

# Tomo-e Gozen サーベイ観測の 最適化アルゴリズムとシステム開発

東京大学修士一年 津々木 里咲

酒向 重行, 瀧田 怜, 紅山 仁, 高橋 英則, 近藤 荘平, 森 由貴 (東京大学),  
大澤 亮 (国立天文台), 諸隈 智貴 (千葉工業大学),  
池田 思朗 (統計数理研究所), João Pedro Pedroso (ポルト大学)

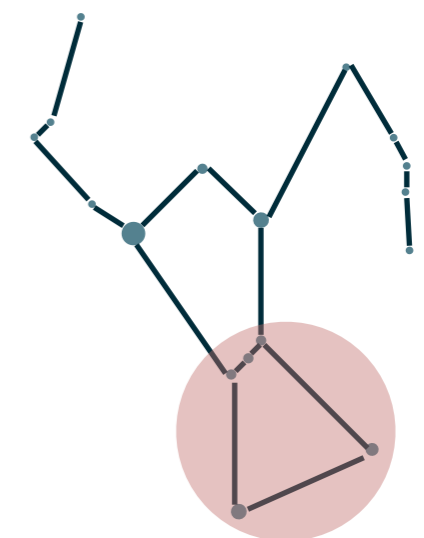
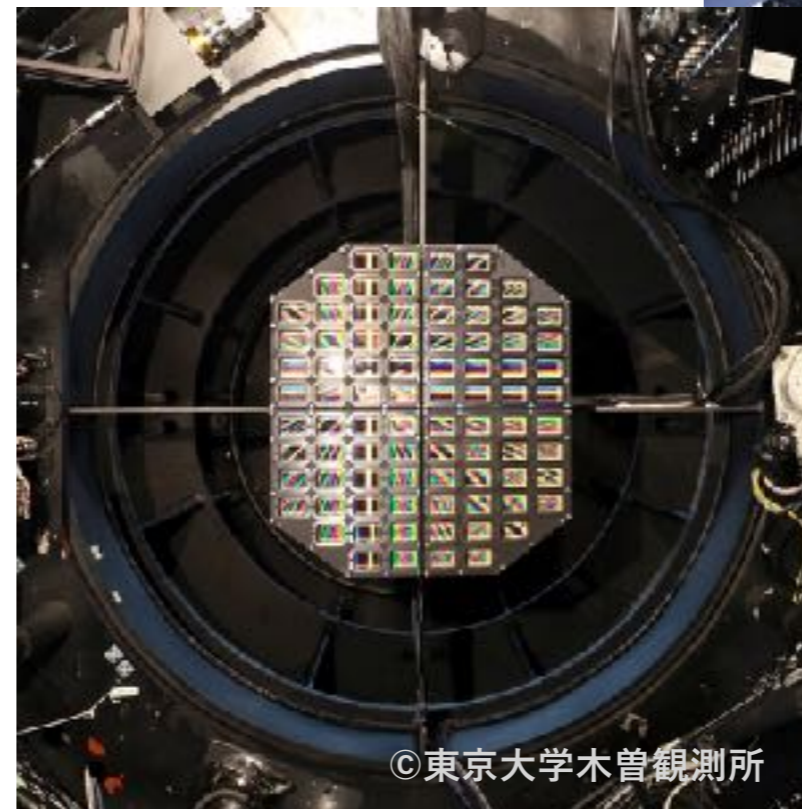
# Tomo-e Gozen



- 長野県木曾郡 東京大学木曾観測所  
口径1.05mシュミット望遠鏡用広視野可視光カメラ
- 84枚の表面照射型CMOSセンサーにより  
**20平方度(直径9度)の視野領域を**  
**0.5秒の時間分解能で動画観測が可能**
- **時間軸天文学**を開拓  
高速移動天体, 短時間突発現象,  
マルチメッセンジャー天文学



Tomo-e Gozenカメラと  
1.05mシュミット望遠鏡

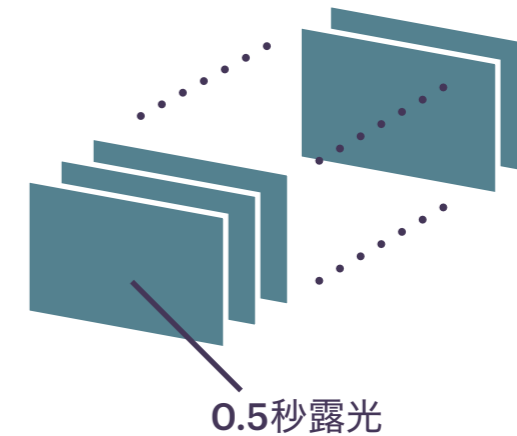


Tomo-e Gozenの視野領域

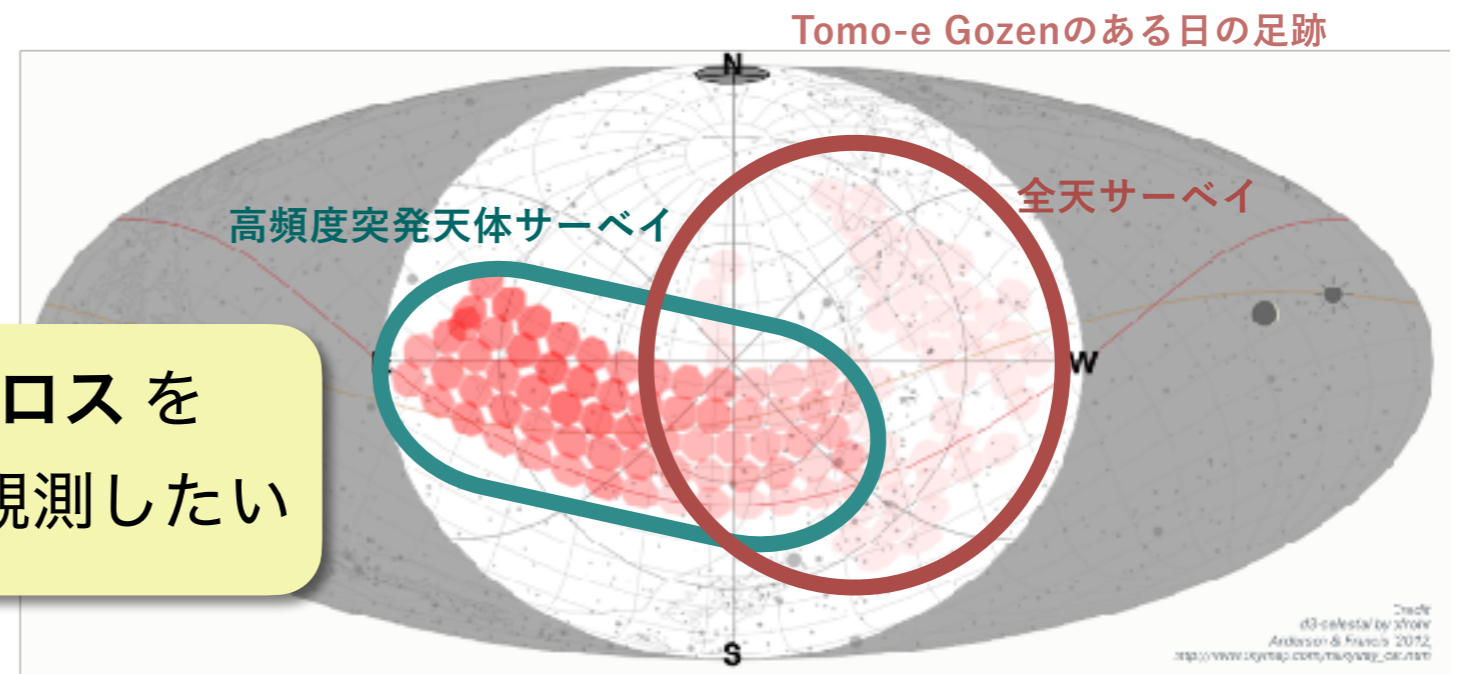
# 毎晩の広域サーベイ観測

短時間突発現象や高速移動天体など確率的に発生するイベントを捉える

- 日没後の2-3時間で**全天サーベイ**を実施
  - ・ 1ポイントあたり**0.5秒露光**×**18フレーム**
  - ・ 一度観測した所は行かずに**なるべく広範囲**を観測
- 残りの時間で**高頻度突発天体サーベイ**を実施
  - ・ 1ポイントあたり**0.5秒露光**×**12フレーム**
  - ・  $-3^\circ \leq \text{Dec} \leq 35^\circ$  ,  
(深夜に南中するRA)から $\pm 70^\circ$   
の範囲内で**満遍なく行き来**する



広域内を **望遠鏡移動によるタイムロス** を  
できるだけ短くした経路で効率良く観測したい

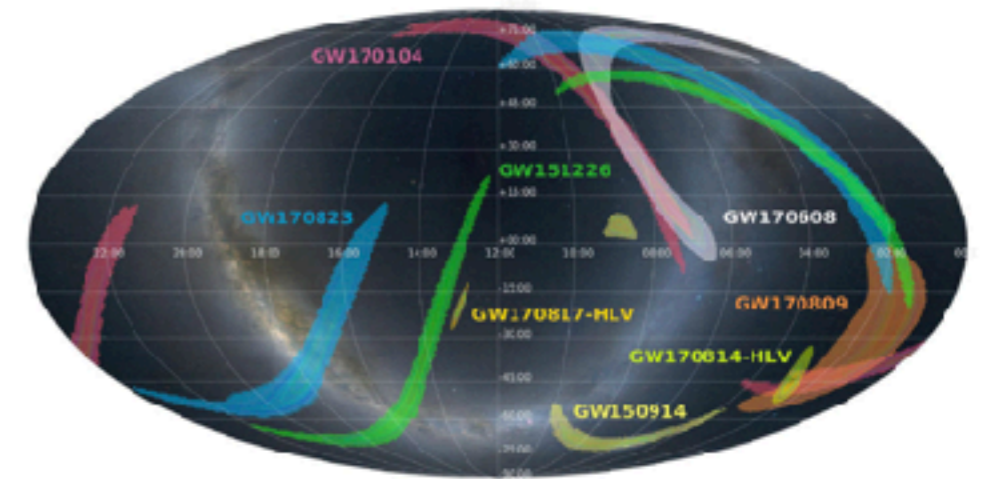


# 重力波 可視光追観測

広域内を効率よく観測していくことは迅速な追観測においても有用

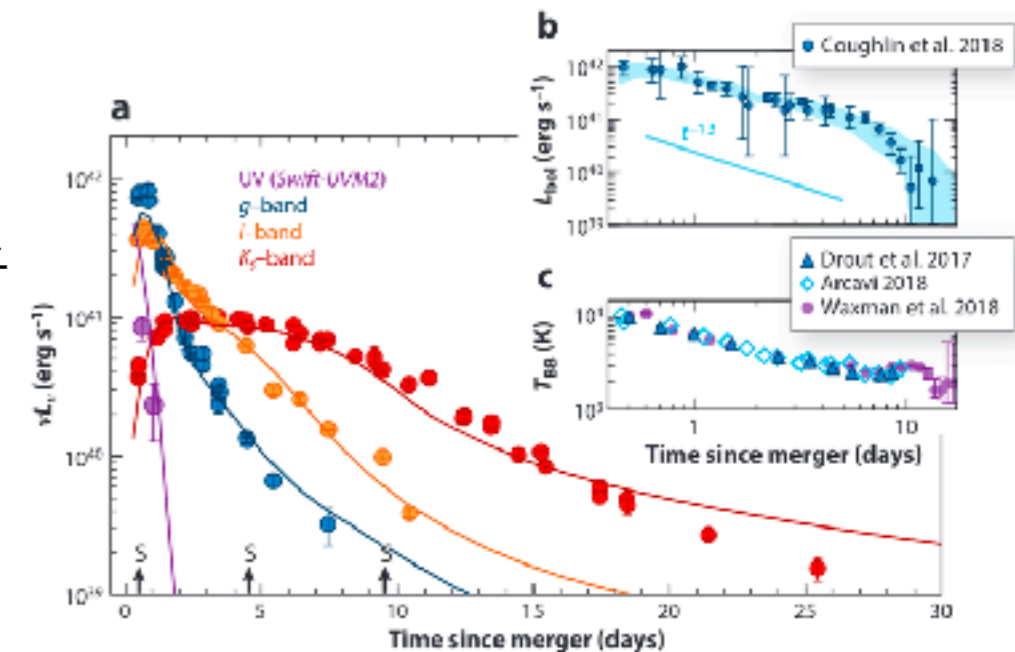
- 重力波検出器ネットワーク(LIGO, Virgo, KAGRA)により、重力波検出後数十秒で波源の**位置情報**(決定精度は**数百平方度**)が報告される
- Tomo-e Gozenではアラートを受けると候補天体特定のための可視光追観測を**優先的に実施**
- 可視光観測では連星中性子星またはブラックホール-中性子星連星の合体から**数日間にキロノバ**がみられる  
GW170817 ... 合体から11時間後に電磁波対応天体を発見  
→ 合体から**11時間以内**の機構が未解明
- 2023年 第4期重力波観測(O4)開始予定**

01,02で報告された重力波の位置情報



Abbott, Benjamin P., et al. "Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA." *Living reviews in relativity* 23.1 (2020): 1-69.

GW170817の対応天体で観測されたキロノバの光度曲線



Margutti, Raffaella, and Ryan Chornock. "First multimessenger observations of a neutron star merger." *Annual Review of Astronomy and Astrophysics* 59 (2021): 155-202.

---

# サーベイ経路の最適化アルゴリズム

João Pedro Pedroso (ポルト大学)

池田 思朗 (統計数理研究所)

諸隈 智貴 (千葉工業大学)

# サーベイ経路の最適化を数理的に解く

- 一般的に数理最適化問題は、決定変数・目的関数・制約条件を数式化して数理最適化 solver に解かせる
- それに倣いサーベイを数式化して solver (gurobi) に解かせると、2,3個のポジションに限定しても12時間では計算が終わらない → 厳密解を得るのは難しい

## 変数

- ▶  $K$  → set of positions to be observed in the sky
- ▶  $I$  → set of positions in the telescope
- ▶  $T$  → number of periods to consider (time discretization)
- ▶  $a_{ikt}$  → connect telescope and sky's positions:
  - ▶  $a_{ikt} = 1$  if at period  $t$  telescope in position  $i \in I$  observes sky's position  $k \in K$
  - ▶  $a_{ikt} = 0$  otherwise
- ▶  $c_{ij}$  → time necessary to move the telescope from position  $i$  to  $j$
- ▶  $d_k$  → time necessary to make observation at sky's position  $k$
- ▶ **Main decision variables:**
  - ▶  $x_{it} = 1$  if telescope is on position  $i$  at period  $t$
  - ▶  $x_{it} = 0$  otherwise
- ▶ **Telescope movement:**
  - ▶  $w_{ijt} = 1$  if at period  $t$  telescope moves from position  $i$  to position  $j$  (possibly,  $j = i$ )
- ▶ **Observed:** (determined in terms of  $x$ )
  - ▶  $y_{kt} = 1$  if sky's position  $k$  is observed at period  $t$ , 0 otherwise
- ▶ **Positions observed:** (determined in terms of  $y$ )
  - ▶  $z_k = 1$  if sky's position  $k$  has been observed

## 目的

$$\text{maximize } \sum_{k \in K} z_k$$

なるべく多くの領域を観測  
= 移動によるタイムロスを削減

## 制約

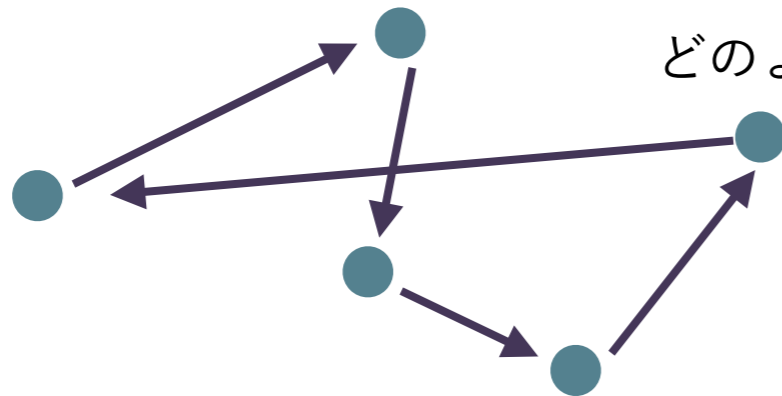
$$\begin{aligned} \text{subject to } & \sum_{i \in I} x_{it} \leq 1 && \text{for } t = 0, \dots, T \\ & x_{i,t-1} = \sum_{j \in I} w_{ijt} && \forall i \in I, t = 1, \dots, T \\ & x_{jt} = \sum_{i \in I: t - c_{ij} > 0} w_{ij, t - c_{ij}} && \forall j \in I, t = 1, \dots, T \\ & y_{k0} = 0 && \forall k \in K \\ & y_{kt} \leq \sum_{i \in I} a_{ikt} x_{it} && \forall k \in K, t = 1, \dots, T \\ & \sum_{t'=t}^{\min(T, t-d_k)} y_{kt'} \geq d_k (y_{kt} - y_{k,t-1}) && \forall k \in K, t = 1, \dots, T \\ & z_k \leq \sum_{t=1}^T y_{kt} && \forall k \in K \\ & \text{(all variables are binary)} \end{aligned}$$

# 巡回セールスマン問題

一般的な数理最適化問題を参考にする：巡回セールスマン問題

(Traveling Salesman Problem)

セールスマンが全ての都市を一筆書きで訪問するとき、  
どのような順番で都市を回るのが最短経路か



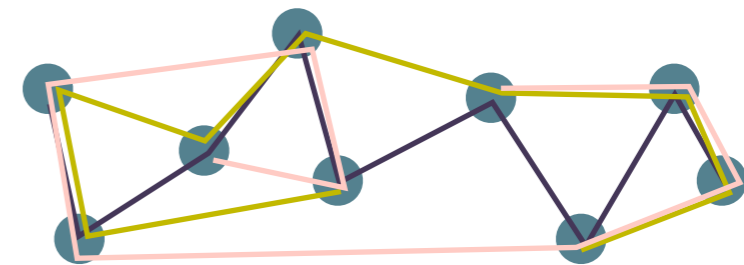
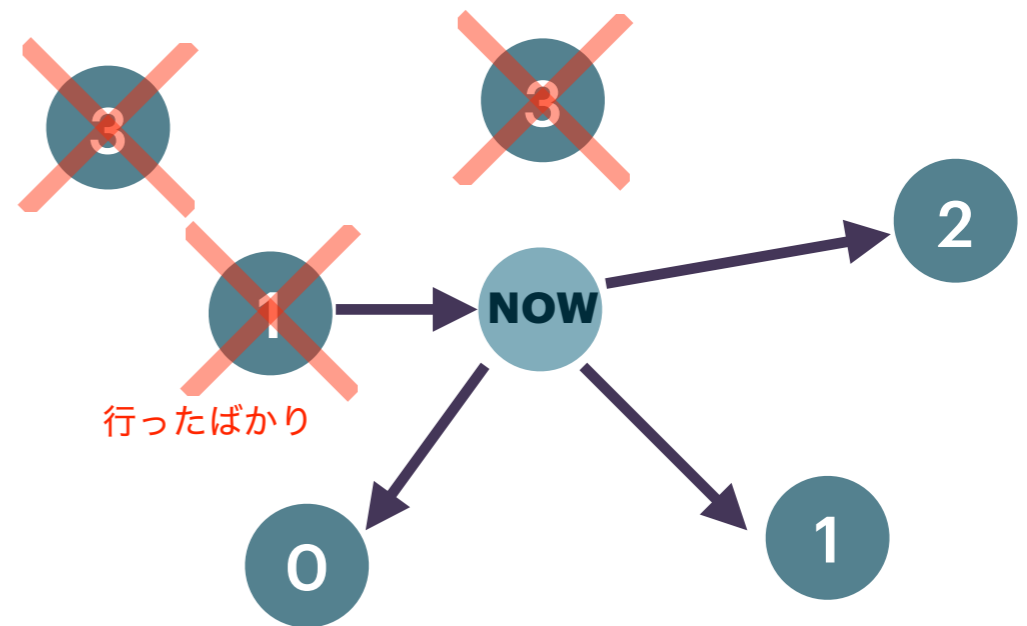
N個の都市に対して経路は  $(N-1)!/2$ 通り (閉路)

開路、閉路、など問題設定は様々

- 全経路を計算して最適解を探すと**計算量が膨大**になる ( $O(n!)$ 、NP困難)
- サーベイ経路の最適化は、動き続ける空を1定時間空けて複数回見るため**TSPよりも複雑**になる → **ヒューリスティックな解法が必要**
- TSPのヒューリスティックな解法のひとつ：**最近傍法**
  - 現在の頂点から最も近い頂点を探す
  - 見つけた頂点を現在の頂点とし、訪れたという印をつけておく
  - 現在の頂点から最も近い頂点を探す → … (繰り返し) …

# Tomo-e Gozen独自の最適化アルゴリズムの開発

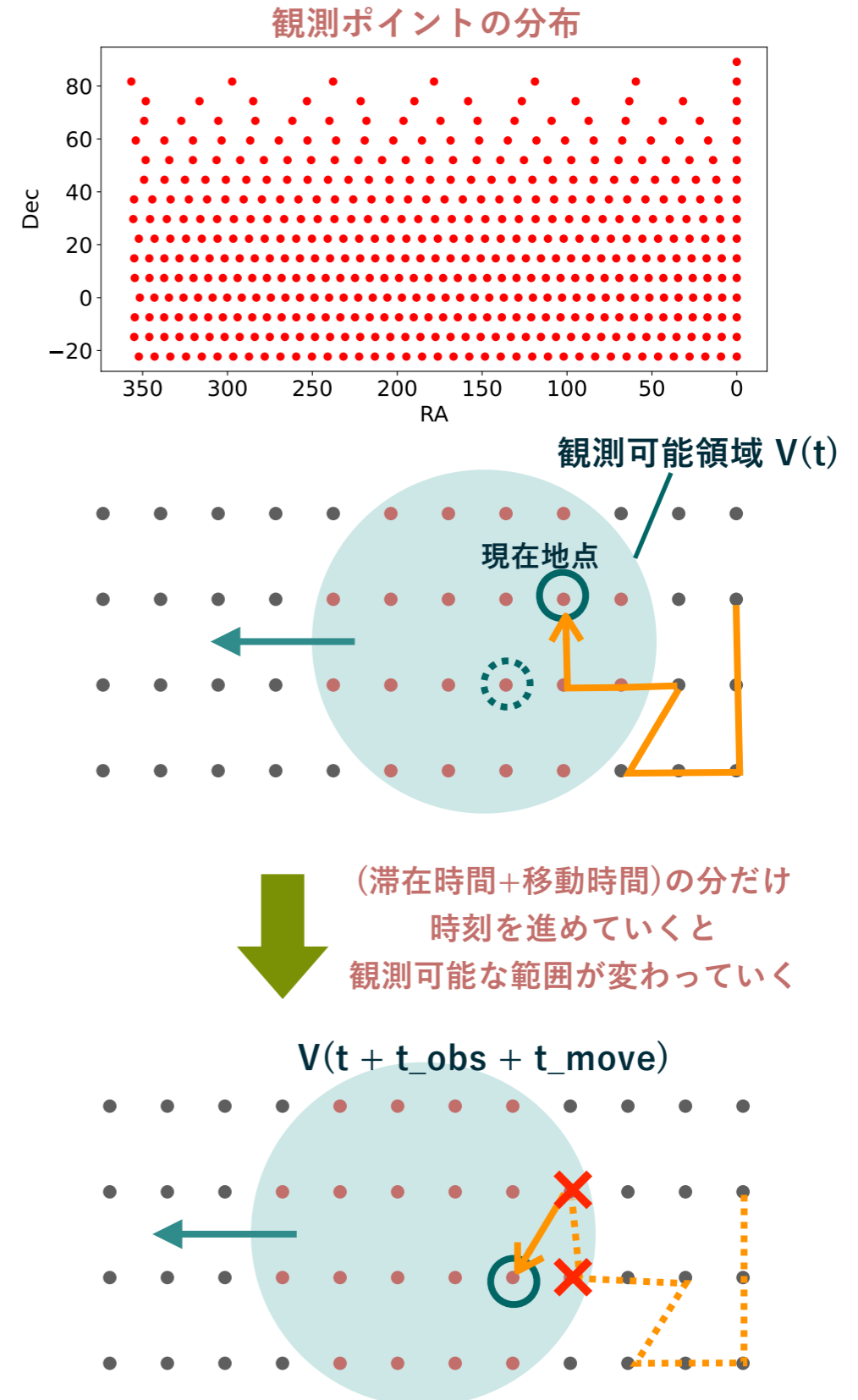
- ・ ”一晩で各領域を3回観測する領域数を最大化したい”という高頻度突発天体サーベイ用にアルゴリズムが開発された
- ・ 最近傍法に「**現在までの各領域の観測回数の最小値をNとしたとき、多くて(N+2)回観測した領域の中から選ぶ**」という考えを追加
- ・ 同じ領域を再び観測する場合は**ある一定以上の間隔**をあけるという条件を設定
- ・ 許容計算時間を設定し、時間内に考えた数万通りの経路の中から**最も結果が良かったもの(=3回以上観測した領域の数が最も多かったもの)**を採用する
- ・ より多くの解を検討するために、次に進むポイントを第二候補まで選定しランダム性を持たせる





# 日周運動の反映

- RA, Decを刻んで観測ポイントを設定
- 観測予定時刻に合わせて観測を計画していく
  - 各ポイントに滞在する時間は~1分  
(ディザリングを含む)
  - 各ポイント間の望遠鏡(赤道儀)移動時間は  
 $\max(\Delta ra, \Delta dec) \times \text{望遠鏡移動速度}[\text{deg/s}]$
- 各時刻において観測可能な(elevation  $\geq 30^\circ$ 、  
十分なインターバル確保済み)ポイントを求め、  
その中から次に進むポイントを決める
- 既に観測した場所を観測する際に最低限あける間隔は  
**高頻度サーベイ** : ~25分  
**全天サーベイ** : 全天サーベイにかける時間  $\times 2$   
( $\rightarrow$  同じ領域を2回以上観測することはできない)



---

# 最適化アルゴリズムを用いた 動的システムの開発

# 前提：観測レシピによる自動観測システム

ポインティング位置, 露光時間, 優先度を指定した”観測レシピ”を1ポインティングあたり1つ作成  
→ 観測レシピを上から順に実行していくことで観測計画に基づく自動観測を実現

観測レシピの例

```
obs.telescheduler > recipeimp > E 0294_2022-11-23T05:00:00.000.recipe
1 Observer: T.Morokuma
2 Project: High-Cadence Survey
3 Priority: 0.9588
4 Time Window: [ '15:00:00.0', '20:38:00.0' ]
5 Operations:
6   - SetFocus: 28.13
7   - Assert: domesLit_open
8   - Pointings: { ra: 77.8438975, dec: 7.4276782 }
9   - MirrorCover: open
10  - Wait: pointing, mirrorcover_open
11  - SetPipeline: [ wcs, stack, neo ]
12  - SetParameters: { gain: high, tinteg_sec: 0.5, nframes: 12 }
13  - Exposures: J0511+0725_dith1
14  - Dithering: [ 0, 1440 ]
15  - Wait: pointing
16  - Exposure: J0511+0725_dith2
17  - Dithering: [ -1980, 0 ]
18  - Wait: pointing
19  - Exposure: J0511+0725_dith3
20  - Dithering: [ 0, -1440 ]
21  - Wait: pointing
22  - Exposure: J0511+0725_dith4
23 Comment:
24 High-Cadence Survey, 284, 2022-11-23T05:00:00.000
25
```

Current Schedule (disabled / activated) ▶ II

15:00 18:00 21:00 24:00 03:00 06:00 09:00

Calibration: 23:53/ Start Observation: 23:53/ End Observation: 23:53

Scheduled Observations (total: 439)

elevation >= 30°になると緑色になる

Scheduler(今晚観測予定)

Queue(今から実行)

上から順に実行

緑の中から優先度の高い順に5個Queueに登録

Executing(現在実行中)

History

システムモニタ

The screenshot shows a software interface for telescope scheduling. At the top, there's a 'Current Schedule' section with a timeline from 15:00 to 09:00. Below it is a 'Scheduled Observations' section showing a grid of observation slots, some of which are green. A yellow box explains that slots become green when elevation is greater than or equal to 30 degrees. Below the grid is a 'Queue' section containing a list of recipes, with a yellow box indicating that the top 5 recipes from the green slots are added to the queue. To the right, there's an 'Executing' section showing a detailed view of a recipe being performed, and a 'History' section showing a list of past observations. A red arrow points from the queue to the executing section, with a yellow box stating '上から順に実行' (execute from top to bottom). The entire interface is labeled 'システムモニタ' (System Monitor).

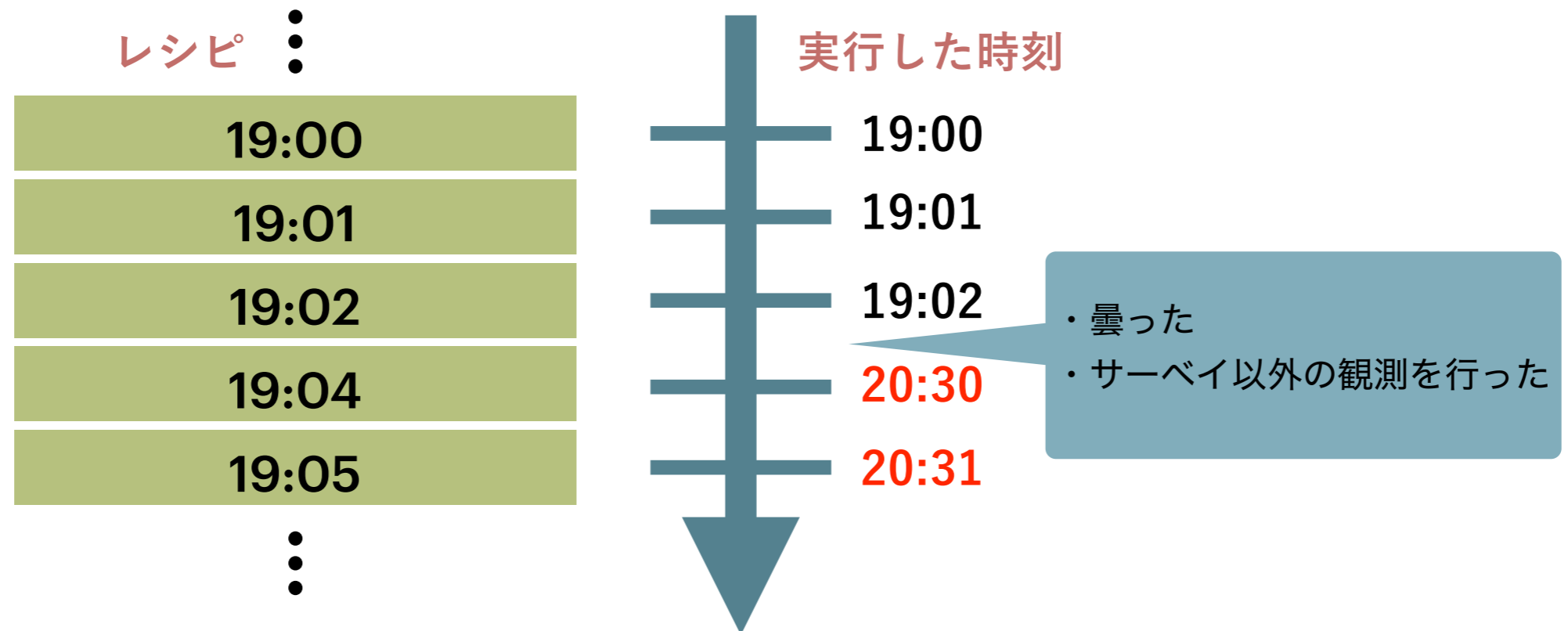
- 太陽高度により自動で観測を開始・終了
- 天候条件を常時自動監視し、悪天候になると実行が停止、再び晴れると自動で再開

# 既存の最適化システムの問題点

既存システム… 日没前に最適化計算をしてサーベイ開始から終了までの経路を一つ決定  
→ 決めた経路に沿って観測レシピを一通り(5-600個)作成し登録しておく  
→ 観測が始まるとそれらを上から順に実行していく

= 一晩中晴れ続けて開始から終了までサーベイが途切れることがないと想定

➔ 観測時のリアルタイムな空の情報が反映できない,  
天候や他の観測イベントによりサーベイが中断されると観測計画がずれていく

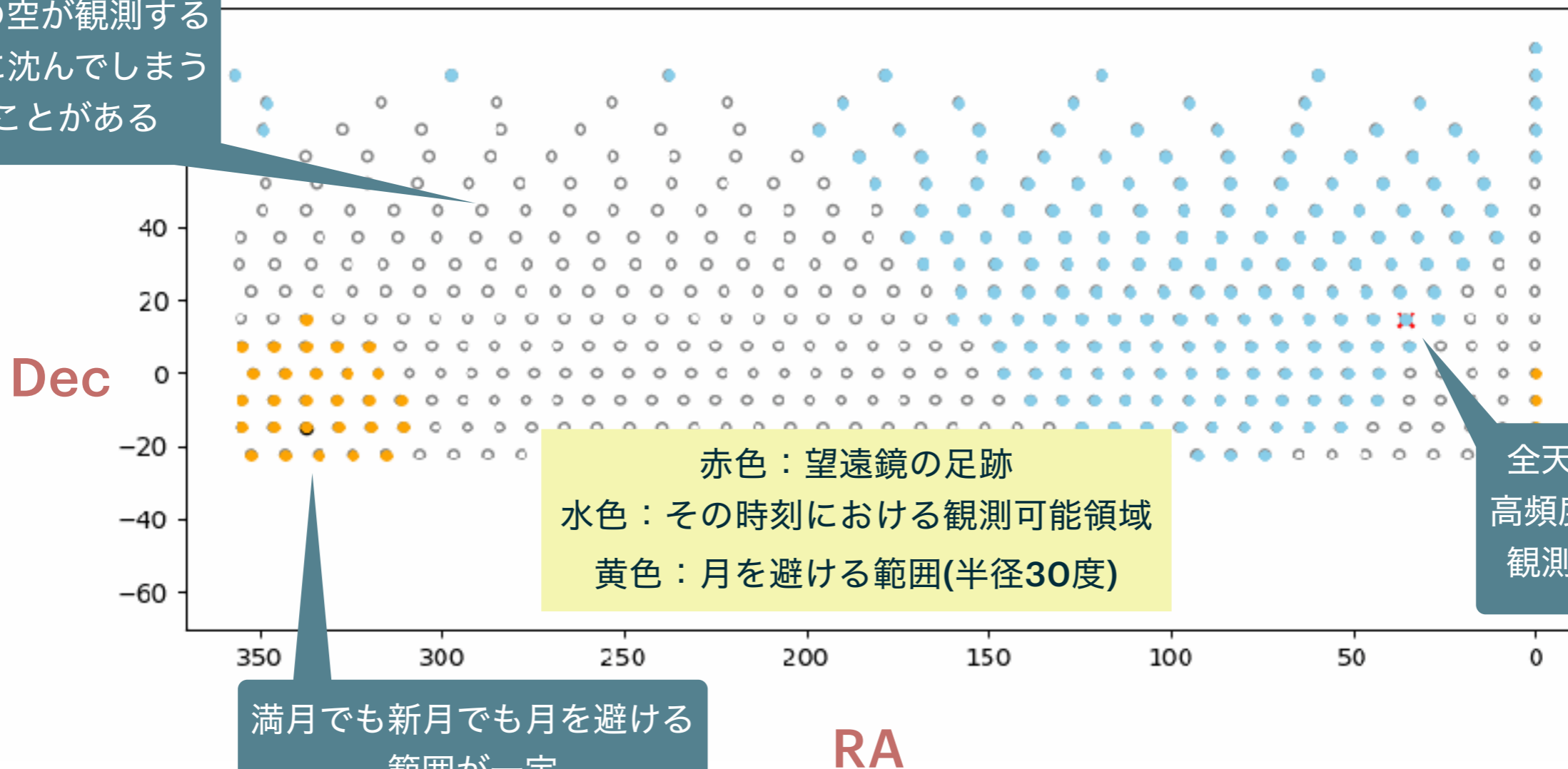


# 既存システムの観測面での要改善点

## 既存システムで立てたある一晚の観測計画

初期位置がランダムなので、夕方の西の空が観測する前に沈んでしまうことがある

2022-10-07 04:58(JST)



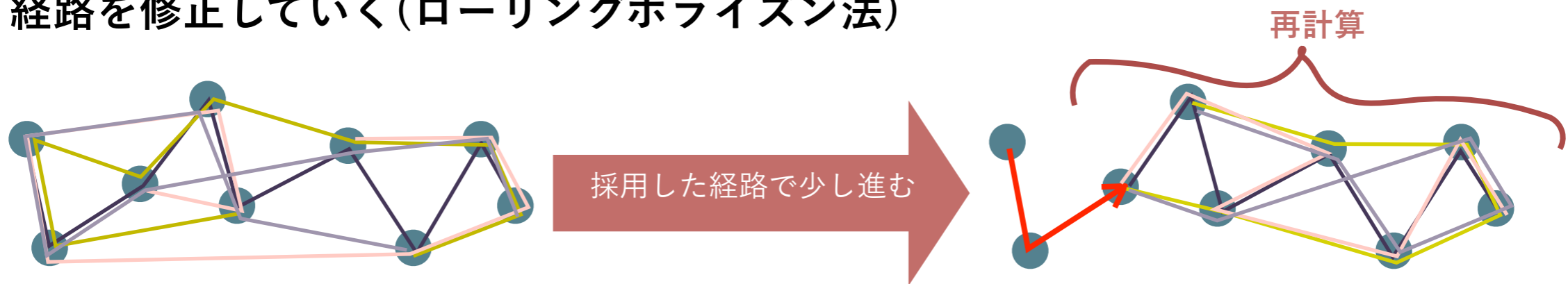
全天サーベイと高頻度サーベイの観測領域が重複

満月でも新月でも月を避ける範囲が一定

# 改良した点

## 動的な最適化システムを実現

- 観測をしながら定期的に最適化をかけ、リアルタイムな情報を反映させながら経路を修正していく(ローリングホライズン法)



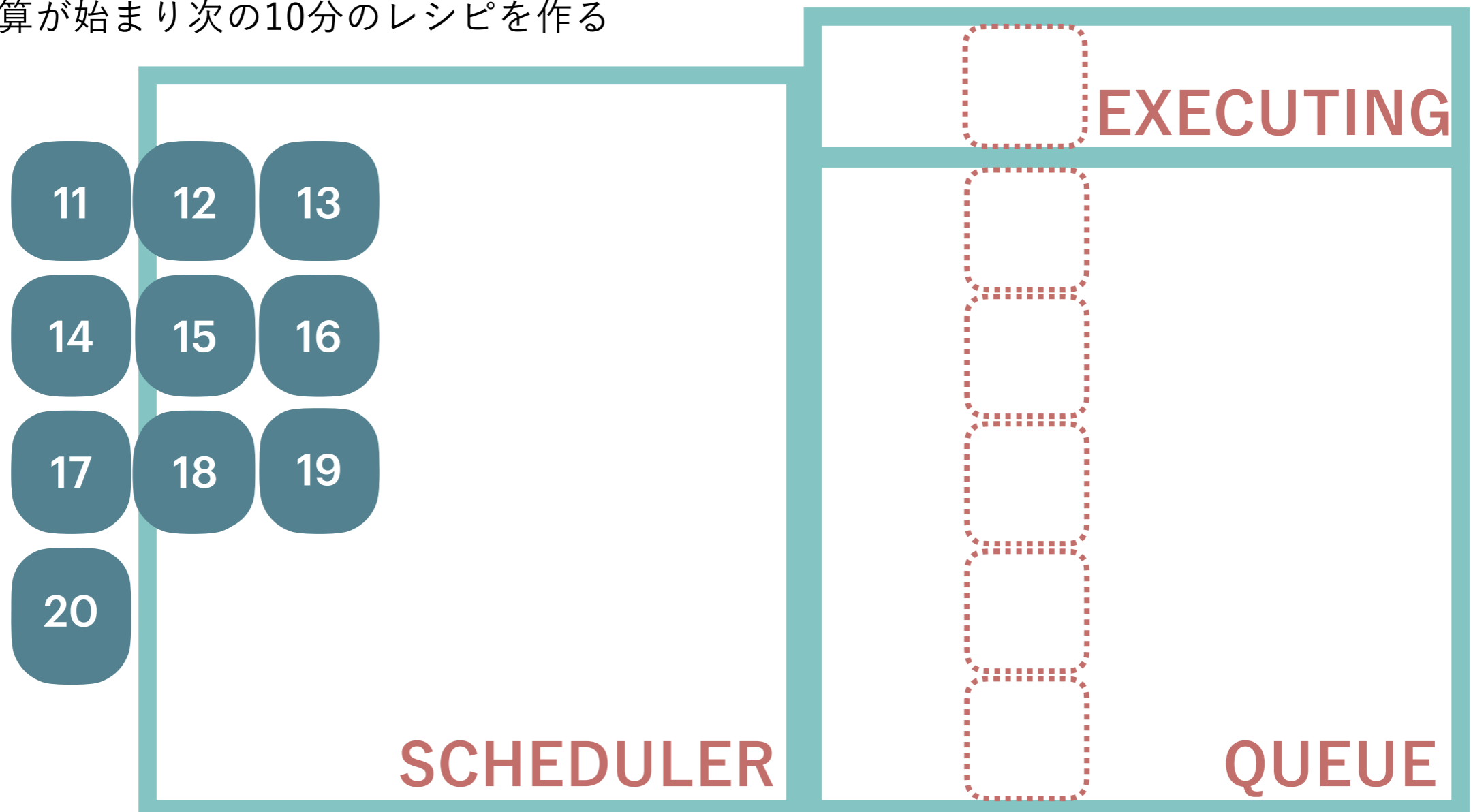
- いつどこを観測したかをレシピ発行の度にデータベースに記録していくことで現在までの観測を踏まえて最適化をかけ直す
- 事前に計算しておくことが可能なデータ(各時刻における観測可能領域)は日没前に準備  
→ 計算時間を大幅に削減
- ・ 全天サーベイでは高頻度サーベイの範囲を避けるように設定
- ・ 月を避ける範囲を月齢に応じて変動させることで無駄を削減
- ・ 初期位置は観測開始時の南西低空( $30^\circ \leq \text{alt} \leq 40^\circ$ ,  $210^\circ \leq \text{azm} \leq 260^\circ$ )からランダムに選択
- ・ 高頻度サーベイは銀河面(銀緯 $\pm 10$ 度)を避けるようにした

# 動的システムの概要

SCHEDULER内のレシピ数を最小限に留める一方で、QUEUEを空にしない

→ 計算した経路で直近10分(個)のレシピを作成し(作成した順に優先度を高くする)、

**後ろから7個目(下の例では④)が実行された**(=SCHEDULERが残り1個になった)ことを確認すると再計算が始まり次の10分のレシピを作る



---

# サーベイ中断時の対処

- 悪天候によりサーベイが中断し、その後晴れてきて再開する場合
  - … ドームが閉じるとQUEUE内のレシピが消される仕様  
(=後ろから7個目が永遠に実行されない可能性がある)
    - QUEUE内の件数と天候条件を常時監視
    - 新しいレシピを作り出す条件を追加：「**QUEUEが2件以下かつ天候条件OKだったら**」  
(QUEUEの件数だけだとドームが閉じている間作り続けてしまう)
- サーベイ以外の観測を一時的に行った場合
  - … (i) サーベイより高い優先度でSCHEDULERにレシピが投入される
  - (ii) QUEUEに直接手動でレシピが投入される
    - どちらも観測が終わるといずれQUEUEが枯渇する
    - 「QUEUEが2件以下かつ天候条件OKになったら」の条件が機能して新しいレシピが作られる
- 観測可能な領域がなくなりレシピが作れなくなった場合
  - … 避けたい領域(月,銀河面,...)が広く、それ以外は観測したばかりでインターバル条件を満たせない
    - レシピが作れるまで**インターバルを徐々に減らして**みる (全天：-1h, 高頻度：-0.1h)
    - レシピを作ることができたらインターバルを元に戻す



# 変則的な事態の対処

あらゆる変則的な事態を想定した条件分岐を用意

→試験運用時に意図的に以下の事態を作り問題なくシステムが機能することを確認

《システム開始》

- ・悪天候により全天or高頻度サーベイ時間の途中から開始
- ・プログラムが途中で落ち手動で再開
- ・悪天候により一度も観測できずに終了

《サーベイ中》

- ・悪天候による中断、晴れてきて再開
- ・サーベイ以外の観測の実施
- ・観測可能なポイントがなくなる



運用開始！！

(日没前)ポイントごとに観測可能時間帯を計算

サーベイ開始・終了時刻の設定



- ・天候OK and 今日初めての観測 and 開始時刻を過ぎた → 初期位置の選択 → **break**
- ・天気OK and 開始時刻を過ぎた → **break**
- ・終了時刻を過ぎた → システム終了



- ・終了時刻を過ぎた → システム終了
- ・はじめて高頻度サーベイ開始時刻を過ぎた → モード切り替え → **continue**
- ・作成したレシピ最新7個のどれかが実行 → **break**
- ・QUEUEが2件以下 and 天候OK → **break**

レシピ10個作成

- ・観測可能なポイントがない → インターバル減らす → **continue**

作成したレシピをSCHEDULERに投入

インターバル通常値に設定

→システムフロー

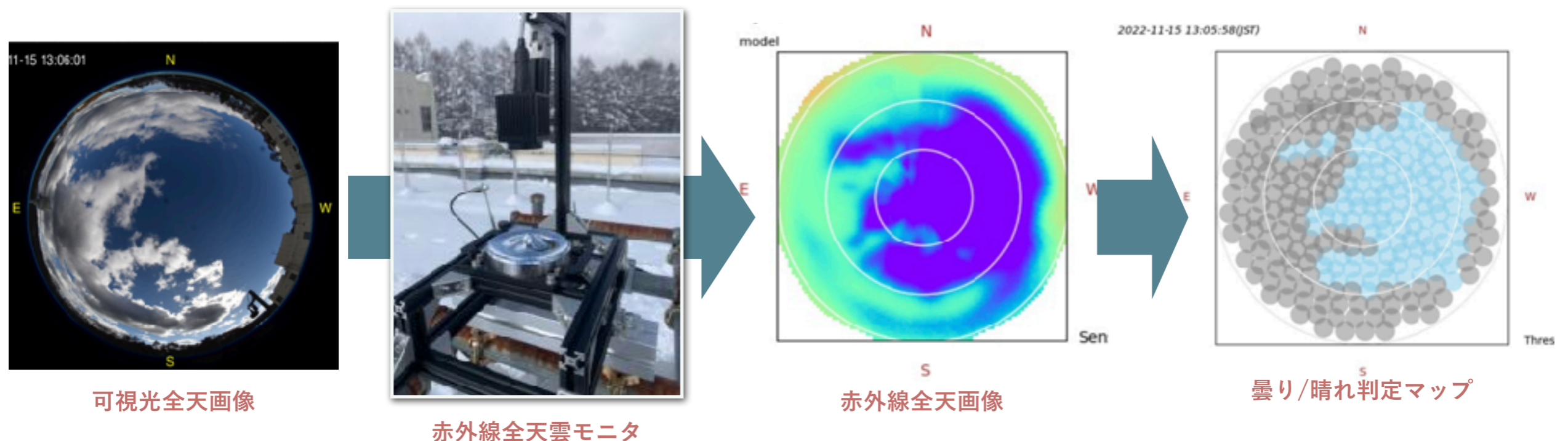
# 雲を避けたサーベイ経路の実現へ

## —— 赤外線全天雲モニタの開発

Tsutsuki R. et al., 2022, SPIE 12184, 121848F / 光学系 特許出願中

リアルタイムな全天の雲分布を把握し、雲を避けたサーベイ経路の最適化を目指す

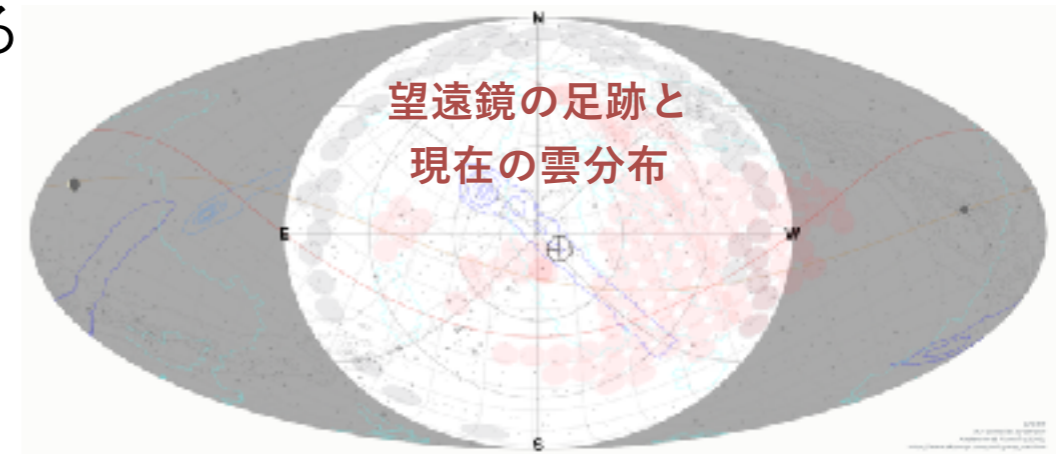
- ・ 赤外線全天カメラを開発し、全天の雲分布を検出することに成功
- ・ 赤外線全天画像からTomo-e Gozenの観測ポイントごとに視野領域内の赤外線放射量を測定し、経験的な閾値をもとに曇り/晴れを判定



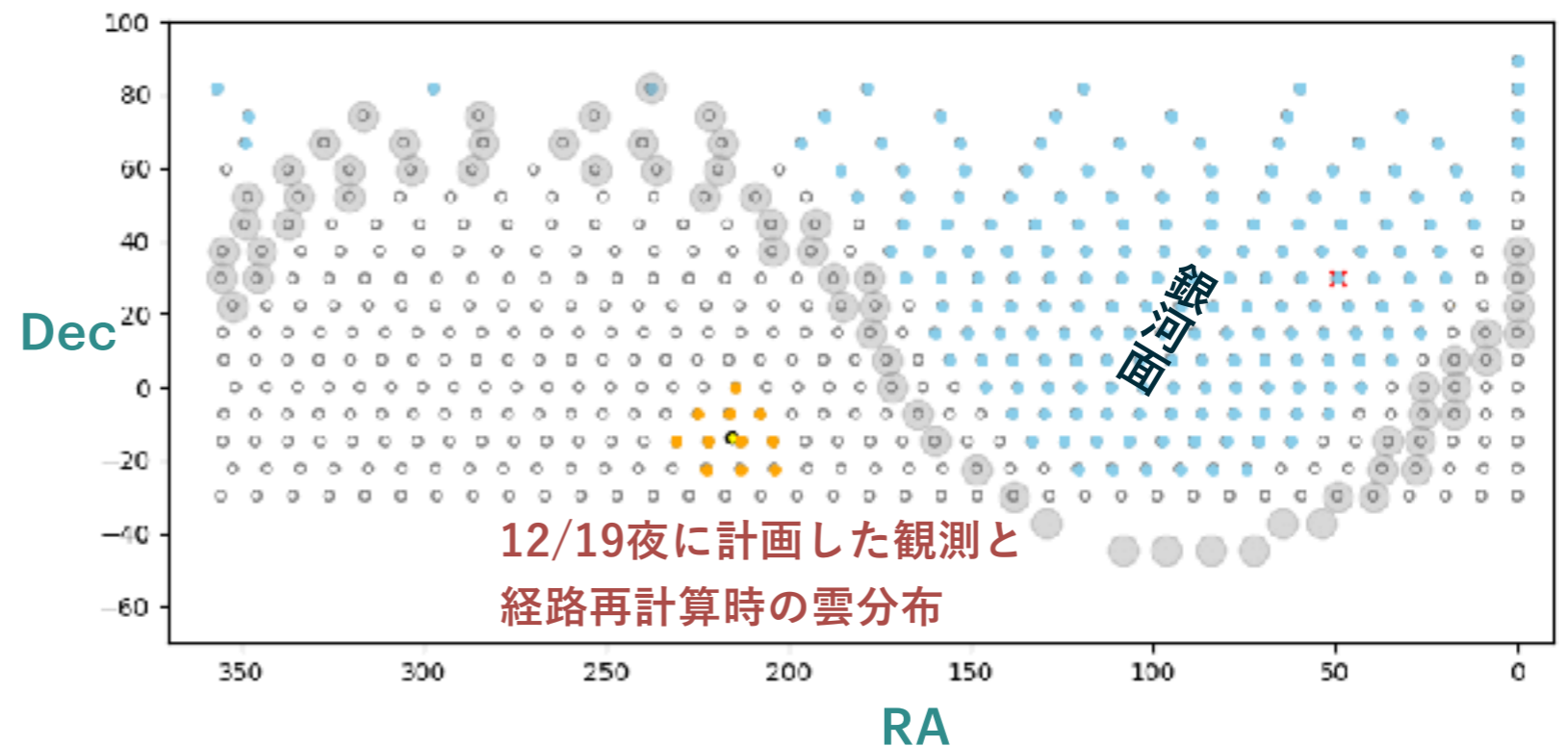
# 雲モニタとの連携を開始

- 曇り領域を観測しても画像が破棄されてしまうため曇り領域を避けることでデータ取得の効率化を図る
- 雲モニタで取得したリアルタイムな曇り領域を最適化プログラムに渡し、**晴れ間領域内**で経路を最適化する
- 雲モニタとの連携観測を**  
**今月(2022/12)に開始した**
- 将来的には簡単な**画像予測**により**現在から数十分晴れ**続ける領域を選定し、  
その中で経路を最適化する

Tomo-e Gozen  
Skymap

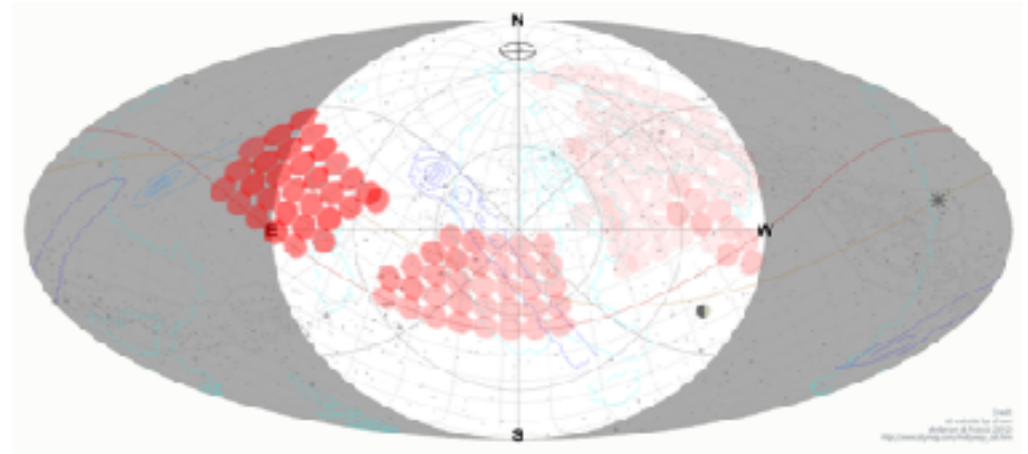


2022-12-20T00:13(JST)



# まとめ

- ・ 木曾観測所Tomo-e Gozenでは  
短時間突発現象や高速移動天体の観測を目的とした  
広域サーベイ観測を每晚実施しており、  
最初に全天サーベイその後特定領域内で高頻度突発天体サーベイを行う
- ・ 広域内を効率良く観測するためのアルゴリズムは重力波可視光追観測でも有用
- ・ サーベイ経路の最適化問題は厳密解を求めるのが難しいため巡回セールスマン問題の最近傍法をもとに独自のアルゴリズムを開発(Pedro,池田,諸隈)
- ・ 既存システムは一晩中サーベイが中断されないことを想定した観測計画をそのまま実行していたため天候や他の観測イベントに対応できていなかった
- ・ リアルタイムな情報を元に定期的に最適化をかけ直す動的な最適化システムへと改良した
- ・ 「最近作成したレシピのうち後ろから7個が実行された」もしくは「QUEUE内のレシピが2件以下でかつ天候条件良し」の場合、経路の再計算と新たなレシピ作成を行う
- ・ 試験運用を経て本格運用を開始
- ・ 赤外線全天雲モニタと連携して雲を避けたサーベイ経路の最適化を開始



初運用時の望遠鏡の足跡