

# 制約違反最小化戦略による室内レイアウト変更計画システム

作原宏美 小野智司 水野一徳 福井幸男 西原 清一  
筑波大学 電子・情報工学系

## 1 はじめに

近年, 計算機を用いて効率的に室内の機器レイアウトを作成するシステムが多く開発されている [1]. しかし, 現在のレイアウトを新たなレイアウトへと変更する場合, 機器の移動作業手順を事前に導出せず作業を行うことが多い. そのため, 作業者が多いにも関わらず効率的に作業できないことがある. 実際の作業を行う前に, 機器の移動経路を決定し, 作業者の分担を決定しておくことにより, 効率的な作業が可能となる. 本稿では, 機器の移動経路および移動作業スケジュールを求める問題を室内レイアウト変更計画問題 (ReArrangement of Floor Layouts: RAFL) と呼び, RAFL を効率的に解くシステムを提案する. 提案するシステムは, RAFL を制約充足問題 (Constraint Satisfaction Problem: CSP) [2] として定式化することで, プランニングとスケジューリングを統一的に扱い, 違反の種類に応じて値割り当てオペレータを使い分けることで, 効率的に作業計画を立案する.

## 2 室内レイアウト変更計画問題

### 2.1 特徴

RAFL は, 変更前後のレイアウト, 作業員 (数), および総作業時間の上限が与えられた場合に, 移動経路からなるプランと, 移動作業担当者および作業タイミングの割り当てからなるスケジュールを求める問題である (図 1). RAFL において, プランニングとスケジューリングを段階的に実行すると, 作業員の割り当てが不可能な場合に, 再度プランニングを行わなければならない, 効率が悪くなる. よって, プランニングとスケジューリングを統一的に扱う必要がある. RAFL は, 空間的競合, すなわち, 機器間の経路の競合, および資源的競合, すなわち, 移動作業間で作業員の奪い合いが発生する. 本問題は, 機器数が増加するにつれて, 機器間での競合発生件数が増大し, 探索空間が指数関数的に増大することとあいまって, 解くことが困難になる.

### 2.2 提案するシステムの方針

2.1 節で述べた特徴を持つ RAFL に対し, 本システムでは, 以下の方針をとる.

方針 1: RAFL を CSP として定式化し, breakout 法によって探索を行う. RAFL を CSP として定式化することで, プランニングとスケジューリングの双方を統一的に扱う. これにより, 汎用の解法を用いることができる. 本システムは, 膨大な探索空間を持つ RAFL

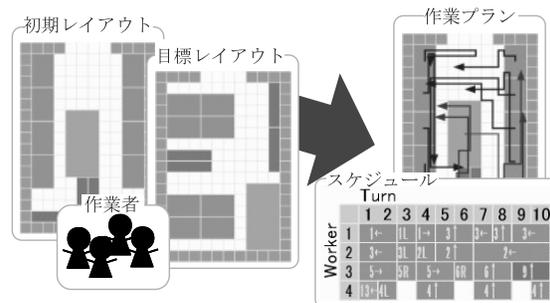


図 1: 室内レイアウト変更計画問題

を効率的に解くために, 制約違反最小化戦略に基づく山登り法 (Min-Conflicts Hill-Climbing: MCHC) [3] を用いる. また, breakout 戦略 [4] を組み合わせることで, 局所最適解からの脱出を可能にし, リスタート戦略を用いることで, 大域的な探索能力の不足を補う.

方針 2: 違反の種類に特化したオペレータにより, 値割り当てを変更する.

本問題では, 空間的競合と資源的競合が併発するため, 1 変数ずつ値割り当てを変更していく方法では, 競合を解消できない (3.3 節). また, MCHC では, とりうる値全てを次ステップの解候補とし, それらの制約違反状況をチェックするため, 問題の規模が大きくなると, 解候補数が増大し, 探索効率が悪化する. よって, 各々の制約に特化した値割り当てオペレータを用意することで, 効率的な競合解消および解候補数の増加防止を実現する.

## 3 提案する方式

### 3.1 定式化

本システムでは, 機器座標変数 (Object Variable: OV) と作業員変数 (Worker Variable: WV) の 2 種類の変数を用いる. OV は, 各時刻における各機器の座標および方向を表す変数であり, 値の変域は,  $x$  座標,  $y$  座標, および機器の方向 (東西南北) からなる値組の集合である. WV は, 各ターンにおける各移動作業の担当作業員を表す変数であり, その値の変域は, 作業員集合のべき集合である. ここでフロアサイズを  $X, Y$ , 総作業員数を  $W$ , 総作業員ターン数を  $T$ , 機器数を  $I$  とすると, 探索空間のサイズは  $(XY \times 2^W)^{IT}$  となる. 15 個の機器, 6 人の作業員, 49 ターン, および  $20 \times 15$  マスのフロアから構成される問題では, 探索空間はおおよそ  $10^{3212}$  となる. 表 1 に本システムで用いる制約の例を示す. breakout 戦略を適用するために, 制約の各インスタンスに違反点数を設定する.

### 3.2 探索

本システムにおける探索の処理手順を図 4 に示す. 以下では各処理の詳細を述べる.

Floor Layout Rearrangement Based on Min-Conflicts Hill-Climbing  
Sakuhara, H., Ono, S., Mizuno, K., Fukui, Y., Nishihara, S.  
Institute of Information Sciences and Electronics, University of  
Tsukuba

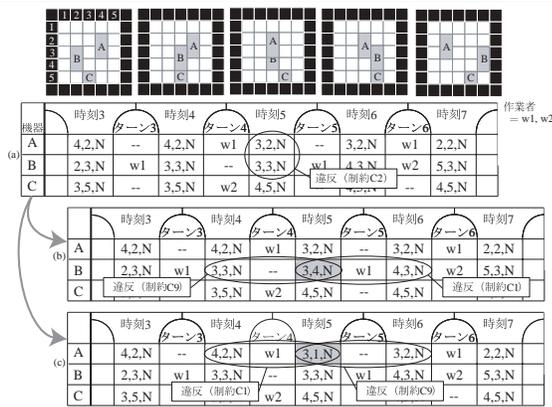


図 2: MCHC では競合を解消できない例

表 1: 制約とオペレータの例

	制約	オペレータ
制約 C1	機器は 1 ターンに 1 マスの移動しか行えない	制約違反を起こしているどちらか一方の変数をもう一方の変数の値と連続になるように変更し、さらに greedy 法によって経路をつなげる
制約 C2	同一時刻において機器同士が重ならない	重なっている機器同士を接するように OV を変更し、さらに greedy 法によって経路をつなげる
制約 C9	移動作業に必要な十分な人数の作業者を割り当てる	余っている作業者を優先的に割り当てる。作業者が足りない場合はランダムに割り当てる

表 2: 実験結果

問題	機器数	探索空間 サイズ	総探索 コスト	探索時間 (s)
問題 1	5	10 <sup>203</sup>	129824.8	219.01
問題 2	7	10 <sup>35</sup>	10762.7	145.26
問題 3	10	10 <sup>1062</sup>	126182.4	2635.54
問題 4	12	10 <sup>1007</sup>	35538.7	508.74
問題 5	16	10 <sup>2626</sup>	25760.8	683.53
問題 6	20	10 <sup>9819</sup>	11718.1	2356.35
問題 7	27	10 <sup>5860</sup>	74710.4	6729.16
問題 8	30	10 <sup>8640</sup>	31436.2	1974.41
問題 9	34	10 <sup>5505</sup>	23226.3	3261.27
問題 10	40	10 <sup>14221</sup>	44160.3	4817.30

初期解生成: greedy 法によって初期解を生成する。また、全ての制約の違反点数を 1 に戻す。

制約違反選択: 制約違反をランダムに 1 つ選択する。

解候補採択: 選択されたオペレータを適用し、次ステップの解候補を生成する。

値変更: 最も制約違反点数の少ない解候補の違反点数が、現在の解候補の違反点数以下ならば、その解候補を採択する。

局所最適解判定: 100 ステップの間、違反点数を改善できなかった場合、局所最適解とみなす。

breakout 戦略: 制約違反情報を保存し、それぞれの違反点数を 1 増やす。

### 3.3 オペレータ

図 2(a) のような状況において、1 変数毎の値割り当て変更によって違反を解消する場合を考える。時刻 5、機器 B の OV の値を変更すると (b)、時刻 4 と時刻 5 の間の作業が移動になるが、ターン 4 に作業者が割り当てられていないため、違反が起こる。また、時刻 5 と

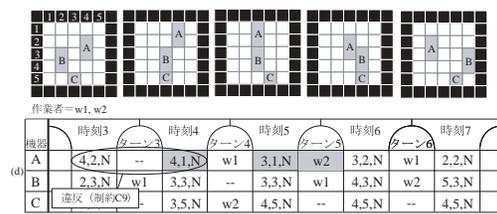


図 3: オペレータによる値割り当て変更の例

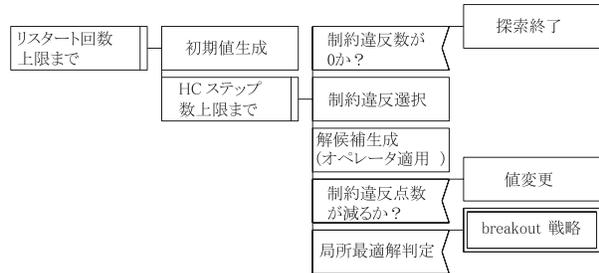


図 4: 探索の処理手順

時刻 6 の間で経路が不連続になるため、違反が起こり、制約違反が増えてしまう。時刻 5 機器 A の OV の値を変更する場合も (c)、時刻 4 と時刻 5 および時刻 5 と時刻 6 の間で違反が起こり、制約違反が増えてしまう。

これに対し、オペレータは次のような値割り当て変更を行う (図 3)。まず、重なっている機器同士を接するように時刻 5 機器 A の OV を変更する。次に、greedy 法により経路の連続性を保つよう前後の変数の値を割り当て直す。これにより、機器 A の時刻 4 と時刻 5 で違反が起こるが、制約違反数が増加しないため、次の状態に遷移することができる。

## 4 実験結果

機器数が 5 個から 40 個の問題 10 問を用いて実験を行った。本実験は、リスタート回数を 10、山登りステップ数の上限を 100,000 として、それぞれ 50 回の実験を行った。表 2 にその結果を示す。探索空間の増加に比べ、処理時間の増加はゆるやかである。

## 5 おわりに

RAFL を CSP として定式化し、breakout 戦略を組み込んだ MCHC により探索を行うシステムを提案した。本システムは、違反の種類に特化したオペレータを用いることで、機器数の多い大規模な問題にも対応することが可能である。

## 参考文献

- [1] 是永基樹, 萩原将文: 対話型進化計算法によるインテリアレイアウト支援システム, 情報処理学会論文誌, Vol.41, No.11, pp.3152-3160 (2000).
- [2] 西原清一: 制約充足問題の基礎と展望, 人工知能学会誌, Vol.12, No.3, pp.351-358(1997).
- [3] Minton, S. et al.: Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problem, *Artif. Intel.*, Vol.58, pp.161-205(1992).
- [4] Morris, P.: The Breakout Method for Escaping from Local Minima, *AAAI*, pp.40-45(1993).