

利用者による制約を記述可能にするナーススケジューリングシステムの提案 A nurse scheduling system that allows users to describe constraints

上野 智弘¹⁾ 乃村 能成²⁾
Tomohiro Ueno Yoshinari Nomura

1. はじめに

ナーススケジューリングは病院独自の制約の存在や多様な勤務形態の存在により複雑になっている。しかし、病院ごとに異なるスケジューリングシステムを作成することは現実的ではない。そこで、ユーザが容易に制約を変更できるシステムが必要となる。本稿ではプログラミング未経験者が使用することを想定するナーススケジューリングシステムを提案する。

2. ナーススケジューリング問題

2.1 スケジューリング資源

ナースのスケジュールを構成する要素を図 1 で示し、詳細を以下に記述する。勤務時間枠と看護師をスケジューリング資源（以下、資源と呼ぶ）として定義する。勤務時間枠は“日勤”や“準夜勤”等の形態と日付と曜日という属性を持ち、看護師は“チーム”や“ラダーレベル”等のグループという属性を持つと考える。ラダーレベルとは、看護師の能力評価の指針であり、ベテランであるほど大きい。

2.2 制約

ナーススケジューリングで考慮する制約は、病院ごとに異なる。しかし個別の制約は少しづつ異なるものの、多くの病院によく似た制約が存在する。そこで、まず特定の病院の具体的な制約を収集し、それを汎化することでパラメトリックな制約を導くことにする。

看護師が 25 人在籍している病院(病棟)でスケジューリングを行っている看護師から、実際に考慮している 15 個のハード制約と 6 個のソフト制約の計 21 個の制約を取得した。そして、資源の組合せにかける制限として、15 個のハード制約を考慮した。調査した病院独自の制約と見られるものに、“深夜または準夜に、同時勤務が難しいと判断された看護師を同時に割り当てない”というものがある。これは“ある勤務時間枠に特定の看護師の中から最大 1 人割り当てられる”と表現しても同様の意味を持つ。上記の表現により、どの時間枠でどの看護師に対しても制約を適用可能となり汎用性が上がる。これにより、取得した 15 個のハード制約を汎化することで 6 つの制約で表現可能であった。

3. ナーススケジューリングシステム

3.1 提案システム

本稿で提案するシステムは、スケジュール作成者がプログラミングを行うことで、制約を追加するシステムで

- 1) 岡山大学大学院環境生命自然科学研究科, Graduate School of Environment, Life, Natural Science and Technology, Okayama University
- 2) 岡山大学環境生命自然科学学域, Faculty of Environment, Life, Natural Science and Technology, Okayama University

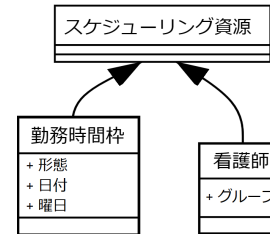


図 1 ナーススケジューリングのスケジューリング資源

ある。これにより、あらかじめ用意されていない独自の制約を考慮できる。また、ナーススケジューリングシステムはユーザとして看護師が想定される。さらに、ユーザである看護師はプログラミング未経験であると想定される。プログラミング未経験が使用しやすい言語として、ビジュアルプログラミング言語が存在する。ビジュアルプログラミング言語は DSL をブロック等の視覚的に理解しやすいオブジェクトとして、記述したものである。そこでビジュアルプログラミング言語を用いることで、プログラミング未経験者でも使用できるシステムにする。3.2 節にて、システムの構成、処理流れおよびデータフォーマットの流れを検討する。

3.2 提案システムの構成

提案システムの構成を図 2 に示す。

図 2 内の語句の説明を以下に示す。

- ユーザ “ユーザ” はナーススケジューリングシステムを使用するスケジュール作成者である。
- 入力インタフェース “入力インタフェース” はユーザがスケジュール作成のために必要となる資源や制約等の情報を入力するインタフェースである。
- DSL 変換部 “DSL 変換部” は DSL を解析し、連言標準形 (CNF) などのスケジュール作成部で求解可能な制約を記述できる形式に変換する。
- スケジュール作成部 “スケジュール作成部” はユーザの入力情報を用いてスケジュール作成を行う。
- 求解結果変換部 “求解結果変換部” は求解結果を人が扱いやすい形式のスケジュールに変換する。
- スケジュール表示部 “スケジュール表示部” はユーザに作成したスケジュール表を表示する。

4. 問題を記述可能な DSL の検討

4.1 既存の DSL の調査

自身のシステムで使用するビジュアルプログラミング言語を定義するにあたり、もともとなる DSL の構成を検討する必要がある。本システムの DSL を検討するにあたり、以下の 2 つを考慮する。

1. スケジューリングに特化した言語であること
2. 様々な制約に対応出来る柔軟性があること

表 1 看護師が 25 人在籍している病院 (病棟) で考慮するハード制約が AUK で記述可能かの検討

通番	制約グループ	協力病院が実際に考慮する制約	記述可能
1	表内に現れる看護師の回数の制約	休みの総数が土日祝の総数以上になる	○
2	複数の時間帯に現れる看護師の回数の制約	夜勤 (準夜と深夜) の総数が 10 以下になる	×
3	単体の時間帯に割り当てる看護師の人数についての制約	土日祝は日勤が 7 人割り当てられる	○
4	単体の時間帯に割り当てるある要素を持った看護師の人数についての制約	深夜と準夜にリーダーが 1 人以上含まれる	×
5	複数の時間帯に割り当てるある要素を持った看護師の回数の制約	夜勤 (準夜と深夜) は 2 連続まで (夜勤専従を除く)	×
6	単体の時間帯に割り当てる特定の看護師について制約	深夜または準夜に, 特定の看護師を同時に割り当てない	×

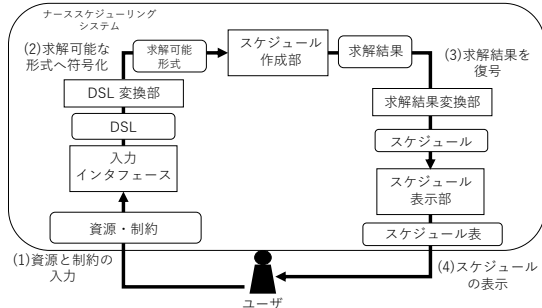


図 2 ナーススケジューリングシステムの構成

そこで上記の点を考慮するために、既存のスケジューリングシステムで使用されている DSL を調査した。時間割編成に関する DSL として、文献 [1] で AUK が提案されている。

AUK は Ruby の内部 DSL であるため、Ruby に準拠した形式である。また AUK は ITC フォーマット [2] を基に提案されている。ITC フォーマットとは、ITC2019 (International Timetabling Competition 2019) で提案されている競技用フォーマットのことであり、AUK は ITC フォーマットと比較してより簡素に記述可能となっている。そのため AUK は記述力が高く、柔軟性のある言語であると言える。しかし、AUK は時間割編成に特化した言語であり、ナーススケジューリングにおける記述力については言及されていない。そこで本研究では AUK でナーススケジューリングを記述可能かを調査する。

4.2 AUK でのナーススケジューリングの記述

AUK の記述能力を確認するために 2.2 節で示した協力病院の資源および制約を AUK で記述することを試みる。

AUK で協力病院の制約を記述可能かについて表 1 で示す。2.2 節にて汎化した 6 個のそれぞれの制約グループについて、該当する制約を 1 個ずつ示す。6 個の制約グループのうち、通番 1 と通番 3 の 2 個の制約グループが AUK で記述可能であることが分かった。しかし、残りの 4 個については記述不可能である。これを踏まえて 4.3 節にて、ナーススケジューリングを記述可能にするための AUK の拡張を検討する。

4.3 AUK の拡張

AUK は、本来大学の時間割を記述するための DSL である。そのため AUK の文法上、記述不可能な制約も存在する。これは図 1 で示した資源に付与している属性に対して制約をかけられないためである。属性を資源として定義することで、AUK で表 1 の 4 個の制約グループについても記述することは可能となる。しかし属性の数だけ資源の数が増えるため、記述が煩雑となる。そこで部分集合としての資源をユーザが簡素に記述するため

の新たな構文が必要となる。新たな構文案として以下の 2 つが考えられる。

案 1 同じ種類の資源を複数指定した集合 (例: 看護師 A と看護師 B からなる集合や、1 日の準夜勤と 1 日の深夜勤からなる集合など) を宣言し、それらを資源とみなし定義する案

案 2 看護師や時間帯という資源に対して属性による部分集合を定める条件式を付記できるようにして、対象となる資源をピックアップする案

つまり、案 1 は部分集合の外延表記、案 2 は内包表記といえる。

4.4 AUK の構文案

4.3 節にてナーススケジューリングを記述可能にするための構文案を検討した。外延表記の案は部分集合となる対象をすべて列挙する必要があるため、資源数が増えると記述が煩雑となるデメリットがある。そこで本システムの DSL では、内包表記である案 2 を採用する。具体的には以下のように記述する。

条件式を用いて部分集合を定義する記述:

資源 (nurse, timeslot) に対し、条件式 (any, ladder <= 2, Sem) を記述する。

ここで“any はすべての”, “ladder <= 2 はラダーレベル 2 以下の”, “Sem はすべての準夜勤の”を示す。

```
nurse {any}
nurse {ladder <= 2}
timeslot {Sem}
```

上記の方針に基づいて AUK を拡張したところ、協力病院で考慮する 15 個のハード制約を記述可能であった。また、拡張後の AUK は条件式を用いて記述するため、競合する資源を列挙する記述方式である拡張前の AUK より簡素に記述可能であるといえる。

5. おわりに

本稿ではナーススケジューリングシステムの構成について提案し、システムの DSL として AUK を拡張した。今後はプログラミング未経験者が使用することを想定したナーススケジューリングシステムの開発を行う。

参考文献

- 松田陸斗, 乃村能成: 大学の時間割編成問題を記述可能な DSL と時間割作成システムの提案, 第 29 回マルチメディア通信と分散処理ワークショップ論文集, 第 29 回マルチメディア通信と分散処理ワークショップ (DPSWS2021), pp. 64-71 (2021).
- Müller, T., Rudová, H. and Müllerová, Z.: ITC2019: International Timetabling Competition 2019, (online), available from (<https://www.itc2019.org>) (accessed 2022-11-22).