グラフ探索理論に基づいた軌道最適化について

A Shortest-Path Graph Search-Based Trajectory Optimiser for Wind-Optimal Trajectories

ブラウン マーク、平林 博子、ビクラマシンハ ナヴィンダ キトマル、虎谷 大地
航空交通管理領域

Mark BROWN, Hiroko HIRABAYSHI, Navinda Kithmal WICKRAMASINGHE and Daichi TORATANI
Air Traffic Management Department

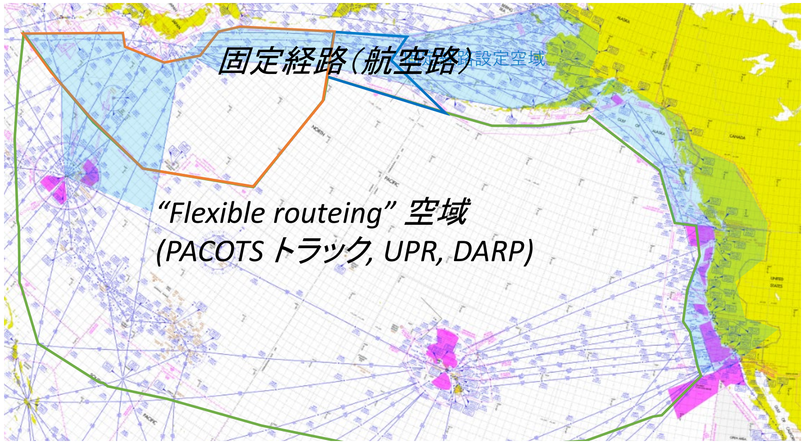
はじめに Background & Motivation

研究課題 Research Problem

ジェット気流から影響を受けるアジア～北米間の便の最適経路設計。

Optimising trans-Pacific flights between Asia and North America, which are affected strongly by the Polar Jet Stream.

> *洋上空域設計を評価する場合、運航者はどの経路を選定するかを予測する必要がある。*
> *When evaluating different possible airspace/route structures, we wish to know what routes operators are likely to choose.*



最適軌道生成の必要

- 軌道ベース運用(TBO)は飛行計画ルートの柔軟性を高める
- UPR及びDARPは燃費効率の良い風最適軌道を可能とするが、風の影響を考慮する軌道生成能力を備えていることが必要。
- *Trajectory-Based Operations (TBO) promises more flexibility to flight plan routes*
- *UPR (User-Preferred Route) and DARP (Dynamic Airborne Reroute Procedure) offer possibilities to plan more fuel-efficient wind-optimal trajectories. Requires a wind-optimal trajectory generation capability.*

> *実用的な最適軌道生成方法を求める*
> - *合理的な計算時間*
> - *操作上の制約を許可（ルート、高度、速度、時間など）*
>
> *Need a practical wind-optimal trajectory generator:*
> *・Reasonable computational time*
> *・Allows operational constraints (route, altitude, speed, time etc.)*

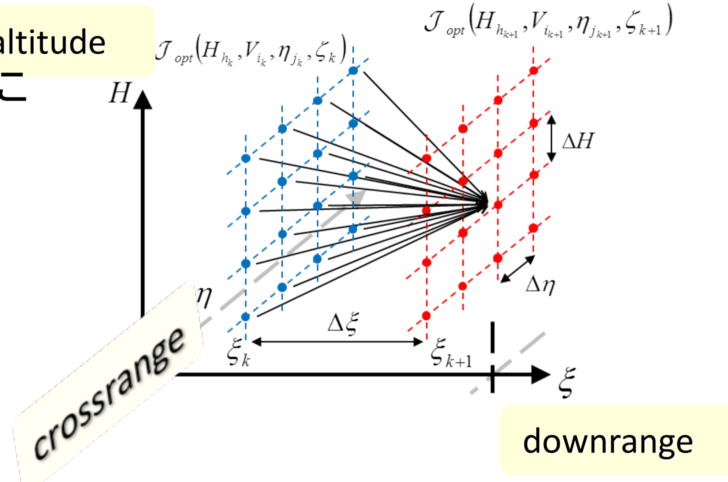
- 飛行計画サービスの提供者は存在するが、専有/商業的な秘密になる傾向があり、最適化方法は公開されていない

Commercial flight planning service providers exist, but their optimisation methods tend to be held secret and are not published.

理想な最適軌道生成 *Ideal Trajectory Generation*

動的計画法 *Dynamic Programming approach*

- 出発地から目的地までの経路の周りの離散的な点を取り除くこと　によって多次元探索空間を分割
  Divide multidimensional search space by "gridding" discrete points around a path from origin to destination.
  Variables: Downrange distance, Cross-range distance, Altitude, Speed
  Initial conditions: mass, altitude, speed, start & end position, initial reference trajectory based on Great Circle.



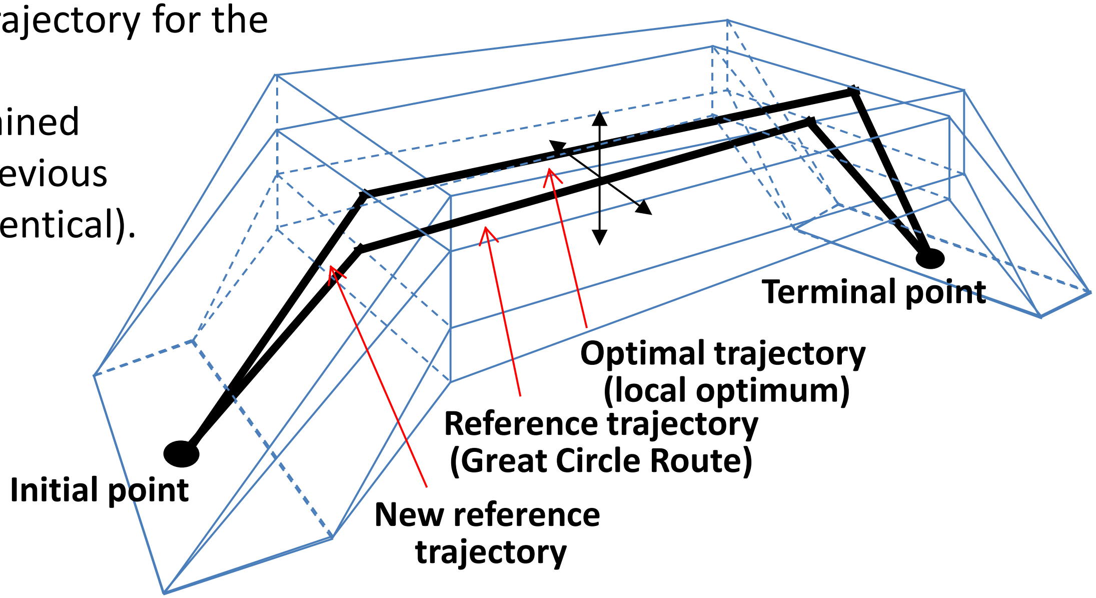
Combinatorial optimization logic in DP method
(*Speed (CAS) axis is omitted*)

Calculation process

- A partial search space is set around the current reference trajectory and a local optimal trajectory is generated.
- If the obtained trajectory is different from the current trajectory, it becomes the new current trajectory for the next iteration of calculations.
- The method repeats until the newly obtained trajectory identically converges to the previous optimal trajectory (fuel burn becomes identical).

欠点 *Drawbacks*

- 多量計算が必要
  High computational cost.
- 軌道制限の適用が困難.
  Hard to apply constraints.



実用的な軌道最適化要件 *Practical Requirements*

合理的な計算時間とリソース *Reasonable computation time & resources*

- 大域的最適解を探すために長時間を費やすことは実用的ではない
  天気予報などの不確定性によって状況がどうしても変化してしまう

  Spending long time searching for global optimum impractical and may be wasteful – weather forecast uncertainty means conditions will change anyway.

運用上の制約を適用する方法 *Ways to apply operational constraints*

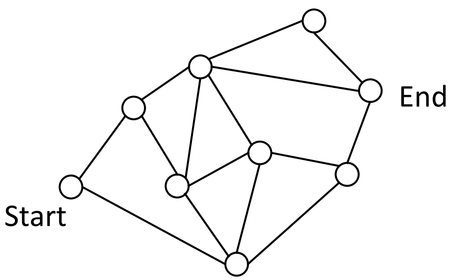
- 横方向の軌道 *Lateral trajectory*
  - 軌道の一部に沿ってATS経路をたどる *follow ATS routes along parts of the trajectory*
  - 特定のウェイポイントを通過 *pass certain waypoints*
  - 悪天候/火山灰/制限空域を避ける *avoid bad weather/volcanic ash/restricted airspaces*

  etc.
- 垂直軌道 *Vertical trajectory*
  - 高度制限（最大高度、最低高度） *level constraints*
  - 利用可能な飛行レベルを使用 *use available flight levels rather than arbitrary altitudes*
  - 連続上昇ではなくステップ上昇 *step climb rather than continuous climb*
- 速度 *Speed*
  - 管制間隔を保つため *for separation*
  - 時間ベースの制御 *time-based control*

グラフ最短経路探索を適用できるか?
Can we apply shortest-path graph search?

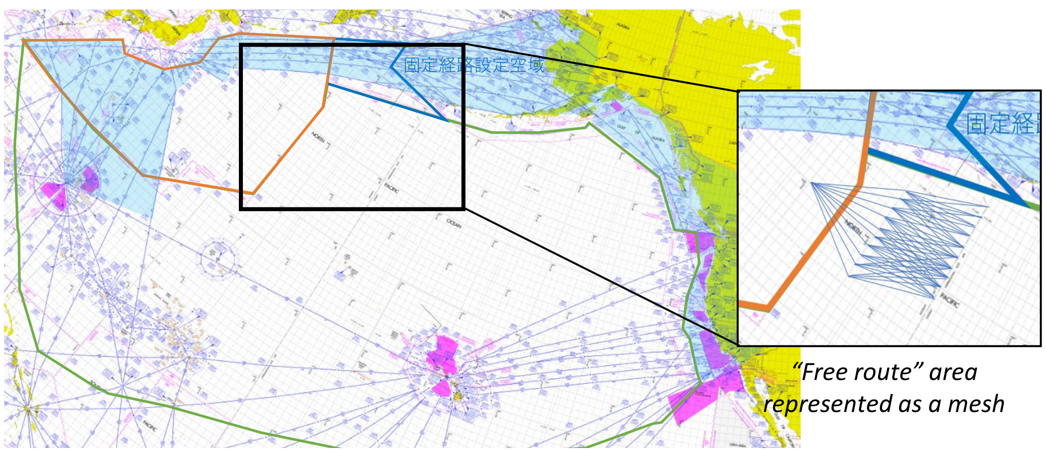
グラフ探索最適化の概念　*Graph Search Optimisation*

General graph shortest-path search:
Graph composed of vertices (nodes) and edges.
Find minimum distance path between two vertices.
(Dijkstra Algorithm, A Algorithm etc.)*

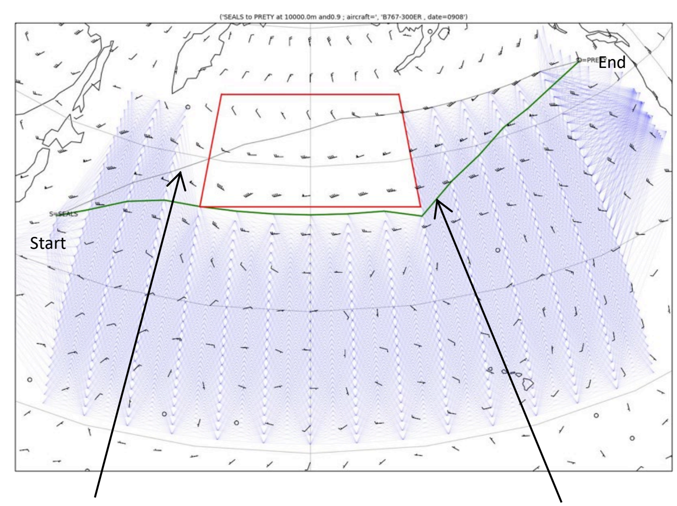


Concept

- Possible lateral flight paths (including ATS routes and free routeing areas) represented as a spatial graph.
- Use fuel consumption or flight time between vertices as the metric instead of Euclidian distance.
  → Graph search yields <u>minimum fuel path</u> or <u>minimum time path</u>.
- Set constraints on graph vertex or edge availability to model
  - Restricted air spaces (with opening/closing times)
  - Volcanic ash/weather etc. (can move)
  - One way or bidirectional routes



"Free route" area represented as a mesh



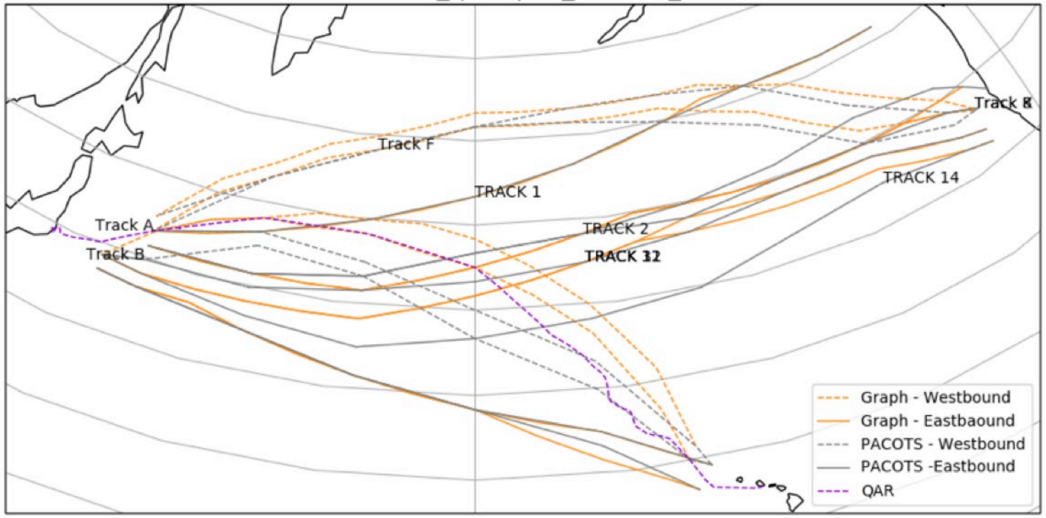
Minimum fuel path without airspace restriction

Minimum fuel path with airspace restriction (edges in red polygon removed from graph)

Compare with PACOTS tracks and actual DARP track

(Restrictions: Only single cruise TAS and altitude set, only a single weather forecast.)

file name = rfl_tpf-tcpv3_170113_b773.txt



Thanks to All Nippon Airways for providing QAR data.

- Graph search-computed minimum time tracks show reasonable correlation with published PACOTS tracks.
- Honolulu to Tokyo flight (QAR) Initially follows published PACOTS track then executes DARP based on more recent weather forecast. Flown track west of 180E coincides with minimum time track computed using graph search.

- Computation time: < 10 seconds (Python/numpy).

まとめと今後の課題 *Conclusion and Future Work*

- 実用的な飛行軌道の、DPベースの制約のない風最適軌道生成機能の欠点を克服するために提案されたグラフ最短経路探索に基づく軌道最適化

  Trajectory optimisation based on graph shortest-path search proposed to overcome drawbacks of DP-based unconstrained wind-optimal trajectory generator for practical flight trajectories.

- 有望な結果を示している概念実証検証

  Proof-of-concept validation shows promising results.

- 今後　Need to extend to:
  - 巡航高度・速度の最適化　*Optimise cruise/speed altitude (step climb)*
  - 運用コストの最適化　*Optimise operating cost (fuel/time, route charges)*
  - ウェイポイントの時間制約を設定　*Waypoint time constraints*

ご静聴ありがとうございました

Thank you for your attention.

Any questions?