

[萌芽発表] 料理の仕上がり温度を考慮した調理工程の動的スケジューリング手法

栢原 悠樹[†] 桑原 寛明^{††} 國枝 義敏^{†††}

[†] 立命館大学院情報理工学研究科 〒 525-8577 滋賀県草津市野路東 1-1-1

^{††} 南山大学情報センター・理工学ソフトウェア工学科 〒 466-8673 愛知県名古屋市昭和区山里町 18

^{†††} 立命館大学情報理工学部 〒 525-8577 滋賀県草津市野路東 1-1-1

E-mail: †is0209kk@ed.ritsumeai.ac.jp, ††kuwabara@nanzan-u.ac.jp, †††kunieda@is.ritsumeai.ac.jp

あらまし 複数品の調理を行うときは1品ずつ調理を行うと調理者の手が空く時間が発生してしまい、効率のよい調理ができない。しかし、複数品の料理を並列に調理するスケジュールの作成は難しい。この課題を解決するためレシピから並列調理スケジュールを作成するシステムが存在する。しかし、既存のシステムではスケジュールの時間と実際の調理時間に差が生まれるとスケジュールが破綻してしまう。また、既存システムでは料理の仕上がり温度が考慮されていないため作った料理が冷めてしまう問題がある。本研究では料理の仕上がり温度に対して優先度を設定し、優先度ベーススケジューリングアルゴリズムを用いた動的なスケジューリング手法を提案する。料理の仕上がり温度に対して優先度を設定することによって温かい料理を最後に作るスケジュールの作成を行う。

キーワード レシピ, 優先度ベーススケジューリング, 仕上がり温度, 時間短縮, 並列調理

Dynamic Scheduling Method of Cooking Process Considering The Finishing Temperature of Dishes

Yuuki KAYAHARA[†], Hiroaki KUWABARA^{††}, and Yoshitoshi KUNIEDA^{†††}

[†] Graduate school of Information Science and Engineering, Ritsumeikan University

^{††} Center for Information and Communication Technology Department of Software Engineering, Faculty of Science Engineering, Nanzan University

^{†††} College of Information Science and Engineering, Ritsumeikan University

E-mail: †is0209kk@ed.ritsumeai.ac.jp, ††kuwabara@nanzan-u.ac.jp, †††kunieda@is.ritsumeai.ac.jp

Abstract When cooking two or more items, if you cook one item at a time, it takes time for the cooker to spare, so you can not cook efficiently. However, it is difficult to create a schedule to cook multiple dishes in parallel. In order to solve this problem, there exists a system for creating a parallel cooking schedule from a recipe. However, in the existing system, the schedule collapses if a difference occurs between the schedule time and the actual cooking time. Also, in the existing system there is a problem that the cooked dishes will cool down because the temperature of the finish of the dish is not considered. In this research, priority is set for the temperature of the finish of the dish, and a dynamic scheduling method using the priority based scheduling algorithm is proposed. We will create a schedule to make a warm dish at the end by setting the priority to the temperature of the finish of the dish.

Key words recipe, priority based scheduling, finishing temperature, time reduction, parallel cooking

1. はじめに

料理レシピは通常複数の調理工程によって構成されており、レシピに定められた順序に従って調理することで料理が完成する。複数品の調理を行うとき、1品ずつ調理を行うと総調理時

間は各料理の調理時間の合計になる。ところが、1品ずつの調理には調理者の手が空く時間がある場合があり、この時間を用いれば調理者は他の作業を行うことができる。このことから調理者の手が空いている時間がある調理は効率的ではない。しかし、複数の料理の調理工程を組み合わせ、並列に調理を行うと

複数品の総調理時間の短縮が可能になる。しかし、そのためには複数の料理の調理工程を組み合わせる調理スケジュールを組み立てる必要がある。調理スケジュールは各工程に必要な時間を目安にして作成する。複数料理の調理工程を組み合わせる調理スケジュールの作成によって総調理時間の短縮を行うことができる。しかし、調理者が初心者であるとき複数の料理を効率よく並列に調理するためのスケジュール(以降 並列調理スケジュール)を考えることは難しい。なぜなら、多くのレシピは調理工程に具体的な調理時間が書かれていないため調理者が工程にかかる調理時間を推定することが難しいためである。さらに、調理時間が判明している場合でも複数の料理の調理工程の組み合わせは膨大であり、並列調理スケジュールを組み立てることは困難である。

ネット上のレシピから並列調理スケジュールを作成する研究として杉本らの研究がある[1]。杉本らは1人の調理者に向けて複数の料理の並列調理スケジュールの作成を行うシステムの開発を行った。そのとき、上述の通りネット上のレシピは工程の時間が不明なものがあり、スケジュールが難しいという課題があった。杉本らの研究では具体的な調理時間が書かれていない工程に対し、各調理工程を作業内容によって分類し、作業内容ごとに調理時間の設定を行うことにより並列調理スケジュールの作成を行っている。実際の調理時間が工程ごとに設定された時間と異なる場合、作成したスケジュールと実際の調理の進み具合に差が発生する。作成されたスケジュールと実際の調理の進み具合との差があると、予め作成したスケジュールは破綻してしまう。そのため、杉本らのシステムは具体的な調理時間が書かれていない工程に対応しているとはいえない。また、杉本らのシステムでは調理時間を最短にするためのスケジュールを提案している。そのため、杉本らのシステムでは温かい料理を先に作ってしまい、温かい料理が冷めてしまうスケジュールが作成されることがある。しかし、現実の調理では温かい料理はできるだけ温かい状態のまま提供されることが望ましい。

また、調理の並列スケジュール作成システムの研究では松島ら[2]による研究がある。松島らのシステムは2人の調理者が分担して作業を行うスケジュールを作成する。松島らのシステムではレシピの調理工程のスケジュール作成だけでなく、手が空いている調理者に対して洗い物のスケジュールを割り当てるスケジュールの作成を行っている。また、連続調理を行ったほうがよい調理工程に対して連続調理を行うように調理工程を選択してスケジュールを行っている。しかし、松島らのシステムでも料理の仕上げり温度を考慮していないため、温かい料理が冷めてしまうスケジュールが作成されることがある。

本研究では複数のレシピから料理が完成したときの温度を考慮した並列調理スケジュールを実際の調理の進行にもなって動的に調整するシステムを提案する。本システムは料理が完成したときの温度によって各料理に優先度を設定する。複数料理の調理工程を組み合わせるスケジュールを行うとき、優先度ベーススケジュールリングアルゴリズムを使用することで料理の仕上げり温度を考慮した並列調理スケジュールの作成を行う。優先度ベーススケジュールリングは優先度の高い工程から順に実

行するスケジュールリングアルゴリズムである。また、本システムでは実際に調理が終了するまでの時間が不確定である調理工程(以降 不確定時間を持つ調理工程)のスケジュールリングを行うために動的にスケジュールの作成を行う。本システムは各工程の調理が終了するごとに再スケジュールを行う。スケジュール作成時では調理工程の調理時間は暫定的に仮時間を設定する。調理工程が終了するごとに実際の調理時間を代入することで各調理工程ごとの調理時間と実際の調理時間の差をなくすることができる。不確定時間を持つ調理工程の調理時間に実際の調理時間が代入されることで不確定時間を確定させ、スケジュールの破綻が発生しないようにすることができる。また、調理状況によって具体的な調理時間が書かれている工程においてスケジュールと実際の調理に差が発生する場合でも調理工程ごとの調理時間が実際の調理時間と等しくなることで差をなくすることができる。

2. 優先度ベーススケジュールリングアルゴリズム

優先度ベーススケジュールリングアルゴリズムとはOSがスケジュールリングを行う対象のタスクにそれぞれ固有の優先度を割り当て、優先度の高いものから順に実行していくアルゴリズムである。優先度が同じタスクがある場合はFCFSスケジュールリングなどを用いてスケジュールの作成を行う[4]。優先度ベーススケジュールリングの方式として静的優先度によるスケジュールリング手法と動的優先度によるスケジュールリング手法がある。静的優先度によるスケジュールリング手法は周期的に起動されるタスクセットに対して用いられ、タスクの生成時に設定された優先度が変化することなくスケジュールリングが行われる[5]。動的優先度によるスケジュールリング手法ではスケジューラが起動される度に優先度の再設定を行い、再設定された優先度によるスケジュールリングが行われる[5]。静的優先度によるスケジュールリングにおいて課題となるのがタスクの飢餓状態である。飢餓状態とは優先度の高いタスクの実行のみが行われ、優先度の低いタスクの実行が長く待たされてしまう現象のことである。動的優先度によるスケジュールリングでは飢餓状態を回避する手法としてエイジング(aging)と呼ばれる手法がある。エイジングは長時間システムが保持しているタスクの優先度を徐々に上げていくことによってもともと優先度が低いタスクでも最終的にシステム内で最大の優先度を持ち、実行されるようになる手法である[4]。

料理の調理工程をスケジュールすることを考えたとき、調理工程には調理時間が不確定であるものがあることからタスクは周期的に起動されるものではない。静的優先度によるスケジュールリング手法を用いた場合、優先度の高い調理工程が先にすべて実行され、優先度の低い調理工程の実行が遅れてしまう。優先度の高い調理工程が全て先に実行されてしまうため、並列調理が行えなくなり、調理時間の短縮ができなくなる可能性がある。動的優先度によるスケジュールリング手法では工程が終了し、スケジューラが呼び出される度に優先度が再設定される。スケジューラが呼び出される度に今まで選択されていなかった工程の優先度が上昇していくことで優先度の高い調理工程のみ

が選択されることがなくなる．これによって料理の優先度を考慮しながら同じ優先度の料理の進具合を近づけることができる．料理の進具合を近づけることで料理の完成タイミングを近づけ、同じ優先度の料理の提供時間をできるだけ近づけることができる．そのため、提案手法では動的優先度による優先度ベーススケジューリングアルゴリズムを利用したシステムの構築を行う．

3. 提案手法

3.1 調理作業

本研究では調理工程を切る・炒める・和える・煮る・おくの5つに分類する．以下に5つの調理工程の定義を示す．

- 切る:ユーザは包丁とまな板を利用して食材の切断や皮剥きを行う．
- 炒める:ユーザはコンロとフライパンを用いて食材が焦げないようにかき混ぜながら加熱する．揚げる作業はここに分類する．
- 和える:ユーザは複数の食材を混ぜ合わせる作業を行う．
- 煮る:ユーザはコンロと鍋を用いて食材を煮込んだり茹でたりする作業を行う．
- おく:ユーザは料理を冷ましたり、味をなじませたりするために料理をおいておく．

上記5つの調理作業について、"煮る"作業と"おく"作業の間はユーザの手間が不要である．"切る"作業はユーザがその作業に集中しなければならず、他の作業を行う余裕はない．"炒める"作業と"和える"作業はユーザの手間が必要な作業であるが、"切る"作業と違い常にユーザが集中して作業を行う必要はない．このため、"炒める"作業と"和える"作業は"煮る"作業と"おく"作業を並行して行えるものとする．

3.2 調理器具

調理作業の実行にはまな板やコンロなどの調理器具が必要となる．一般に調理器具の個数には制限があり、調理器具を無限に利用することはできない．本研究では調理器具としてまな板、コンロ、鍋、フライパンの4つを想定する．3.1節で述べた5つの調理工程と必要な調理器具の関連を表1に示す．表1は各作業を行うとき調理器具が必要な場合は○、不要場合は×がついている．本研究ではレシピに調理器具の指定がない場合は鍋で煮る工程を行い、フライパンで炒める工程を行うこととする．また、本研究において想定する調理環境を表2に示す．

表1 作業分類と必要な調理器具

	まな板	コンロ	鍋	フライパン
切る		×	×	×
炒める	×		×	
和える	×	×	×	×
煮る	×			×
おく	×	×	×	×

表2よりコンロが2つあるため、コンロを使う調理作業は鍋やフライパンなどの調理器具があれば並列調理を行うことができる．コンロ以外の調理器具はそれぞれ1つずつの環境を想定し

ている．そのため、調理工程によっては調理器具の競合が発生するため、その制約を考慮する必要がある．

表2 調理環境

調理者	1人
コンロ	2口
まな板	1つ
鍋	1つ
フライパン	1つ

3.3 仕上がり温度

調理の仕上がり温度を考慮したスケジュールの作成のために動的優先度による優先度ベーススケジューリングアルゴリズムを用いて工程選択を行う．各料理の優先度は料理が温かい料理である場合には低い値、冷たい料理である場合は高い値をつける．温かい料理の優先度を低い値にすることによって工程選択では温かい料理の工程が選ばれにくくなるため、温かい料理がなるべく後に完成するようにスケジュールが組まれる．なお、今回の提案方式では、調理の最後の工程が"煮る"と"炒める"料理は温かい料理であるとする．それ以外の工程であれば冷たい料理とする．

3.4 調理時間

スケジュール作成には各工程の所要時間（以下 目安調理時間）の情報が必要となる．目安調理時間を持たないと調理工程の選択において、4.4にて述べる複数の調理工程のパラメータが同じときに1つの工程を選択することができないためである．不確定時間を持つ工程はレシピに記載されている全体の目安調理時間から時間が書かれている工程の時間の合計を減算し、調理時間が書かれていない工程の数で割ったものを目安調理時間として設定している．目安調理時間を各工程に暫定的に設定することで総調理時間が長いが調理工程が多い料理について、各工程あたりの目安調理時間を短くすることができる．パラメータが同値の場合目安調理時間が短い工程が選択されることから、目安調理時間を短くできた調理工程は他の調理工程よりも優先して選択される．これによって調理工程が多い料理があっても調理工程の多い料理が優先的に選択されることで調理の終了タイミングを揃えることができるようになる．

設定された目安調理時間は調理を行うときの目安であるため実際の調理時間と異なる場合がある．従来のスケジューリングシステムでは目安調理時間と実際の調理時間に差が生じると作成されたスケジュールは破綻していた．しかし、提案手法は工程にかかる時間を計測し、工程終了ごとに再スケジューリングを行うことにより時間差を考慮したスケジューリングを行える．

4. アルゴリズム

4.1 逐次処理によるスケジューリング

優先度ベースアルゴリズムを用いて、料理の仕上がり温度を考慮した複数料理の逐次処理によるスケジューリングを行うときの実行手順を以下に記す．

- (1) 各料理ごとに料理の仕上がり温度による優先度の設定を行う．

(2) 調理可能な工程に料理の優先度を基にした優先度の設定を行う。

(3) 調理工程の優先度が最大の調理工程を選択する。

(4) 2と3の作業をすべての料理の調理が終了するまで行う。

逐次処理のスケジュールでは2と3の作業は1つの調理工程しか選択されていない。2と3の作業において複数の調理工程を選択できるようにすることで並列調理スケジュールの作成を行えるようにする。

4.2 調理工程の並列化

調理工程を並列化するにあたって、各調理工程が並列可能であるか分類する。

3.1節から調理作業の並列可能性を判断する。"煮る"と"おく"の工程ではユーザは他の工程を行える。そのため、"煮る"と"おく"工程は他の全ての工程と並列で作業を行うことができる。ただし、"炒める"と"煮る"はコンロが空いていなければ並列作業を行うことはできない。また、"煮る"と"煮る"は鍋とコンロが空いていなければ並列作業を行うことができない。

表3に作業分類と並列可能性についてまとめた表を示す。表3は行の作業を行うと仮定したとき、列の作業を並列して行うことができるかを表す。が付いている部分は並列作業が可能であり、×が付いている部分では並列作業が不可能であることを表す。がついている部分は調理器具が残っていれば並列作業が可能であることを表す。

表3 作業分類と並列可能性

	切る	炒める	和える	煮る	おく
切る	×	×	×		
炒める	×	×	×		
和える	×	×	×		
煮る					
おく					

4.3 料理の優先度設定

調理の仕上がり温度を考慮したスケジュールの作成のために動的優先度による優先度ベーススケジューリングアルゴリズムを用いて工程選択を行う。各料理の優先度は料理が温かい料理である場合には低い値、冷たい料理である場合には高い値をつける。温かい料理の優先度を低い値にすることによって工程選択では温かい料理の工程が選ばれにくくなるため、温かい料理がなるべく後に完成するようにスケジュールが組まれる。調理の最後の工程が"煮る"と"炒める"料理は温かい料理であるとする。それ以外の工程であれば冷たい料理とする。

4.4 各工程の優先度設定

各料理の次に行う工程(以下 実行待ち状態の工程)についてそれぞれ優先度の設定を行う。各工程の優先度は料理の優先度を基にしたパラメータである。パラメータはスケジューラの起動ごとに料理の優先度の実行待ち状態になってから選択されていない時間と定数の積を加えることで設定される。この計算によって長く選択されていない工程の優先度を上昇させるエイジングを行い、飢餓状態の工程が発生しないようにしている。

また、選択されていない時間を優先度の計算に組み込むことによって各料理の調理の進行具合をできるだけ揃えることが可能である。パラメータ設定中の工程がまな板やコンロなどの調理器具が足りない、もしくは並列作業ができないなどの原因で実行できない場合はスケジューラが起動されたときに保持していたパラメータを記録したあとにパラメータを0(最低値)にすることによって実行不可能な工程を選択できないようにパラメータの設定を行っている。パラメータが記録された後にスケジューラが呼び出された場合、パラメータの計算において各料理の優先度ではなく記録したパラメータを用いて計算を行う。これによってエイジングによって増加したパラメータが初期化されることを防ぎ、増加したパラメータを保ったまま新たな調理工程との比較を行うことができる。

4.5 工程選択

各工程のパラメータ設定をしたあと、工程の選択をする。工程の選択は優先度ベースアルゴリズムとSJFを用いる。各工程のパラメータを比較し、パラメータが最大の工程を選択する。最大値となるパラメータを持つ工程が2つ以上ある場合には工程の目安調理時間が最も短いものを選択する。これはスケジューリングアルゴリズムの1つであるSJFの選択手法を利用している。SJFはタスクの平均待ち時間を最小にできるアルゴリズムである。SJFによる選択を行うことでスケジューラが再度呼び出されるまでの平均待ち時間を短くすることができる。これによって、再スケジュールの回数を増やすことでスケジュール上の空き時間を少なくすることができる。工程選択後は実行待ちの工程で並列実行可能なものの探索を行うために再度スケジューラの呼び出しを行う。4.4節において述べたようにパラメータの計算過程において、工程の実行が不可能であると判断された工程はパラメータを0にする。また、現在選択中の工程もパラメータを0にすることによって選択不可能にする。そのため、並列実行が不可能である場合はすべての工程のパラメータが0になる。全ての工程のパラメータが0になったとき、これ以上並列実行は不可能であると判断し、選択された工程を実行状態にする。パラメータが0でない工程がある場合は工程選択を行うことにより並列実行する工程の決定を行う。これを現在実行待ちの工程すべてのパラメータが0になるまで繰り返し実行する。

4.6 工程の実行

工程の選択が終了した後、選択された工程は実行状態に移る。実行状態の工程は工程の作業に必要な調理器具を確保し、ユーザからの入力を入力待ち状態に移る。このとき、各調理作業に必要な調理器具は表1に示されている。ユーザは選択された工程の作業を行い、作業が終了したときに工程に対応した文字を入力することによって作業を終了させる。

4.7 工程終了の処理

工程が終了した料理は次の工程に進む。次の工程に進んだ後はスケジューラを呼び出すことで次のスケジュールを作成する。新しい工程は選択されていない時間が0であるためパラメータは料理の優先度と等しくなる。既に選択されて実行状態であるが終了していない工程はパラメータを0とする。また、すべて

の工程が終了した料理は工程のパラメータを 0 にする．4.4 節から 4.7 節までの工程をすべての調理が終了するまで行うことでスケジュールが完成する．

5. 実験

本研究における評価実験ではレシピの目安調理時間の合計と提案システムによる総調理時間の差を計測する．提案システムにおいて全ての料理が目安調理時間で終了したときに提案システムによって総調理時間をどれだけ短縮できたのかを計測，比較する．また，温かい料理が完成してから全ての料理の調理を終えるまでの時間を計測する．これにより，提案システムにおいて温かい料理の終了時間を近くすることができたかの調査を行う．この実験では味の素のレシピサイト [6] から入手した 60 品のレシピから 3 品を選び， ${}_{60}C_3 = 34220$ 通りのパターンについて実験する．

5.1 実験結果

実験において，提案システムが温かい料理と判定した料理は 35 品，冷たい料理と判定した品は 25 品であった．このとき，5 品の温かい料理と想定される料理が冷たい料理であると判定された．これは最後の調理工程が"切る"などの加熱を行わない工程であったため冷たい料理であると判定されたためである．後述の表 4 と表 5 では誤って分類されたデータを用いている．

表 4 に温かい料理と冷たい料理の組み合わせと各々の場合の総調理時間からの短縮時間の平均を示す．表 4 より温かい料理のみや冷たい料理のみを組み合わせた場合よりも温かい料理と冷たい料理を組み合わせたときのほうが平均して多くの時間を短縮できていることがわかる．温かい料理のみの組み合わせの場合，調理工程の実行のためにコンロや鍋などの調理器具を必要とする工程が多く，調理器具の解放を待つ必要のある工程が多かったことが平均短縮時間を増やせなかった原因と考えられる．冷たい料理のみの場合，"切る"，"和える"などの並列調理を行うことができない工程が多く，並列調理を行える箇所が少なかったことが平均短縮時間を増やせなかった原因と考えられる．また，表 5 に温かい料理と冷たい料理の組み合わせと最初の温かい料理が完成してから全ての料理の調理が終了するまでの平均経過時間を示す．表 5 より温かい料理が完成してから平均して 10 分以内に全ての料理を終了させることができている．全パターンにおける総調理時間の平均が 52.6 分であり表 4 の情報からシステムを使わない場合の総調理時間の平均は 65～70 分となる．このとき 1 品の平均調理時間は 21～23 分となることから調理の順番によっては温かい料理が完成してから 22 分ほど待たされる可能性があったため，仕上げ温度に優先度をつける手法によって温かい料理を比較的温かい状態のまま他の料理を完成させるスケジュールの作成ができているといえる．

6. まとめ

本研究では料理の完成時の温度に対して優先度を与えることで料理の仕上げ温度も考慮した調理工程の動的なスケジューリング手法の提案，実装を行った．各料理について最後の調理工程の作業が加熱する工程であるかで料理の仕上げ温度を判

表 4 実験結果 短縮時間

温かい料理	冷たい料理	平均短縮時間
3 つ	なし	7.45 分
2 つ	1 つ	18.1 分
1 つ	2 つ	14.8 分
なし	3 つ	9.17 分

表 5 実験結果 温かい料理完成から終了までの時間

温かい料理	冷たい料理	平均経過時間
3 つ	なし	8.51 分
2 つ	1 つ	5.34 分
1 つ	2 つ	0 分

断することによって優先度を設定し，調理工程に動的な優先度を与えることで料理の仕上げ温度によるスケジューリングを行った．

実験の結果，最も平均の短縮時間が短かった温かい料理を 3 品選んだ場合でも平均して 7 分時間短縮を行うことができることが確認できた．また，提案システムでは冷たい料理が完成してから温かい料理を完成させ，温かい料理の完成時間をできるだけ近くしたスケジュールの作成を行うことができている．しかし，料理の仕上げ温度の判定が誤って行われているといった課題が発見された．この誤判定によって，実際には温かいはずの料理を先に完成させてしまうスケジュールを作成してしまい，できるだけ温かい状態の料理で料理を提供することができなかった．今後の課題として，機械学習やパターンマッチングを用いることで料理の仕上げ温度の判定を向上させる必要がある．

文献

- [1] 杉本和香奈, 佐藤哲司: 既存レシピを活用した並行調理スケジュール法の提案と評価, DEIM Forum 2012, 2012.
- [2] Yukiko Matsushima, Nobuo Funabiki, Toru Nakanishi. "A Proposal of Cooking Model and Cooking Step Scheduling Algorithm for Multiple Dishes". IMECS 2011. vol.1, P127-p131, 2011.
- [3] 高田弘章, 岸田昌巳, 宿口雅弘, 南角茂樹: リアルタイム OS と組み込み技術の基礎, TECHI, Vol.17, p27-28, 2003.
- [4] 大久保英嗣: オペレーティングシステムの基礎, p35, サイエンス社 (1997).
- [5] 藤倉俊幸: リアルタイム/マルチタスクシステムの徹底研究, TECHI, Vol.15, p184-p185, 2003.
- [6] "レシピ大百科", 味の素, <http://park.ajinomoto.co.jp/recipe>, 2017.8.28 参照.