

## 多種料理の調理順最適化アルゴリズムの提案

松島 由紀子<sup>†1</sup> 船 曳 信 生<sup>†1</sup> 中 西 透<sup>†1</sup>

本論文では、忙しいライフスタイルの人が、効率よく食事を作ることを目的として、多種料理の調理順最適化アルゴリズムを提案する。まず、材料の加工、加熱、鍋の洗浄からなるキッチンモデルを定義する。その上で、各料理の調理順探索問題を組合せ最適化問題として定式化し、シミュレーティッド・アニーリングを用いたアルゴリズムを提案する。Java 言語を用いて本アルゴリズムを実装し、例題として 6 種類の料理を与えて評価を行った結果、実際の調理結果とは約 15 分の誤差が生じたが、多種料理の調理順最適化に有効であることが明らかとなった。

### A Proposal of a Cooking Order Optimization Algorithm of Various Menus

YUKIKO MATSUSHIM,<sup>†1</sup> NOBUO FUNABIKI<sup>†1</sup>  
and TORU NAKANISHI<sup>†1</sup>

In this paper, we propose a *cooking order optimization algorithm of various menus* for busy persons to cook them efficiently. We first define the *kitchen model* composed of the ingredient processing stage, the heating stage, and the pan washing stage. Then, we formulate the cooking order search problem as a combination optimization problem, and present its algorithm using the simulated annealing method. We implemented our algorithm as a Java application, and verified the effectiveness through the simulation and the real cooking of six menus where the time difference between their results is about 15 minutes.

<sup>†1</sup> 岡山大学大学院 自然科学研究科  
The Graduate School of Natural Science and Technology, Okayama University

### 1. はじめに

近年、健康指向の高まりとコスト指向から多くの人が外食を避け、自炊する傾向にある。職場での昼食においても、コンビニ弁当の購入から、手作り弁当の持参に変更する会社員が増えている。また、メタボリックシンドロームや食育といった言葉も、日常で耳にするようになり、ファーストフードからスローフードへと、多くの人が健康な食生活を送りたいと望むようになっている。

一方、共働きの一般化や長時間労働、核家族の浸透といったライフスタイルが影響し、家族の中で食事を作る人の数が減少し、食事に費やす時間の確保が難しくなっている。特に、バランスのよい献立を立てたとしても、これまでに作ったことのない料理が献立に含まれている場合、それらを効率よく作るための調理順を考えるのは容易ではない。更に、料理を任意の順番に作っていくと、最後の料理を作り終えた頃には、最初の料理が冷めてしまうという問題もある。これらの原因により、多くの人が、食事バランスガイド<sup>1)</sup> に示されている食事の望ましい組み合わせや量を達成できていないのが現状である。

そこで本論文では、1 人の調理者が、様々な種類の料理をまとめて、効率よく作れるように、それらの調理順に関する最適化問題を定式化し、シミュレーティッド・アニーリング (SA)<sup>2)</sup> を用いた調理順最適化アルゴリズムを提案する。提案アルゴリズムでは、各料理の調理手順を「切る」、「加熱する」などの作業段階に分解し、それらの作業順を、異なる料理間で適切に並べ替えることで、最適な調理順を探索する。その際、材料をコンロで加熱している間は調理者の手が空くことから、他の作業を並列して行えるという点に着目する。そして、可能な限り同時にすべての料理を完成するとともに、全体の調理時間を短縮することを目指す。

関連研究には、マルチメディアを用いて、初心者でも失敗なく、上級者は更に技術を向上させる、料理支援ソフトウェア HappyCooking<sup>3)</sup> などがある。HappyCooking では、料理のレシピを「卵を割る」「卵を焼く」などの作業タスクに細かく分けた上で、その順序を最適化し、全体の調理時間の最小化とすべてのレシピの終了時刻の差を最小化する。その上で、表示される映像と音声に従って調理することで、正しく調理することを目指す。

以下、本論文の構成を示す。2 では、キッチンモデルの定義と各作業の詳細について述べる。3 では、多種料理の調理順探索問題を組合せ最適化問題として定式化する。4 では、本問題に対するアルゴリズムを提案する。5 で本アルゴリズムの評価と考察を行う。最後に、6 で本論文のまとめを行う。

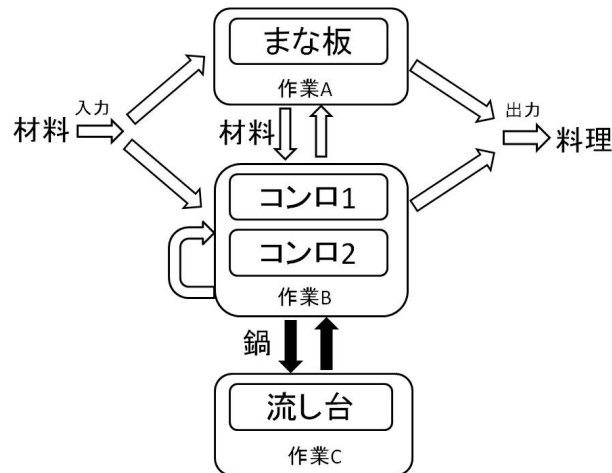


図 1 キッチンモデル  
Fig. 1 The kitchen model.

## 2. キッチンモデル

本論文では、調理手順や器具についての制約を与えるために、図 1 のキッチンモデルを考える。本モデルでの調理者数は 1 である。また、簡単のために、必要な材料は事前にすべて揃っているものとし、野菜などの洗浄作業は考えないものとする。更に、各作業の前後に材料や料理を置くスペースは十分にあるものとする。

### 2.1 作業間の動作

キッチンモデルの入力として、予め料理毎に分けられた材料が、料理毎にまとめて与えられるものとする。また、1つの器具で同時に調理できる料理は 1つとする。

本キッチンモデルにおいて、白い矢印は、材料の流れを表しており、「茹でてから切る」、「切ってから炒める」など、作業 A (まな板) と作業 B (コンロ) のどちらからでも始めることが可能である。また、調理の過程で、「茹でる→切る→炒める」のように、作業 A と作業 B を行ったり来たりすることや、「切る→炒める→煮る」のように、作業 B を繰り返すことも可能である。一方、黒い矢印は、鍋の流れを表しており、使い終わった鍋を必要に応じて洗浄するものとする。

### 2.2 作業 A

作業 A では、1つのまな板で材料の加工 (皮むき、切断など) を行う。作業 A は調理者の手間がかかるため、1つの料理の材料加工を終えるまで、次の料理の材料加工など手間のかかる作業を行うことはできない。以下では、作業 A で行われる材料の加工を“切る”と呼ぶこととする。

### 2.3 作業 B

作業 B では、同性能のコンロ 2 つで材料の加熱 (炒める、煮る、焼く、茹でるなど) を行う。材料の加熱には、調理者の手間がかかる作業と、調理者の手間がかからない作業の 2 種類がある。以下では、便宜のために、調理者の手間がかかる作業を“炒める”と呼び、調理者の手間がかからない作業を“加熱する”と呼ぶこととする。“炒める”では、焦げ付きを防止するために調理者が材料を攪拌する必要があるため、1つの料理の“炒める”作業を終えるまで、次の手間のかかる作業に進むことはできない。一方、煮る、焼く、茹でるなどの“加熱する”作業は、調理者の手間がかからないため、手間のかかる作業 (作業 A や作業 C、作業 B の“炒める”) を並行して進めることができる。なお、“加熱する”作業の途中で材料を数秒間だけ混ぜる作業については考えないものとする。

### 2.4 作業 C

作業 C では、調理を終えた鍋を洗浄する。作業 A と同様、作業 C も調理者の手間がかかるため、作業 C と並行して進めることができるのは、作業 B の“加熱する”のみである。以下では、作業 C で行われる鍋の洗浄を“洗う”と呼ぶこととする。

### 2.5 作業時間

各作業にかかる時間は、各段階で必要となる、料理ごとに分けられた材料が調理される時間とする。ここで例として、カレーを作ることを考える。カレーの材料は、玉ねぎ、人参、じゃがいも、牛肉である。作業 A では、玉ねぎ、人参、じゃがいもの皮むきと切断を行うが、材料の一部である牛肉は加工しないものとする。このとき、玉ねぎ、人参、じゃがいもを加工する時間が作業 A での作業時間となる。牛肉は野菜の加工が終わるまで作業 A で待機し、加工を終えた他の材料と一緒に次の作業へ渡される。

## 3. 問題の定式化

ここでは、多種料理の調理順探索問題を組合せ最適化問題として定式化する。まず、3.1 で本問題に与えられる入力について述べ、3.2 で与えられた入力から導き出されるべき出力について述べる。次に、3.3 で本問題での目的条件について述べる。

### 3.1 入力

本問題の入力は、以下の3つである。ここでは、作業Bでの作業種類、作業時間は、“炒める”と“加熱する”の2種類を区別して与えた。また、鍋の洗浄時間は、どの料理の後でも一律3分とした。

- (1) 料理を食べる人数（食数）
- (2) 鍋の数
- (3)  $n$  個の料理  $V = \{1, \dots, n\}$
- (4) 料理  $i$  の料理名 ( $i \in V$ )
- (5) 料理  $i$  の作業  $j$  での作業種類：切る（作業A）、炒める（作業B）、加熱する（作業B）
- (6) 料理  $i$  の作業  $j$  での作業時間（単位は分）

### 3.2 出力

出力は、各料理の調理作業順  $\sigma$  である。

### 3.3 目的条件

今回は、でき上がったすべての料理をその日の食事としてすぐに食べることを想定し、最初に完成した料理と、最後に完成した料理の、調理終了時刻の差を最小化することを目的とする。以下の評価関数で、 $C_i$  は料理  $i$  の完成時刻（単位は分）を表す。

$$f(\sigma) = \max_i \{C_i\} - \min_i \{C_i\} \quad (1)$$

## 4. アルゴリズム

提案アルゴリズムは、基本的にはSAに基づいており、本章で述べる手順で構成する。

### 4.1 調理の前提条件

まず、本アルゴリズムにおける調理者とコンロの構成について述べる。

#### 4.1.1 調理主体

本問題では、調理者（人）が手間のかかる作業（切る、炒める、洗う）に従事している間、コンロ1・コンロ2で加熱作業（加熱する）を行っている間、調理主体で各料理の調理を進める。そのため、提案アルゴリズムでは、これらの調理主体の手が空いた段階で、その調理主体が次にすべき作業を求めることとしている。

#### 4.1.2 2つのコンロの選択手順

本アルゴリズムでは、次の手順に従って、調理に利用するコンロを選択する。

- (1) コンロ1が空いている場合は、コンロ1を選択する
- (2) コンロ1が使用中でコンロ2が空いている場合は、コンロ2を選択する

### 4.2 前処理

提案アルゴリズムの前処理では、料理を食べる人数（食数）に応じて、調理対象となる各料理の作業時間を計算する。各料理の材料の量は、1人分の量と食数との積になるため、作業Aにかかる時間も食数に比例して増加する。一方、作業Bでは、材料が増えることで鍋の温度上昇の遅れが生じるが、作業Aに比べて、食数による影響は小さい。そこで、食数を  $a$  とした場合、作業Aの作業時間を1人分の場合の  $a$  倍、作業Bの作業時間を1人分の場合の  $(1 + 0.1 \cdot a)$  倍としている。次に、コンロを使用する料理で、“加熱する”、“炒める”を実行する毎に1つの鍋が必要として、 $n$  個の料理を調理するための鍋の使用回数を数える。

### 4.3 料理の順序付け

本アルゴリズムでは、 $n$  個の料理の調理開始順を決めるために、各料理に1から  $n$  の順番を与える。この順番に従って、調理者やコンロで、次に実行開始可能な作業があるかどうかを調べる。まず、初期解ではランダムに順番を与える。2回目以降では、一つ前の解の任意の一箇所について、隣同士の料理の順番を交換することで、新しい解を生成する。

### 4.4 時刻毎の実行開始作業の選択

ここでは、1分刻みで与えられる時刻  $T$  毎に、調理主体である調理者と2つのコンロが実行開始する作業の選択方法について述べる。

#### 4.4.1 調理者の実行開始作業の選択

時刻  $T$  において、調理者の手が空いている場合、次の手順に従って、先頭の料理から順に調べ、その時刻に実行する料理の作業を選択する。

- (1) 以下の3つの条件をすべて満たす最初の料理を探索する。これは、3条件の内、1つでも満たさなければ、時刻  $T$  において、その料理が調理者の作業対象にならないからである。
  - 未完成である
  - 別の作業で調理中でない
  - その料理の開始すべき作業が“加熱する”でない（調理者が不要なため）
- (2) (1) で見つかった料理の作業が“切る”であれば、調理者の作業対象に、本料理の”切る”をセットする。
- (3) (1) で見つかった料理の作業が“炒める”であれば、空いているコンロを探す。
  - (a) 空いているコンロがあれば、調理者とそのコンロの作業対象に、本料理の”炒

める”をセットする。

(b) なければ、次の料理を調べるために、(1)に戻る。

#### 4.4.2 コンロの実行開始作業の選択

時刻  $T$  において、コンロが空いている場合、次の手順に従って、先頭の料理から順に調べ、その時刻に実行する料理の作業を選択する。その際、使用するコンロは 4.1.2 で述べた手順に従って選択する。

- (1) 以下の3つの条件をすべて満たす最初の料理を探索する。これは、3条件の内、1つでも満たさなければ、時刻  $T$  において、その料理がコンロの作業対象にならないからである。
  - 未完成である
  - 別の作業で調理中でない
  - その料理の開始すべき作業が“切る”または“炒める”でない（調理者の手間が必要なため）
- (2) (1) で見つかった料理の作業が“加熱する”であれば、調理者の作業対象に、本料理の“加熱する”をセットする。

#### 4.5 鍋の洗浄開始の選択

時刻  $T$  において、鍋を使う料理の作業が終了した場合、使用済みの鍋を洗浄するために、使用済み鍋カウンタを1増やす。また、使用済みの鍋が発生した場合、鍋を使用する残り回数が用意された鍋の数を超える場合には、今後の鍋の使用のために、調理者は現在行っている調理の手を止めて、使用済みの鍋がなくなるまで、鍋を洗浄する。

#### 4.6 調理の進行

本アルゴリズムでは、1分刻みで与えられる時刻  $T$  毎に、調理の進行を図る。その際、2.3 で述べたように、“加熱する”は調理者の手間がかからないので、“切る”、“炒める”、“洗う”を並行して進めることができる。 $n$  個の料理の調理順を求めるために、すべての料理が完成するまで、以下の(2)から(4)を繰り返す。

- (1) 時刻  $T$  を初期化する ( $T=0$ )。
- (2) 料理が完成した場合、その調理終了時刻を記録する
- (3) 鍋を使う作業が終了した場合、4.5 の手順に従って、鍋の洗浄作業を開始する。また、その作業開始時刻を記録する。
- (4) 調理者またはコンロが空いている場合、4.4 の手順に従って、実行開始作業を決め、その作業を開始する。また、その作業開始時刻を記録する。

表 1 SA のパラメータ  
Table 1 Parameters for SA.

パラメータ	値
初期温度	0.5
温度降下方法	(前回温度) $\times$ 0.9
温度降下回数	100 回
各温度での試行回数	(料理数) $\times$ 100 回

表 2 実行環境  
Table 2 The environment.

項目	仕様
OS	Microsoft Windows XP Home Edition Version 2002 Service Pack 3
CPU	Intel Core 2 Duo CPU E7300 2.66GHz
メモリ	2.67GHz, 2.00GB RAM

#### 4.7 SA のパラメータ

本アルゴリズムにおける SA の各パラメータを表 1 に示す。ただし、ある温度で調理順の更新が一度も起こらない場合には、それ以降の低い温度では解の改善が見込めないとして、本アルゴリズムを終了する。

### 5. 評価

本章では、提案アルゴリズムを Java 言語を用いて実装し、6 種類の料理を与えた例題に対するシミュレーション、および、実際の調理に適用した場合の評価結果について述べる。評価に用いた PC の実行環境を表 2 に示す。

#### 5.1 評価方法

評価例題として与えた 6 種類の料理の作業時間を表 3 に示す。料理を食べる人数（食数）は 4、使用する鍋の数は 3 とした。

#### 5.2 評価結果

ランダムに与えた初期解および本アルゴリズムによる最良解における、料理の調理開始順、その各作業実施順を、それぞれ、表 4、表 5、表 6、表 7 に示す。また、2 つの解での目的関数値を表 8 に示す。表 8 より、アルゴリズムによる調理順で調理する方が、ランダムな調理順に比べて、料理の完成時間差が 46 分短縮されることが示された。

表 3 例題での料理と 1 人分の作業時間 (分)

Table 3 Menus and cooking time as an example.

$i$	料理名	作業	$j = 1$	2	3	4
1	鱈大根	種類	切る	炒める	加熱する	
		時間	1	2	15	
2	小松菜の雪花	種類	加熱する	切る	炒める	加熱する
		時間	3	1	2	8
3	きざみ昆布の炒め煮	種類	切る	炒める	加熱する	
		時間	2	2	10	
4	里芋の煮物	種類	切る	加熱する		
		時間	4	30		
5	味噌汁	種類	切る	加熱する		
		時間	3	8		
6	きゅうりの浅漬け	種類	切る			
		時間	1			

表 4 料理の順序 (初期解)

Table 4 The order of meals(first solution).

$i$	料理名	調理開始時刻	調理終了時刻
1	きざみ昆布の炒め煮	0	23
2	きゅうりの浅漬け	16	20
3	味噌汁	20	43
4	鱈大根	35	56
5	小松菜の雪花	0	61
6	里芋の煮物	53	99

### 5.3 調理結果

ここでは, 5.2 で求めたアルゴリズムによる調理順に従って実際に調理した結果を図 2 示す. 今回の検証で用いたキッチン, 2 で与えたキッチンモデルと同じ環境である.

### 5.4 考察

表 2 の環境において, 表 3 の料理で提案アルゴリズムを実行した結果, 実行時間は平均 0.272 秒であった. よって実行時間は十分短く, 実用的使用に耐えられると言える.

しかし, 例題の 6 種類の料理を実際に調理した結果, 最初に完成した料理 (鱈大根) と, 最後に完成した料理 (小松菜の雪花) の調理終了時刻の差は約 48 分であった. アルゴリズムの最良解の目的関数の値が 33 分であるのに対し, 約 15 分の遅延が発生した原因は, 作業 A で肉や魚を加工した場合に, 今回のキッチンモデルで考慮しなかった, まな板や包丁の洗浄作業が発生し, それ以外の作業を進められなかったためと考えられる. 今後, これら

表 5 調理順 (初期解)

Table 5 The cooking order(first solution).

作業開始時刻	料理・器具名	作業種類	作業時間
0	きざみ昆布の炒め煮	切る	8
	小松菜の雪花	煮る	3
3	鍋	洗う	3
11	きざみ昆布の炒め煮	炒める	2
13	鍋	洗う	3
	きざみ昆布の炒め煮	煮る	10
16	きゅうりの浅漬け	切る	4
20	味噌汁	切る	12
23	鍋	洗う	3
35	鱈大根	切る	4
	味噌汁	煮る	8
39	鱈大根	炒める	2
41	鍋	洗う	3
	鱈大根	煮る	15
44	鍋	洗う	3
47	小松菜の雪花	切る	4
51	小松菜の雪花	炒める	2
53	里芋の煮物	切る	16
	小松菜の雪花	煮る	8
69	里芋の煮物	煮る	30

表 6 料理の順序 (最良解)

Table 6 The order of meals(optimal solution).

$i$	料理名	調理開始時刻	調理終了時刻
1	里芋の煮物	0	49
2	鱈大根	19	40
3	きざみ昆布の炒め煮	28	66
4	味噌汁	36	62
5	小松菜の雪花	0	73
6	きゅうりの浅漬け	65	69

表 7 調理順 (最良解)

Table 7 The cooking order(optimal solution).

作業開始時刻	料理・器具名	作業種類	作業時間
0	里芋の煮物	切る	16
	小松菜の雪花	煮る	3
3	鍋	洗う	3
19	鱈大根	切る	4
	里芋の煮物	煮る	30
23	鱈大根	炒める	2
25	鍋	洗う	3
	鱈大根	煮る	15
28	きざみ昆布の炒め煮	切る	8
36	味噌汁	切る	12
40	鍋	洗う	3
49	鍋	洗う	3
54	きざみ昆布の炒め煮	炒める	2
	味噌汁	煮る	8
56	鍋	洗う	3
59	小松菜の雪花	切る	4
	きざみ昆布の炒め煮	煮る	10
63	小松菜の雪花	炒める	2
65	きゅうりの浅漬け	切る	4
	小松菜の雪花	煮る	8

表 8 目的関数の値

Table 8 The value of the objective function.

解	目的関数の値(分)
初期解	79
最良解	33



図 2 調理結果

Fig. 2 The cooking result.

のモデル化が必要と言える。

## 6. まとめ

本論文では、複数の料理をまとめて効率良く作ることを目的として、SAを用いた多種料理の調理順最適化アルゴリズムを提案した。例題として6種類の料理を与えて評価した結果、ランダムな調理順で調理する場合に比べて、最初と最後に完成した料理の時刻の差が大幅に短縮されることが示された。しかし、実際の調理に適用したところ、15分程度の誤差が生じ、キッチンモデルの改良が必要であることが明らかとなった。それに加えて、今後の課題として、電子レンジ、オーブンなどの考慮、料理の作業時間データベースの構築、他の料理への適用、複数人数での調理への拡張などが挙げられる。

## 参考文献

- 1) 厚生労働省, 農林水産省: 食事バランスガイド. <http://www.j-balanceguide.com/>.
- 2) 柳浦睦憲, 茨木俊秀: 組み合わせ最適化, 朝倉書店 (2001).
- 3) 浜田玲子, 井手一郎, 佐藤真一, 坂井修一: マルチメディア調理支援ソフトウェア「HappyCooking」(2006). 第2回デジタルコンテンツシンポジウム.