

SAGAS: Simulated annealing and greedy algorithm scheduler for laboratory automation

新井 悠也 (筑波大学 生物学類) 指導教員: 尾崎 遼 (筑波大学 医学医療系)

【Introduction】

生命科学実験は、実験者・実験スケジュール・使用する材料 (試薬や生物) など、様々な要素によって構成される。現在、人間が手動で行う生命科学実験には幾つかの問題が存在する。まず、実験には実験者の多くのリソースが割かれる。また、実験によって得られた知見は、論文にまとめて公開されるが、その再現性は高いとは言いがたい。実際、製薬会社への調査では、著者への問い合わせを行っても 50% の論文の再現性が取れないという結果が出ている [1]。こういった問題に対して、実験自動化という研究分野がある。

実験自動化とは、実験者の手技を実験機器で代替し、実験そのものへの人間の介入を減らす研究分野である。実験自動化は、複雑な実験手順や大量の実験実行が求められる中で、実験結果の品質の保証、実験コストの削減、実験効率の向上に重要である。

自動化された実験室において多くの実験を効率よく処理するためには、従来の人間用の実験スケジュールではなく、各実験自動化機器に実験手順を適切に割り当てたスケジュールが必要となる。そのスケジュールでは、サンプルの変化や変性や分解を防ぐために実験手順同士の前後関係 (dependency) や、時間制約 (TCMB) を考慮する必要がある。

このスケジューリング問題 (S-LAB 問題) に対し、Itoh らは、分枝限定法というアルゴリズムを用いて時間制約を満たすスケジューラ Slab.jl を開発した [2]。Slab.jl では制約を満たしつつ最も実験時間 (execution time) が最も短いスケジュール (厳密解) を出力できるが、複雑な実験に対してはスケジュール時間が非常に長くなるという問題点があった。そこで本研究では、焼きなまし法と貪欲法を組み合わせたスケジューリング手法、SAGAS を提案する。SAGAS は、Slab.jl では現実的な時間でスケジュール不可能な問題にも、制約を満たしたスケジュールを出力することが示された。さらに、グリッドサーチを用いたシミュレーションに基づく実験室構成の設計 (実験室設計) に SAGAS を利用できることを示す。

【Method】

1. 問題設定

スケジュールとは、複数の実験から構成される実験群を、実験機器群に割り当てることである。それぞれの実験は1つ以上の実験手順から構成される。それぞれの機器は固有のタイプを持ち、それぞれの実験手順は処理できる機器のタイプがただ一つ決まっている。さらに、各実験手順は2つの制約、Dependency (ある実験手順は、ある実験手順の終了後に開始しなければならない。) と TCMB (ある実験手順の開始時刻とある実験手順の開始時刻の差の絶対値は、 α 以内でなければならない。 (α は TCMB ごとの定数)) を持つ。

2. 提案手法

今回提案する SAGAS は、メタヒューリスティックアルゴリズムである焼き鈍し法 (SA) と、貪欲法というアルゴリズムを用いてスケジュールを探索する。SAGAS は SA の後に貪欲法により修正 (Mod) を行うパートと、単純な貪欲法 (Greedy) のみのパ

ートに分かれる。SAGAS はこれらのうち、より良いスケジュールを出力する (図1)。以下では、SA-Mod の性能評価の基準として Greedy を用いる。また、実験室設計では、4種類の機器を用いる複雑な S-LAB 問題を用いてそれぞれの機器の台数を 1-4 の範囲で変化させたすべての組み合わせでそれぞれスケジュールを 5 回ずつ行い、それぞれの機器構成での最小値を比較した。

【結果】

1. Slab.jl との比較

Slab.jl が 1.6 sec で計算を終了した問題に対して、SAGAS が同程度の計算時間で厳密解を求めることができるかを検証するため、平均 1.5 sec の計算時間でスケジュールを 5 回実施した結果、SAGAS は 5 回とも厳密解を出力した。

2. 複雑な S-LAB 問題への適用

Slab.jl が 1 h 以内でスケジュールを発見できなかった 3 種の S-LAB 問題に対し、3 min の SA でスケジュールを 5 回ずつ実施した。その結果、SAGAS はすべての S-LAB 問題において制約を満たした解を見つけることに成功した。SA-mod 解の execution time は、Greedy と比較して最大で 37% 短縮された (図2)。

3. 実験室設計

タイプ 3、4 の実験機器の数を増やすと、タイプ 1、2 の実験機器の数を増やすよりも影響が大きいことが示された。詳細は当日の発表にて述べる。

【結論】

SAGAS は、S-LAB 問題の厳密解または制約を満たした解を数分で求められることが示され、S-LAB 問題に幅広く適応できることが示唆された。さらに、SAGAS の計算速度を生かすことで、シミュレーションに基づく実験室構成設計もグリッドサーチで実現できることが示された。これらは生命科学実験の実験室自動化に大きく寄与する結果である。

一方で、SAGAS のアルゴリズムでは、装置の物理的な空間配置を考慮していないこと、問題によっては Mod step にさらに数分の計算時間が必要なこと、静的なスケジュールを求めるため機器の故障を考慮できないことなどが、今後の課題である。

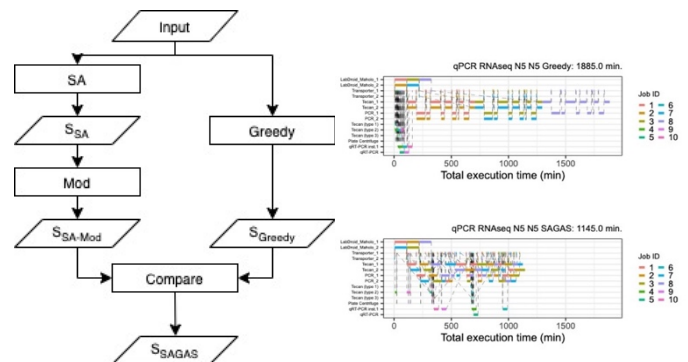


図1. SAGAS の主要なステップ 図2. スケジュール結果の例

【References】

[1]. Prinz, F. et al., Nat. Rev. Drug Discov. 10, 712 (2011).
 [2]. Itoh, Takeshi D. et al., SLAS Technology 26 (6), 650-59 (2021).