多様な調理環境に応じた調理法変換による複数レシピの並行調理手順作成手法

如澤 駿介[1,a)]　角 康之[1,b)]

概要：家庭で複数の料理を同時に調理する際，レシピは単一料理・特定の調理環境を前提として記述されているため，器具の競合や人手制約への対応は調理者の経験に大きく依存している．本研究では，各調理工程を代替的な調理法へ置換できる仕組みと，家庭ごとの器具台数や作業人数などの資源制約を考慮した並行調理スケジューリングを組み合わせることで，家庭環境に適応した複数レシピの調理手順を自動生成する手法を提案する．提案手法では，レシピ手順を構造化し，工程ごとに所要時間・作業負荷・使用器具・依存関係などの属性を付与した上で，調理法の置換とスケジューリングを統合的に行う．また，ユーザが調理法の選択や環境設定を対話的に操作できるシステムも実装した．効果検証として，コンロ 1 口という制約の厳しい環境下で複数レシピの調理を行い，調理法変換の有無による調理時間と平均作業負荷を比較した結果，調理時間の短縮に加え，人手資源の待機時間が減少し，平均稼働率（作業負荷）が上昇した．

1. はじめに

料理を行う際には，レシピサイトや書籍に掲載された手順を参照することが多い．こうしたレシピは，単一の料理を対象として，投稿者の調理環境や実際に行った手順をもとに記述されている．

家庭での食事準備では，主菜・副菜・汁物といった複数の料理を食卓に並べることが一般的である．そのため，複数の料理を並行して調理する必要があるが，レシピには単一料理の工程のみが書かれているため，どのような手順で複数料理を並行調理することが効率的かは，調理者自身の経験に大きく依存する．

さらに，家庭ごとに調理環境（器具の台数や種類）が異なるため，投稿者と同じ環境が揃わない場合も多い．例えば，パスタを調理する際，麺を茹でる鍋とソースを作るフライパンでコンロが 2 口必要になる場合があるが，家庭によってはコンロが 1 口しかないこともある．このような制約下では，レシピ工程を単に並べ替えるだけでは器具の競合により作業が詰まってしまい，並行調理そのものが成立しない．

実際の家庭では，調理者が自身の知識や経験をもとに，湯沸かしを電気ケトルで行う，パスタ茹でを電子レンジで代用するといった工夫をすることで器具競合を回避している．しかし，こうした代替的な調理法はレシピに明示されることが少なく，やはり調理者の経験に依存する形となっている．

そこで本研究では，家庭の調理環境（調理器具の種類や数）に応じて，レシピ工程の実行順序だけでなく，工程の調理法そのものを代替手段へ置換しながら，複数レシピの並行調理手順を生成する手法を提案する．各工程について代替的な調理法への変換方法を定義することで，同じ調理操作を行う工程を異なる器具や手順で実行できるようにする．さらに，置換後の工程列に対して器具・調理人数・依存関係を制約とした資源制約スケジューリングを適用し，家庭環境に適応した調理手順を自動生成する．次章では，このようなレシピの構造化や調理スケジューリングに関する従来研究について整理し，本研究の特徴について述べる．

2. 関連研究

複数の料理を同時に調理するための手順生成に関しては，従来よりタスクスケジューリングに基づく研究が行われている．Nakabe ら [1] は，複数レシピの調理工程をタスクとして構造化し，調理時間の最小化を目的とした最適調理手順提示システムを提案している．本研究と同様に調理工程をタスクグラフとして扱い，依存関係や器具利用の競合を考慮して並行調理手順を生成する点に特徴がある．特に，DF/IHS に基づく最適化アルゴリズムに対して，調理特有の洗浄タスクやユーザ資源の占有を考慮した拡張を導入し，現実的な計算時間で最適手順または準最適手順を得

1 公立はこだて未来大学
a) s-jyozawa@sumilab.org
b) sumi@acm.org

© 2026 Information Processing Society of Japan.

られることを示している．また，Kimura ら [2] は，ファミリーレストランの厨房を対象とした調理プロセススケジューリング手法を提案している．この研究では，注文ごとに複数の料理が同時に完成することがサービス品質に大きく影響する点に着目し，調理工程をジョブショップスケジューリング問題として定式化している．対象となる調理工程は，コンベアオーブンによる自動加熱工程と，盛り付けや食材配置といった手動操作工程から構成され，工程間の順序制約，時間遅れによる品質劣化，作業場所間の移動時間，オーブン入口のスペース容量など，飲食店特有の複雑な制約を考慮したモデルが構築されている．これらの研究以外にも，様々な手法で調理手順の計算を行っている研究がある [3], [4].

レシピのスケジューリングを行うには，レシピに書かれている手順ごとの依存関係を明らかにする必要がある．そのため，レシピデータを依存関係を明確にした構造化データに変換する研究も数多くある．Pan ら [5] は，レシピの調理手順をより深く構造化することを目的として，テキストと画像を併用したマルチモーダルなワークフロー構築手法を提案している．彼らは 9,800 件以上のレシピに対して，各ステップ間の順序関係や並列可能性を含むワークフローグラフを人手で注釈した大規模データセット MM-ReS を構築し，これを用いて工程間の因果・依存関係を推定するニューラルモデルを学習している．モデルはテキスト記述だけでなく調理画像も入力として活用することで，具材投入のタイミングや熱処理工程の開始など，文章だけでは曖昧になりやすい手順上の関係性をより精度高く推定できることを示した．また，Jermsurawong ら [6] は，レシピに記述された調理手順を構造的に理解することを目的として，RecipeFlow Graph というグラフ表現を提案している．RecipeFlow Graph では，調理ステップをノードとして配置し，レシピ内の手順の流れや依存関係を明示化する．レシピ文から操作語や対象食材を抽出し，ステップ間の前後関係を機械学習モデルにより推定する枠組みを構築している．

本研究も同様に，レシピを各調理タスクとその依存関係からなる構造化データへ変換し，複数レシピの手順生成を行う．しかし，従来研究が主として器具の競合や工程依存関係に基づくスケジューリングに焦点を当てていたのに対し，本研究ではさらに，レシピに明示されていない代替的な調理方法を選択・置換できる仕組みの導入を目指す．これにより，家庭の調理環境の差異に対して，工程の実行順序を調整するだけでなく，工程そのものを再構成することで適応できるようになる．

3. 提案手法

3.1 手法の概要

本研究では，複数レシピを家庭の調理環境に適応させて並行調理するため，レシピ工程の調理法を代替手段へ置換しながら，資源制約下での並行調理スケジューリングを行う手法を提案する．

従来の調理スケジューリング研究では，レシピに記述された工程をそのまま用いて実行順序を最適化する手法が提案されてきた．しかし，家庭ごとに調理環境（器具の台数や種類）が異なるため，工程の実行順序を調整するだけでは調理器具の競合を解消できない場合がある．

本研究では，レシピに記述された調理工程を家庭環境で利用可能な代替的な調理法へ置き換える仕組みを導入する．例えば，鍋でパスタを茹でる工程を電子レンジによる調理へ，コンロでの湯沸かしを電気ケトルによる湯沸かしへと置き換えることで，調理器具の競合を回避する．調理法を変換した後の工程列に対して，調理器具の台数，作業負荷，工程間の依存関係を制約条件としたスケジューリングを適用することで，家庭環境に適応した並行調理手順を生成する．

以降では，まずレシピデータの収集と構造化（3.2 節），調理法の変換（3.3 節），スケジューリングアルゴリズム（3.4 節）について順に説明し，最後に試作システムの実装（3.5 節）について述べる．

3.2 レシピデータの構造化

3.2.1 レシピデータの収集

レシピデータは，レシピ投稿サイトである Cookpad（クックパッド）から収集した．様々な料理カテゴリから約 5000 件のレシピを取得し，各レシピについて料理名，材料，調理手順のテキスト情報を抽出した．

3.2.2 構造化データへの変換

収集したレシピの調理手順テキストを，タスクスケジューリングに必要な構造化データへ変換する．従来研究では，レシピテキストから工程間の依存関係を推定する手法 [5], [7] や，調理時間を予測する手法 [8], [9] が提案されている．しかし，本研究ではレシピの構造化手法そのものには焦点を当てず，大規模言語モデル（LLM）を用いて各調理手順をタスクとして整形した．

図 1 に構造化データの例を示す．この例では，パスタ料理の 16 工程が表形式で表現されている．そして各工程は以下の属性を持つ：

- **id**: 識別番号
- **action**: 調理動作の内容
- **ingredients**: 使用する食材
- **tools**: 使用する調理器具

© 2026 Information Processing Society of Japan.

- **time**: 所要時間（分）
- **labor_units**: 作業負荷
- **depends_on**: 依存する工程の id

labor_units は，調理者がその工程にどの程度張り付く必要があるかを 0.0〜1.0 の値で表す．0.0 は完全に放置可能（例：煮込み），1.0 は完全に拘束（例：混ぜ続ける）される工程を意味する．例えば，図 1 の id=3「パスタを茹でる」は 0.2（ほぼ放置可能），id=9「みじん切りしたニンニクを加えて炒め，香りを立たせる」は 0.8（ほぼ拘束）として定義されている．

depends_on は工程間の依存関係を表し，例えば id=15「茹でたパスタを加えて全体を混ぜ合わせる」のように複数の前提工程に依存する場合は `depends_on=4,14` と表現する．

この方法により，収集した約 5000 件のレシピのうち，現在 37 件を構造化データへ変換した．LLM が生成した時間や依存関係の情報には誤りが含まれることがあったため，すべて人手により確認・修正を行った．

id	action	tools	ingredients	time	labor_units	depends_on
1	鍋に水を入れ、塩を加えて火にかける	鍋, コンロ	水, 塩	0.5	1.0	
2	沸かす	鍋, コンロ		4.0	0.0	1
3	パスタを茹でる	鍋, コンロ	パスタ	9.0	0.2	2
4	パスタをザルにあけて水気を切る	ザル	パスタ	0.5	1.0	3
5	ニンニクの皮を剥き、みじん切りにする	包丁, まな板	ニンニク	1.0	1.0	
6	ベーコンを短冊切りにする	包丁, まな板	ベーコン	1.0	1.0	
7	小松菜をざく切りにする	包丁, まな板	小松菜	1.0	1.0	
8	フライパンにオリーブ油を入れて熱する	フライパン, コンロ	オリーブ油	1.0	0.5	
9	みじん切りにしたニンニクを加えて炒め、香りを立たせる	フライパン, コンロ	ニンニク	2.0	0.8	5, 8
10	短冊切りのベーコンを加えて炒める	フライパン, コンロ	ベーコン	2.0	0.8	6, 9
11	ざく切りの小松菜を加えて炒める	フライパン, コンロ	小松菜	2.0	0.8	7, 10
12	塩・粗挽き黒胡椒で味付けする	フライパン, コンロ	塩, 粗挽き黒胡椒	0.5	0.5	11
13	●（醤油・砂糖・粉末だしの素）を加えて全体を炒める	フライパン, コンロ	醤油, 砂糖, 粉末だしの	1.0	0.8	12
14	火を止める	フライパン, コンロ		0.1	0.1	13
15	茹でたパスタを加えて全体を混ぜ合わせる	フライパン, コンロ	パスタ	1.0	1.0	4, 14
16	仕上げに粗挽き黒胡椒をふる	フライパン, コンロ	粗挽き黒胡椒	0.2	0.5	15

図 1: レシピの構造化データの例（パスタ料理）

3.2.3 調理目的の定義

各調理工程に対して，上記の属性に加え「調理目的（purpose）」を定義した．調理目的とは，その工程が何を達成しようとしているかを示す抽象的なラベルであり，調理法を代替する際の基準となる．例えば，「鍋でパスタを茹でる」と「電子レンジでパスタを調理する」は異なる調理法であるが，どちらも `cook_pasta` という同じ調理目的を持つ．

調理目的は人手で定義した．代表的な調理目的として，`boil_water`（湯を沸かす），`cook_pasta`（パスタを加熱調理する），`mince`（食材を細かく切る）などがある．

3.3 調理法の変換

本研究では，家庭ごとに異なる調理器具の構成に適応するため，レシピに記述された調理工程を別の調理法へ置換する仕組みを導入する．工程そのものを再構成することで，器具の競合を回避し，より効率的な調理手順を生成できると考える．

3.3.1 調理目的による工程の抽象化

前節で定義した purpose 属性を用いることで，異なる調理法であっても同じ目的を持つ工程を統一的に扱える．例えば，「鍋で湯を沸かしてパスタを茹でる」と「電子レンジでパスタを加熱する」は異なる調理法であるが，どちらも `cook_pasta` という同じ調理目的を持つ．この抽象化により，調理目的が同じであれば工程を別の調理法へ置換することが可能になる．

3.3.2 変換方法の定義

各調理目的について，代替となる調理法への変換方法を定義する．変換方法には，代替となる調理法の操作内容，使用する器具，所要時間などを記述する．例えば，`boil_water`（湯を沸かす）については，「鍋とコンロを使う」方法から「電気ケトルを使う」方法への変換を定義する．

図 2 に変換の仕組みを示す．同じ調理目的（purpose）を持つ一連の工程をまとめて，別の調理法へと置き換える．例えば，`cook_pasta` というタグが付いた「湯を沸かす→パスタを入れる→茹でる」という工程列を，「電子レンジで加熱する」という単一の工程へ変換できる．同様に，`boil_water` タグがついた工程を電気ケトルでの湯沸かしへ変換することも可能である．

調理法の変換により，例えば「コンロが 1 口しかない」といった環境制約下でも，電気ケトルや電子レンジを活用することで器具の競合を回避し，並行調理を実現できる．

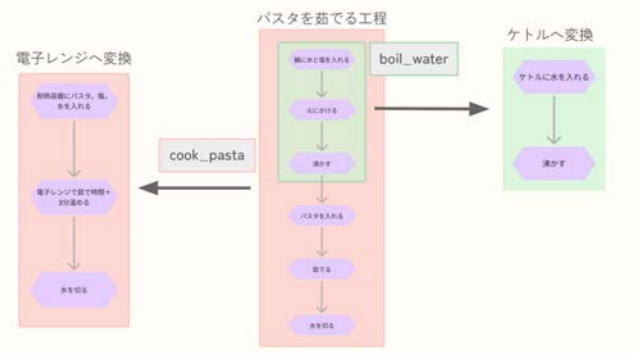

図 2: 調理目的に基づく変換の仕組み

3.4 調理手順のスケジューリング

本研究では，構造化した複数レシピの工程列を入力として，家庭環境に適応した並行調理手順を生成する．

複数レシピの調理を同時に行う場合，各工程の実行順序が適切でないと，調理器具の競合や作業の過負荷が生じ，全体の調理時間が長くなる．そこで本研究では，全体の調理時間の短縮と，調理器具・作業負荷の制約を満たす並行調理手順の生成を目的としたスケジューリングを行う．

3.4.1 クリティカルパスに基づく優先度決定

本手法では，各工程について，その工程から料理完成までにかかる最長の時間（クリティカルパス長）を，工程の

© 2026 Information Processing Society of Japan.

実行状態が更新されるたびに再計算することで動的に算出する．また，複数レシピを統合する場合には，各レシピについて未完了部分のクリティカルパス長を算出する．

3.4.2 リストスケジューリングによる並行実行

依存関係が解消された工程を候補として抽出し，クリティカルパス長に基づき以下の 3 段階の基準で優先度を付与する：

(1) **レシピの残りクリティカルパス長**: 遅延しそうなレシピを優先
(2) **工程のクリティカルパス長**: そのレシピのボトルネックとなる作業を優先
(3) **工程の処理時間**: より時間の長い工程を優先

この順序に従い，以下の資源制約を満たす工程から順に開始する：

- **人手制約**: labor_units で作業負荷を数値化し，現在の作業負荷の合計が上限を超えないようにする
- **器具制約**: 調理器具の利用可能台数の範囲内で工程を実行する

工程終了時に資源状態と依存関係を更新し，この処理を全工程が完了するまで繰り返す．これにより，同時に実行可能な工程を動的に判断しながら並行調理手順を生成する．

図 3 にスケジューリング結果の例を示す．複数レシピの工程が，調理器具や作業負荷の制約を満たしながら並行に配置されている．

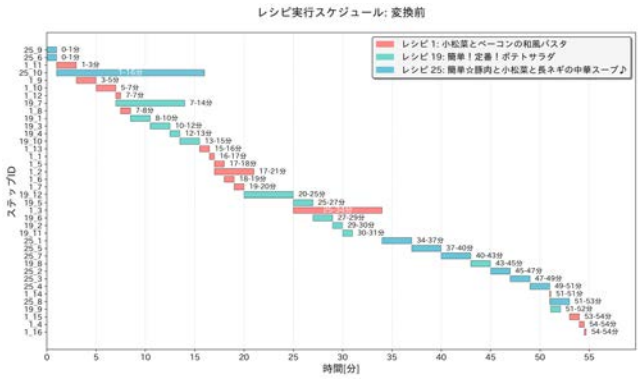

図 3: スケジューリング結果の例

3.5 調理手順提示システム

提案手法を実装し，並行調理手順をユーザに提示するシステムを試作した．システムは調理者が触れるフロントエンド（Vue.js）と，スケジューリングなどを行うバックエンド（FastAPI）から構成される．図 4 にシステムの画面例を示す．以下に主要な機能を説明する．

3.5.1 レシピ選択と調理環境の設定

ユーザはレシピ名の部分一致検索により，調理したいレシピを選択する．次に，家庭の調理環境として，利用可能な調理器具の台数と同時に作業可能な人数を設定する．設定後，元のレシピ工程をそのまま用いたスケジューリングが実行される．

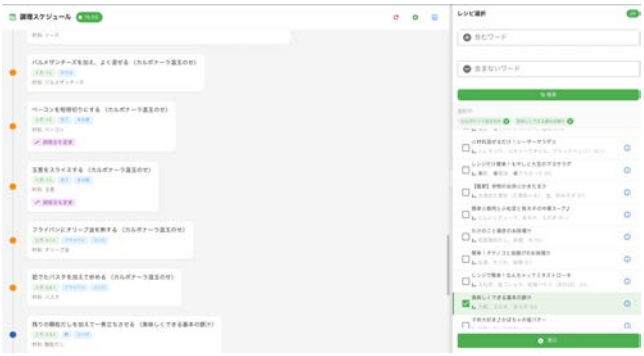

図 4: レシピ選択と調理環境の設定画面

3.5.2 調理手順の提示

スケジューリング結果はタイムライン形式で表示され（図 5），各工程の開始時刻，所要時間，使用器具が時系列で示される．また，フローチャート形式での表示も可能で（図 6），工程間の依存関係を視覚的に確認できる．

図 5: タイムライン表示

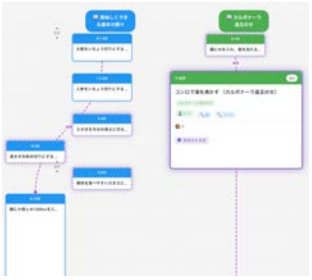

図 6: フローチャート表示

3.5.3 調理法の変換

スケジューリング結果の表示において，置換可能な工程には調理法変更ボタンが表示される．ユーザがこのボタンを選択すると，該当工程に適用可能な代替調理法の一覧が提示される．ユーザが代替調理法を選択すると，該当工程が置換され，再スケジューリングが実行される．

4. 効果検証

4.1 検証方法

提案手法による調理効率化の効果を確認するため，複数レシピの並行調理における総調理時間の短縮と，作業効率（作業負荷の推移）の変化を，予備的な効果検証として整理した．

4.1.1 検証条件

構造化済みのレシピから，主菜・副菜・汁物などを組み合わせた 2 つのパターンを選択した．各パターンについて，以下の調理環境を設定した：

© 2026 Information Processing Society of Japan.

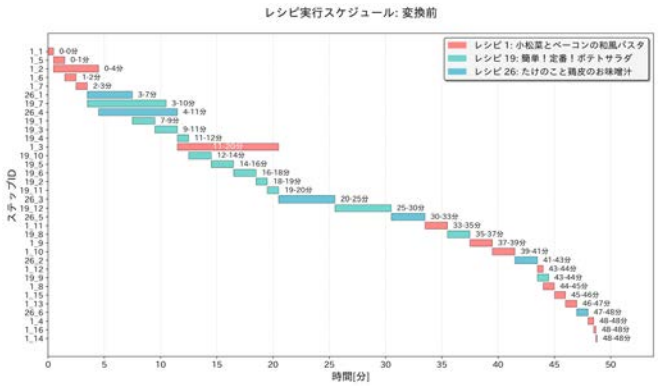

図 7: パターン A（調理法変換なし）

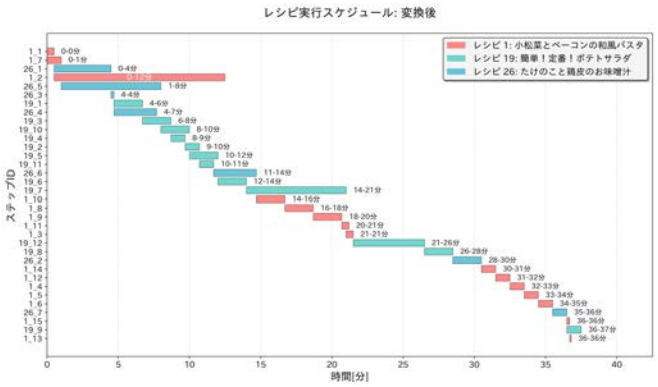

図 8: パターン A（調理法変換あり）

表 1: 各パターンの評価結果

パターン	レシピ構成	変換なし		提案手法		改善率	
		時間 (分)	平均稼働率	時間 (分)	平均稼働率	時間	稼働率
A	和風パスタ+ポテサラ+筍味噌汁	48.8	0.77	37.5	0.91	23.2%	18.2%
B	生姜焼き+かきたま汁+大根の煮物	37.5	0.57	18.7	0.68	50.1%	19.3%

- コンロ：1 口
- 鍋：2 個
- フライパン：1 個
- 作業者：1 人（labor_units 上限=1.0）

各パターンについて，調理法変換なし（ベースライン）と調理法変換あり（提案手法）の 2 通りでスケジューリングを実行し，以下の指標を算出した：

- **総調理時間**：全工程が完了するまでの時間
- **平均稼働率（平均作業負荷）**：スケジュール全体における labor_units の時間平均（作業の詰まり・空白の発生状況をみるための指標）

4.2 結果

図 7 と図 8 にパターン A の調理法変換前後のスケジューリング結果を示す．また，表 1 に各パターンの結果を示す．変換前（図 7）では，パスタの茹で工程とソースの調理がコンロ 1 口で競合し，順次実行となっていた．この競合により，パスタ茹でが完了するまでソース調理を開始できず，全体の調理時間が 48.8 分となっていた．また，平均稼働率は 0.77 であり，調理者は器具の空きを待つ時間が多く発生していた．

一方，変換後（図 8）では，パスタ茹でを電子レンジに変換したことでコンロの競合が緩和され，ソース調理と並行実行できるようになった．この変換により，コンロを用いる工程（ソース調理等）を優先的に配置しつつ，パスタ調理を電子レンジで並行して進められる．結果として，全体の調理時間は 37.5 分となり，変換前と比較して 11.3 分（23.2%）短縮した．また，平均稼働率は 0.91 となり，調理者が継続的に作業可能な状態をより長く維持できた．

パターン B では，生姜焼き，かきたま汁，大根の煮物という 3 品目を対象とした．変換前は 37.5 分であったのに対し，変換後は 18.7 分となり，50.1%短縮した．この差は，大根の煮物をレンジ調理することでコンロの占有が減ったため，フライパン工程（生姜焼き）と汁物を優先的に調理できたためと考えられる．平均稼働率も 0.57 から 0.68 に上昇しており，器具待ちに起因する停滞が相対的に減ったことが示唆される．

両パターンを平均すると，調理時間は 36.7%短縮し，平均稼働率は 18.8%向上した．本稿の範囲では統計的な一般化は行わないが，少なくとも，コンロ 1 口という制約がボトルネックになりやすい状況では，「工程の順序調整」だけでなく「工程自体の置換」によって資源競合を緩和できる可能性がある．

5. おわりに

本研究では，目的に基づく調理法変換と資源制約下での並行調理スケジューリングを組み合わせ，家庭の調理環境（器具台数・人手上限）に適応した複数レシピの並行調理手順を生成する手法を提案した．従来の調理スケジューリング研究では，レシピに記述された工程をそのまま用いて実行順序を最適化する手法が中心であった．本研究では，各工程に調理目的（purpose）を付与することで工程を抽象化し，目的が同じであれば代替器具や代替手順への置換を可能とした．これにより，家庭ごとに異なる器具構成に対して，工程の実行順序だけでなく調理法そのものを調整しながら並行調理手順を生成できる．また，ユーザが調理法

© 2026 Information Processing Society of Japan.

変換を選択し，変換後の手順を対話的に確認できるインタフェースを試作した．

効果検証として，コンロ 1 口という制約の厳しい環境において 2 パターンのレシピ組み合わせを対象に，調理法変換なし／ありの比較を行った．その結果，平均 36.7%の調理時間短縮と，平均 18.8%の平均稼働率の向上が観察された．これは，鍋でのパスタ茹でを電子レンジに置換する，コンロでの湯沸かしを電気ケトルに置換するなどの調理法変換により，コンロの競合による停滞を緩和できたためと考えられる．

今後の課題は，まず，変換ありきのスケジューリングを行えるようにする．現状では変換前の工程列に対してスケジューリングを行った後に置換を適用しており，変換の適用はユーザ選択に依存している．今後は，各工程に付与した調理目的（purpose）に基づいて，候補となる調理法の組合せを探索しながらスケジューリングを行い，置換によって変化する所要時間・使用器具・依存関係をスケジューリングの意思決定に反映する枠組みを検討する．

また，現状の調理法変換は purpose に対する固定的なルール適用に留まっているため，汎用的な変換の導出が課題である．今後は，目的を満たす手順を個別規則として列挙するのではなく，抽象化した手順表現に基づいて多様な具体手順を導出できる仕組みを検討する．

参考文献

[1] Nakabe, J., Mizumoto, T., Suwa, H. and Yasumoto, K.: Optimal Cooking Procedure Presentation System for Multiple Recipes and Investigating Its Effect, *Algorithms*, Vol. 14, No. 2, p. 67 (2021).

[2] Kimura, Y., Shimizu, K., Tsuboi, T., Hasegawa, D., Ishikawa, K., Kimura, K., Tanaka, M., Ozeki, K., Zhou, J. and Shigeno, M.: An Approach to Cooking Process Scheduling for a Family Restaurant, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 12, No. 3, pp. 17–00633 (2018).

[3] Bollini, M., Barry, J. and Rus, D.: BakeBot: Baking Cookies with the PR2, *Interpreting and Executing Recipes with a Cooking Robot*, Springer Tracts in Advanced Robotics, Vol. 88, Springer, pp. 481–495 (2013).

[4] Matsushima, Y., Funabiki, N., Nakanishi, T. and Watanabe, K.: An application of cooking-step scheduling algorithm for homemade cooking and its extensions, *Proceedings of the 1st IEEE Global Conference on Consumer Electronics*, IEEE, pp. 94–98 (2012).

[5] Pan, L., Chen, J., Wu, J., Liu, S., Ngo, C.-W., Kan, M.-Y., Jiang, Y. and Chua, T.-S.: Multi-modal Cooking Workflow Construction for Food Recipes, *Proceedings of the 28th ACM International Conference on Multimedia*, ACM, pp. 1132–1141 (2020).

[6] Jermsurawong, J. and Habash, N.: Predicting the Structure of Cooking Recipes with RecipeFlow Graphs, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 1349–1354 (2015).

[7] 白井圭佑，亀甲博貴，森信介：オープンドメイン手順書のフローグラフ予測，自然言語処理， Vol. 31, No. 2, pp. 479–503 (2024).

[8] Sugioka, K., Kamei, S. and Morimoto, Y.: BERT Pre-training for Cooking Time Prediction from Cooking Recipes, *2024 Twelfth International Symposium on Computing and Networking (CANDAR) Workshops*, IEEE, pp. 157–162 (2024).

[9] 卓也阿部，淳司立間，雅樹青野：D-018 料理レシピサイトから抽出される特徴に基づいた調理時間予測，情報科学技術フォーラム講演論文集（FIT 2015），Vol. 14, No. 2, 情報処理学会，pp. 103–104 (2015).

© 2026 Information Processing Society of Japan.