CLPを用いたナース・スケジューリング問題における実用解の効率的探索

An Efficient Approach for Searching Practical Solution in Nurse Scheduling using Constraint Logic Programming

西川 理規[*1] 松井 藤五郎[*2] 大和田 勇人[*1]
Riki Nishikawa Tohgorou Matsui Hayato Ohwada

[*1]東京理科大学大学院 理工学研究科 経営工学専攻
Department of Industrial Administration, Graduate School of Science and Technology, Tokyo University of Science

[*2]東京理科大学 理工学部 経営工学科
Department of Industrial Administration, Fanculty of Science and Technology, Tokyo University of Science

Scheduling nurses to staff shifts is a major problem in hospital. The necessity of maintaining a certain level of service and skill in the makeup of every shift, while balancing the workload among the nurses involved, is incredibly difficult. Recently, a lot of research for nurse scheduling problem has been proposed in a broad range of fields, however it is difficult to find available roster effectively on each hospital from many solutions which are gotten with these technique. Therefore, this paper proposes search-algorithm embedded specific constraint (e.g. skill and ability) that each hospital has, using Constraint Logic Programming (CLP) which is one of much technique. Accordingly, with this approach a available roster can be created within several tens of seconds.

1. はじめに

医療現場では、人の命に関わるため、医療スタッフのサービスの質を低下させることは出来ない。特に、24 時間切目なく活動しなくてはいけないナースは、患者と接する機会も多いため、常にナース個人の良好な健康状態を保持する必要がある。よって、ナースの勤務表を作成する（ナース・スケジューリング）には、看護の質とナース個人の生活の質の両方を考慮に入れられている。しかし、看護と個人の質のバランスを取りながら、各病院ごとに持つ業務の関係などの条件を全て満たすことは、非常に難しい。

よって、このような勤務表を作成する問題を解決するために、現場に存在する制約条件を満たし、効率的に解を求められるようなスケジューリングアルゴリズムの研究が現在までにされてきている。例えば、数学的なアルゴリズムを使用した線形(整数) 計画法、近似解を探索するメタヒューリスティックの 1 つである遺伝的アルゴリズムなどを使って、効率的に解を求められるような手法が多く提案されている。

しかし、これまでにさまざまな手法が提案され、ソフトウェアが開発されている一方で、それがまだ広まらない、またはソフトウェアがあっても使われない現状にある。その理由の 1 つとして挙げられるのは、複雑な日本の医療環境にあると考えられる。現在の日本では、慢性的な人員不足であり、この状況下において、考慮しなくてはいけない制約条件（人員不足を補うための制約など）が多数存在する。これによって、ソフトウェアが作成した勤務表が制約を満たしていない場合、手直しや改善をしなくてはいけないという手間が発生してしまい、これがソフトウェアを使わない理由であると考えられる。

よって本研究は、それらの手法の中で、制約プログラミングの要素を持つ制約論理プログラミングに焦点を当てることにする。この制約論理プログラミングは、論理プログラミングに制約解消のプログラムを組み込んだプログラミング言語であり、特定のナースに割り当てを行うなどユーザの意図する要求を論理プログラムによりルール化し、対話型のシステムを実装することが可能であるため、現場が求める勤務表を作成することが期待される手法の 1 つである。つまり、制約を満たさなかった場合の制約もルール化することによって、手直しや改善の手間を省略することが可能であると考えられる。

しかしこの手法は、出来れば守りたい制約の違反が最小になるまで、可能性のある解を全て探索して厳密解を出す分枝限定法を適用する場合が多いが、ナース・スケジューリング問題のように問題の規模が大きくなると、宣言した制約などで解法が複雑になり、探索する時間が非常にかかってしまう。

そこで本研究では、制約論理プログラミングを用いて、対象となる現場に応じた実用的である勤務表を効率的に探索する手法を提案することを目的としている。

連絡先: 西川 理規，東京理科大学理工学研究科経営工学専攻 大和田研究室，千葉県野田市山崎 2641，04(7124)1501，j7409622@ed.noda.tus.ac.jp

2. ナース・スケジューリング

実際の勤務表では、通常、各看護部署で作成されているが、ナースの人数も所属部署も固定されている。対象部署のナース数を m（人）、対象期間を n（日）とすると、m × n といったマトリクスに、日勤や夜勤などといったシフトを割り当てていくことになる。

また、勤務表作成において考慮する必要がある基本的な制約条件は、大きく分けて 5 つの条件を考慮に入れる必要がある。

- 毎日の各勤務に必要な人数を確保すること
- スキルレベルなどを考慮して各勤務のメンバーを構成すること
- 各看護婦について各勤務の回数が決められた範囲であること
- セミナーなどその他の業務や休みの希望を達成すること
- 禁止される勤務パターンを入れないこと

しかし、現実的にこれらの制約条件を全て満たすことは、非常に難しいため、ある程度は妥協したものにする必要がある。そこで、多数ある制約条件を、必ず守らなくてはいけない制約条件（HARD 制約）と出来れば守りたい制約条件（SOFT 制約）の 2 つに分けることにより、制約を緩和させて、問題を解決をすることが多くなっている。

3. 制約論理プログラミング

制約論理プログラミング（CLP）とは、Prolog などの論理プログラミングに、制約解消を目的とした制約プログラミングを組み込んだプログラミング言語である。

この制約プログラミングとは、変数間の関係を制約という形で記述するプログラミングパラダイムの一種であり、解決したい問題を制約の集合として記述してコンピュータに与えると、コンピュータが制約を満たした解を導き出すことを目的としたものである。

この制約論理プログラミングは、「宣言性」「制約による探索の効率化」「ホスト言語が Prolog」という 3 つの特徴を持つ。特に「宣言性」の特徴により、問題の記述（制約を宣言）と探索を別に考えることが出来る。

また、この制約論理プログラミングを使ったナース・スケジューリングの手法も多数提案されてきているが、日本のように人手不足が背景にある病院では、人手を補うための制約や特定のナースしか出来ない作業があるという制約などが存在するため、非常に多くの制約を考慮しなくてはいけない。この状況で INTERDIP([2]) のような従来の手法を用いると、SOFT 制約が多く存在することになり、非常に非効率な探索となり、特定の解を出すことが出来ない。

4. 提案手法

4.1 SP 制約

本研究では、従来手法で SOFT 制約として扱われていた制約を 2 つに分割し、制約を「HARD 制約」「SP(Specific) 制約」「SOFT 制約」という 3 つ扱うことにする。

この SP 制約とは、従来の SOFT 制約の中で特に優先順位が高いもののことである。例えば、ナース・スケジューリング問題における SP 制約は、「スキル」「チーム」などの各ナースの持つ制約であったり、「役職によって日勤の割り当て数が異なる」などの対象とする現場が持つ特有の制約が含まれ、HARD 制約よりは厳密性を必要としないものである。また、SP 制約を満たさなかった場合の制約も含まれる。

またその他（優先順位が低い）の制約は、従来どおり SOFT 制約と呼ぶことにする。

4.2 アルゴリズム

本研究では、HARD 制約と SP 制約を満たしつつ、SOFT 制約違反が最小となる解を探索する手法を提案する。

全体のアルゴリズムは、図 1 である。この全体のアルゴリズムは、INTERDIP([2]) の「シフトの種類ごとに割り当てをするフェーズが違う」という特徴を取り入れている。

このアルゴリズムは、まずナースの人数（人）× 対象期間（日）の勤務マトリクスを生成し、HARD 制約と SOFT 制約を宣言する。次にその宣言された勤務マトリクスに、事前に分かっているナースの希望する日のシフトの割り当てを行う。そして、各シフトを割り当てると同時に SP 制約を満たすようにする探索アルゴリズムにマトリクスを渡す。一通り割り当てた勤務マトリクスに、まだ割り当てられていないセルに対して、

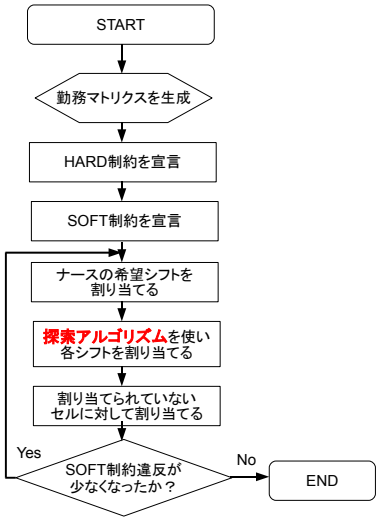

図 1: 全体のアルゴリズム

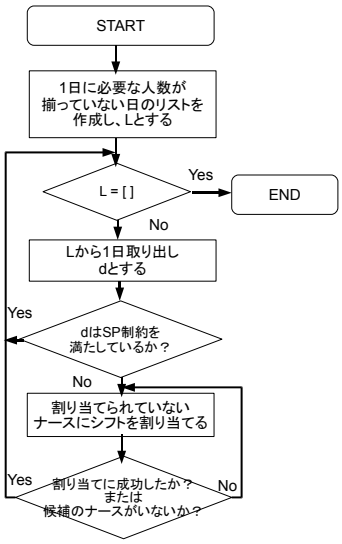

図 2: 探索アルゴリズム

HARD 制約を元に探索を行い、候補の解を出す。その候補の解に、どれだけ SOFT 制約の違反があるのかを調べ、もう 1 度探索を開始する。そして、再び探索された勤務表の SOFT 制約違反が前回求められた勤務表よりも変わらない場合は、プログラムは終了し勤務表が完成するが、小さい場合は、もう 1 度探索を行う、という内容となっている。

各シフトを割り当てる探索アルゴリズムは、図 2 である。まず、勤務マトリクスから各シフトごとの必要な人数が揃っていない日をリスト L に格納する。そのリストが空で無い場合、リストの 1 番先頭の日にち情報 d を取り出す。

そして、この d に割り当てがされているナースと割り当てがされていないナースをそれぞれリストに格納する。この割り当てられているナースのリストに、SP 制約を満たしているナースがいるかを調べ、満たしていない場合は割り当てを行う。例えば、熟練者が 1 人必要であるという SP 制約がある場合、割り当てされているナースに熟練者がいるかどうかを調べ、いない場合は制約を満たしていないことになるため、割り当てられ

ていないナースのリストから熟練者の候補を取り出す。その候補は、割り当てが少ないナースを優先的に選び、割り当てを行う。一方、候補のナースが無い場合、SP 制約の違反を無視して、処理を終了する。

また、候補のナースに割り当てを行う際、シフトの並びや勤務数などで割り当てが失敗する場合がある。その場合は、再び候補を選び、割り当てを行う。

このアルゴリズムで SP 制約を満たす解を探索し、この探索結果から分枝限定法を使うと、予め狭められた領域から SOFT 制約違反が最小である解を探索するため、効率的に解を出すことが出来る。

4.3 実行例

本研究では、A 病院の薬剤師ナースのデータを使用する。日勤と夜勤の 2 交替制であり、ナースは 16 人、スケジューリング期間は 28 日（4 週間）となっている。

また、以下のような基本的な制約が存在する。

- 日勤は 8 時間、夜勤は 16 時間勤務である
- 労働時間は 4 週間で 160 時間以内
- 平日の夜勤は 1 人が担当
- 休日の日勤と夜勤は各 1 人が担当
- 夜勤の次の日は休み（平日）
- その他の勤務は通常の日勤として扱われる

さらに、各ナースにはそれぞれ「部長（チーフ）」「パートタイマー」「ドラッグインフォメーション」「ケモセラピー（抗がん剤）担当」「HIV 薬担当」「病棟担当」「薬局担当」「発注担当」という業務担当があり、各担当に制約がそれぞれ存在する。

このサンプルデータを元に HARD 制約と SOFT 制約、SP 制約の 3 つに制約を分ける。表 1 は、その分けた制約を示している。

表 1: 3 つの制約

HARD制約

・4週間で160時間以内 ・7連続勤務禁止
・平日と休日のシフト数（日勤・夜勤）

SOFT制約

・ケモセラピー担当が抗がん剤作りを出来るだけ行う（3パターン）

SP制約

・各日の必要人数（日勤）は、仕事によって異なる
・曜日制約がある（祝日の前日には必要人数が増えるなど）
・各ナースに担当する仕事がある
　パート、婦長、ドラッグインフォメーション
　ケモ担当、病棟長・病棟担当、薬局
・各ナースに出来る・出来ない仕事とシフト(シフトパターン）がある

この制約を図 1 と図 2 のアルゴリズムに組み込むことになる。まず、「勤務時間」「平日と休日のシフト数」「7 連続勤務禁止」を HARD 制約として、そして「抗がん剤作業は抗がん剤担当ナースが行う」を SOFT 制約として宣言を行う。

そして、要望があるナースに対してのシフト割り当てを行い、「抗がん剤 (4-1)」「休日夜勤」「平日夜勤」「休日日勤」「抗がん剤 (3)」「抗がん剤 (2)」「平日日勤」という SP 制約の強い順にそれぞれ割り当てを行っていく。この割り当てを行う探索戦略は、各ナースが出来る・出来ないシフトや仕事などの SP 制約を考慮して、割り当てを行うことになる。例えば、パートタイマーに対して日勤を割り当てる場合、既に割り当てられているナースのリストの中にパートタイマーのナースがある場合、その日の探索は終了する。一方で、日勤が割り当てられていない場合は、SP 制約を満たしていないことになるため、割り当てられていないナースのリストからパートタイマーであるナースを取り出し、その中で日勤の割り当てが少ないナースを選び、割り当てを行う。割り当てに失敗した場合は、再び候補を選び、割り当てを行う。

そして、一通り割り当てた勤務表に、まだ割り当てられていないセルに対して、HARD 制約を元に探索を行い、勤務表を完成させる。この勤務表が前回の解より SOFT 制約違反が小さくなったのかを調べ、小さくなった場合は再び探索し、変化が無い場合は探索を終了する。これにより、勤務表が完成する。

5. 実験

5.1 実験環境

実験に使用した環境は以下のとおりである。

CPU： Intel 1.06GHz

メモリ： 1.49GB

OS： Microsoft Windows XP

処理系： ECLiPSe 5.10

5.2 実験結果

制約論理プログラミングシステムである ECLiPSe([3]) を使い、データサンプルを元に 3 つの期間 (10 月 10 日〜11 月 6 日、11 月 7 日〜12 月 4 日、12 月 5 日〜1 月 1 日）の勤務表を作成した。その結果の一例は、表 2 である。

表 2: 勤務表 (10 月)

	JOB	10月10日	11日	12日	13日	14日	15日
1	part	Day	Rest	Rest	Rest	Rest	Day
2	part	Day	Rest	Rest	Rest	Day	Rest
3	chief	Day	Rest	Rest	Rest	Day(3)	Day
4	chief	Day	Rest	Rest	Rest	Day	**Rest**
5	drag inf	Rest	**Rest**	**Rest**	Day	**Day**	**Day**
6	drag inf	Day	Rest	Rest	Rest	Day(2)	Rest
7	other	Day	Rest	Rest	Rest	Day	Day
8	other	Rest	Rest	Rest	Rest	Day	Day
9	chemotherapy	Day(1-4)	Rest	Day	Rest	Day	Day(3)
10	chemotherapy	Day(3)	Day	Rest	Rest	Day(1-4)	Night(2)
11	chemotherapy	**Night(2)**	/	Rest	Rest	Rest	Day(1-4)
12	ward chief	**Rest**	Rest	Rest	Rest	Day	Day
13	ward chief	Day	Rest	Rest	Rest	Night	/
14	ward staff	Day	Rest	Rest	Night	/	Day
15	ward staff	Day	Rest	Night	/	Day	Rest
16	ward staff	Day	Night	/	Rest	Day	**Day**

この結果では、日勤は「Day」、夜勤は「Night」、休みは「Rest」、抗がん剤作業 4-1 は「Day(4-1)」、抗がん剤作業 2 は「Day(2)」、抗がん剤 3 は「Day(3)」、抗がん剤作業 2 を担当する夜勤は「Night(2)」、夜勤の次の日の半日シフトは「/」と表現され、灰色のセルは事前にナースからの要望があったシフトとなっている。

探索時間と SOFT 制約違反の結果である表 3 から、それぞれの期間で 1 個目の解を出す探索時間が約 7.5 秒、さらに全

表 3: 3 期間の探索時間と SOFT 制約違反

期間 (28日間)	探索 (1回目)	探索 (全体)	SOFT制約違反 (違反数/全体)
10/10 – 11/6	7.8(s)	16.28(s)	17/54
11/7 – 12/4	7.7(s)	13.89(s)	20/57
12/5 – 1/1	7.03(s)	14.28(s)	18/45

体の探索時間が約 15 秒という短時間で勤務表を完成させることが出来ている。また、SP 制約の違反は極端に小さくなっている一方で、SOFT 制約の違反は極端に小さいとはいえない。しかし、データ元である A 病院からは問題なく利用出来るという評価であった。

よって、提案したアルゴリズムを使って、現場に応じた勤務表を効率的に探索することが出来たといえる。

6. 結論

本研究では、制約論理プログラミングを用いて、対象となる現場に応じた実用的である勤務表を効率的に探索するために、従来の研究で SOFT 制約として扱われていた制約をさらに、優先度が高い「SP 制約」と優先度の低い「SOFT 制約」の 2 つに分割した。そして、SP 制約を考慮したヒューリスティックな探索と SOFT 制約違反を最小にする分枝限定法を合わせたアルゴリズムを提案した。

これにより、データサンプルにおいて解を効率的に探索出来、さらにその解が実際の現場でも問題なく利用出来るということを確認することが出来た。

参考文献

[1] 池上 敦子 (2005). ナース・スケジューリング-調査・モデル化・アルゴリズム-, 統計数理, 第 53 巻 2 号, 231–259.

[2] Slim Abdennadher, Hans Schlenker (1999). INTER-DIP - An Interactive Constraint Based Nurse Scheduler, , PACLP99, London, 1999.

[3] Krzysztof R. Apt and Mark G. Wallace (2007). Constraint Logic Programming using ECLiPSe , CAMBRIDGE.