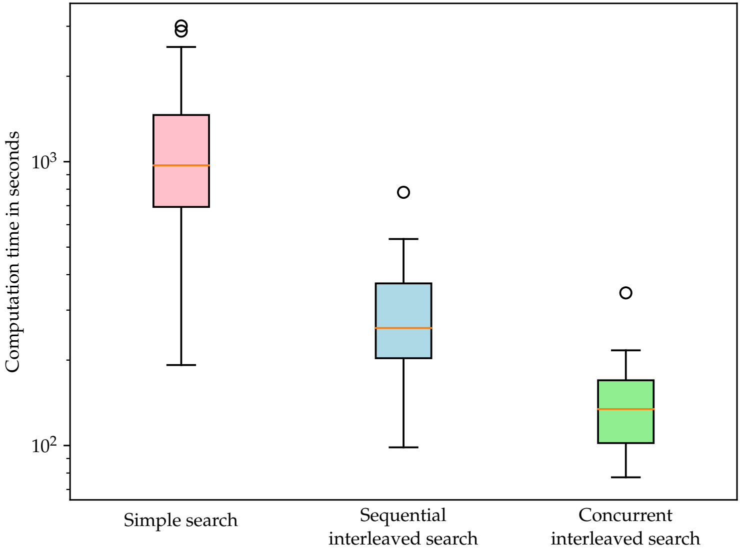
Compare run-times of :

- Sequential simple search
- Sequential interleaved search
- Concurrent interleaved search

Selection criteria:

- Number of targets: 32, 33
- Solution time: $>$ 1 minute
- Optimality reached with simple search.
- Branching occurs.

Search and Concurrency



Search and Concurrency

Table: Percentage improvements over simple search

Name	*Sequential*	*Concurrent*
t32_d4_1.2.k	84	90.96
t32_d4_1.3.m	79.3	95.33
t32_d4_1.3.p	80.29	93.91
t32_d4_1.4.q	57.64	88.96
t32_d4_1.4.r	64.96	92.69
t32_d6_1.3.l	75.93	92.41
t33_d2_3.2.k	68.42	95.05
t33_d2_3.2.l	81.41	88.74
t33_d2_3.2.m	77.12	82.09
t33_d2_3.3.q	70.12	92.55
t33_d4_3.2.g	83.44	93.77
t33_d4_3.3.k	78.94	96.89
t33_d4_3.4.n	57.06	90.54
t33_d4_3.4.p	85.26	97.17
t33_d4_3.4.q	72.96	97.47
t33_d6_3.4.k	37.99	80.23

Acceleration Schemes

Compare run-times of sequential interleaved search with:

- Strict label dominance (lesser label pruning, iterations)
- Relaxed label dominance (more label pruning, iterations)

Selection criteria:

- Number of targets: 32, 33
- Solution time: > 1 minute
- Optimality reached with strict dominance condition
- Branching occurs

Acceleration Schemes

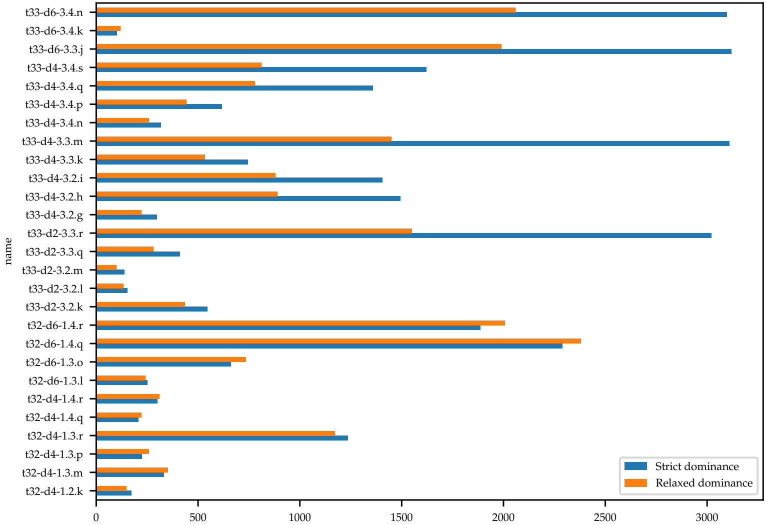


Table of Contents

Summary

- Formulated TOP for fixed-wing drones
- Proposed an exact algorithm to find Dubins paths with branch-and-price
- Acceleration schemes provide significant performance boost over existing works in literature

Next Steps

Future research on TOP with fixed-wing drones:

- Smarter discretization
- Concurrent interleaving
- Improve bounds with cuts, local search
- Variants (time windows, multiple depots)

Next Steps

My research pipeline:

- Flight scheduling with uncertain delays
- Submarine resurfacing policies with uncertain localization errors
- Vacation planning (multi-city, multi-modal transport, etc)
- Coordinated drone routing aka timing constraints
- Decomposition techniques for large graphs

References

Sylvain Boussier, Dominique Feillet, and Michel Gendreau. "An exact algorithm for team orienteering problems". In: *4or* 5.3 (2007), pp. 211–230.

I-Ming Chao, Bruce L Golden, and Edward A Wasil. "The team orienteering problem". In: *European journal of operational research* 88.3 (1996), pp. 464–474.

Lester E Dubins. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". In: *American Journal of mathematics* 79.3 (1957), pp. 497–516.

Giovanni Righini and Matteo Salani. "New dynamic programming algorithms for the resource constrained elementary shortest path problem". In: *Networks: An International Journal* 51.3 (2008), pp. 155–170.

Slides will be available at `https://sujeevraja.github.io`.

The paper will be up in arxiv in a few weeks.