

# エージェントベースモデリングによる 物流倉庫のオペレーション最適化

○深井颯 出口弘 (東京工業大学)

## Optimization of logistics warehouse operations by Agent Based Modeling

\* S. Fukai and H. Deguchi (Tokyo Institute of Technology)

**概要**— 近年、日本においては人口減少と連動して物流の現場における作業人員の減少が問題となっている。その一方でECサイトの発展などにより、物流の現場が日々受ける注文の幅は広がり続けている。物流倉庫ではこの両方の問題に対応すべく、様々なオペレーション法を開発し、組み合わせることによって業務をこなしてきた。一方でこうしたオペレーション法の管理のノウハウは現場においてのみ蓄積されており、俗人的スキルとなってしまっている。しかし新たに倉庫を新設する場合や、それまでにないオペレーション方式を導入する場合にはこうした既存の人的リソースを活用するのは難しい。そこで本研究では、物流現場におけるオペレーションをエージェントベースでモデリングすることにより、最適なオペレーションの客観的な議論を可能にした。

**キーワード:** 最適化, 物流倉庫, エージェントベース, モデリング, ABS

### 1 はじめに

近年ECサイトなどの発展により、物流倉庫の処理する注文の性質の多様化が著しくみられる。例えばto Bの注文かto Cの注文かだけでも、注文の含む量や種類などの性質は大きく異なる。こうした多様な注文性質の幅に対応するため、物流業界ではいくつかもオペレーション形態が開発され、現場ではこれらを組み合わせることによって日々の業務を行っている。

一方でどのような注文にはどのオペレーション形態を用いるのが適しているかは現場の判断に依存しており、定量的な議論はあまりされていない。そこで本研究では、実際の物流倉庫のご協力のもと、実際の物流倉庫をエージェントベースにモデリングする。またそのモデルをフィッティングし、シナリオ比較を行うことで、注文に応じた最適なオペレーションの提案を実現する。

### 2 先行研究

倉庫のオペレーションをエージェントベースな手法で最適化するアプローチ自体は一般的であり、これまでいくつかの提案・実証例が存在する。ただ一方でこれら従来研究では、モデルのフィッティングで用いられる倉庫のデータが十分に得られることを前提としているという問題がある。従来の研究では、ある1つの倉庫或いは実験的な倉庫をオペレーション最適化の対象としていたため、その倉庫に大量の計測装置を取り付けたり、オペレーションに介入することによって大量のデータを得ることが可能であった。一方で、多くの倉庫においてオペレーションを最適化することができるような汎用モデルを考える場合は、この大量のデータを得られること前提としたアプローチは現実的ではないと言える。複数の倉庫にモデルのフィッティング・最適化を試みる場合には、それぞれの倉庫でのデータが必要となる。しかしすべての倉庫において大量の機器を取り付けるのは難しく、また日々のオペレーションを行っている倉庫においてオペレーションに介入するのは困難である。そこで本研究では、少量のデータからエージェントベースに構築した物流倉庫モデルをフィッティングし、最適なオペレーションを導くアプローチ法を提案する。

### 3 アプローチ法・実行方式

前述のように、物流倉庫では多様な注文形態に対応するために様々なオペレーション法が開発されている。その中で今回最適な組み合わせ方を検討するオペレーション法は、“導入コストが比較的安価な”種まきアプローチ”と”摘み取りアプローチ”である。これらはオペレーションの仕方を提案する上位の概念で、それぞれ下位にそのアプローチ法を実現する実行方式が存在する。それぞれのアプローチ法と実行方式、用いられる場面の対応を Table 1 で示す。

Table 1: 各オペレーションと実行方式。

オペレーション概念	実行方式	用いられる場面
種まきオペレーション	1PCS方式	to C, 1注文に1商品
	トータル方式	to BとtoC, 1注文に2商品以上
摘み取りオペレーション	オーダー方式	to B, 1注文に多量の商品を含む場合

以降では、この2つのアプローチ法とそれぞれに対応する実行方式の詳細を説明していく。

#### 3.1 種まきアプローチ

種まきアプローチは、注文に関係なく距離が近い商品をまとめてピッキングするという考え方である。例えば、注文A, B, Cが存在し、 $O=A^{\wedge}+B^{\wedge}+C^{\wedge}$  ( $A^{\wedge} \subset A, B^{\wedge} \subset B, C^{\wedge} \subset C$ ) に含まれる商品同士は距離が近く、同様に  $P=(A^{\vee})^{\wedge}+(B^{\vee})^{\wedge}+(C^{\vee})^{\wedge}$  に含まれる商品同士も距離が近いとする。このとき種まきアプローチの概念に従うと、Oには別々の注文の商品が含まれているが纏めてピッキングを行い、同様にPも纏めてピッキングを行う。このアプローチ法を用いると、距離が近い商品同士をまとめてピッキングすることができ、歩行距離を抑えられるという利点がある。一方で後述するように、注文に関係なくまとめてピッキングを行っているため、注文ごとに再分配の作業が必要になるという欠点が存在する。再分配については、以降で述べる摘み取りアプローチを実現する実行方式の中で詳しく説明する。

#### 3.1.2 1PCS方式

1PCS方式は、1つの注文に対し商品が1つしか含まれていない注文のみに対して、種まきアプローチを用

いる場合の手法である。作業者はピッキングリスト (n 個の商品が記されているとする) に従い、n 個の商品を回収する。その後、各商品はそれぞれ単一の注文に結びついているため、手作業で商品 1 つ 1 つを各々段ボールに積み替える作業を行う。これが前述の、再分配と呼ばれる作業である。作業のフローと概念図を Fig. 1 及び Fig. 2 で示す。

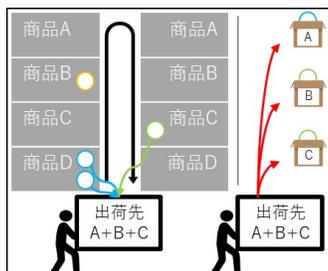


Fig. 1: 1PCS 方式の概念図。



Fig. 2: 1PCS 方式のフロー。

### 3.1.3 トータル方式

トータル方式は、1 つの注文に対し商品が 2 つ以上含まれる場合に対して、種まきアプローチを用いる場合の手法である。前述と同様の A, B, C の注文があった場合においてトータル方式を適用する場合を考える。まず作業者は種まきアプローチの概念に従い、互いに商品が近い O と P をそれぞれ別々にピッキングすると注文 A, B, C の荷物は O と P に混在してしまう

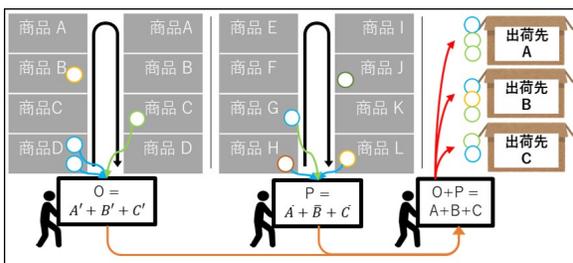


Fig. 3: トータル方式の概念図。



Fig. 4: トータル方式のフロー

ため、荷合わせという工程で  $R=O+P$  を作成する。その上で最後に、R から注文 A, B, C への再分配を行い、梱包する。このとき再分配には SAS (Shutter Assortment System) と呼ばれる機械的な仕分けシステムを用いており、効率化されている。作業の概念図とフローを Fig. 1 及び Fig. 2 に示す。

### 3.2 摘み取りアプローチ

摘み取りアプローチは、注文ごとに別々にピッキングを行うという考え方である。例えば、S, T, U という注文があった場合を考える。作業者はまず注文 S に含まれる商品をまとめて回収し、その次に注文 T の商品を回収、最後に注文 U の商品を回収する。このアプローチではピッキング完了時点で商品は注文ごとに分かれているため、種まきアプローチと異なり再分配の必要がないため時間が短縮される。一方で注文をまとめず 1 つ 1 つピッキングを行うためその分の行き帰りの歩行距離等が余計にかかるという欠点が存在する。以降で摘み取りアプローチを実現する実行方式の解説を行う。

#### 3.2.1 オーダー方式

オーダー方式は、SKUが多い商品が多い注文に対して用いられる。なおSKU (Stock Keeping Unit) とは、注文における1つの商品あたりの数量のことである。SKUが多ければ移動の必要が少なくなるため歩行コストが下がる一方で、種まきオペレーションを用いると再分配コストが大きくなってしまふ。このためSKUが多い場合には摘み取りオペレーションを実現するオーダー方式が用いられる。

いま、A, B, C という注文がある場合を考える。オーダー方式ではまず、前述の摘み取りオペレーションの概念に従い、それぞれの注文に含まれる商品を別々にピッキングする。そして最後に、それぞれピッキングした商品をそのまま段ボールに移すことで、作業を完了する。ピッキングを独立に行っているため、種まき方式のように再分配の工程が必要ないのが特徴である。作業の概念図とフローを Fig. 5 及び Fig. 6 で示す。

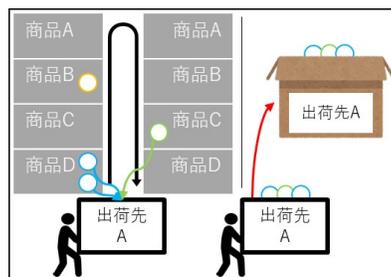


Fig. 5: オーダー方式の概念図



Fig. 6: オーダー方式のフロー

## 4. モデルの構築

本研究ではモデルを、1回のピッキング工程を処理する”ピックモデル”と、ピックモデルを用いてピック工程を含めすべての工程を処理する”全体モデル”の2種類を構築した。なおここで言うピック工程とは、各実行方式の作業フロー内で示されているピッキングの工程のことである。全体モデルでは、各実行方式に従って、ピッキングを含めるすべての工程を、ピックモデルを内部に含めつつ再現する。なお、モデルの入力はピッキングリスト群と倉庫形状情報、出力はそのピッキングリスト群をその倉庫で処理した場合にかかる各工程での所要時間である。

## 5. モデルのフィッティング

はじめにでも述べたように、本研究では従来研究のように大量のデータ取得を前提としないことで、多くの倉庫にフィッティングし最適オペレーションを提案できるような、汎用モデルの提案を目指す。大量のデータ取得を前提としないのは、多数の倉庫で同じようにIoT機器から得られる大量のデータが得られると考えるのは現実的ではないためである。実際の物流倉庫では、日々の業務に関連する情報は蓄積されているが、それ以外の情報はあまり望まず、また実際に業務を行っている倉庫でのIoT計測機器の導入や人的計測などは難しいのである。そこで本研究では、一般的な倉庫で蓄積されているようなデータをベースに、少量のデータから、モデルの汎用活用可能なフィッティング方法を提案する。以降ではまず、少ないサンプルからモデルをフィッティングするために必要なデータの前処理について説明する。そしてその次に、ピッキングモデルと全体モデルについて、それぞれをフィッティングするためのいくつかのアプローチを提示したうえで、それぞれを適用すべき場面を示す。

### 5.1 データの前処理

少量の、現場に蓄積されているデータからモデルのフィッティングを行う場合、2つの問題が発生する。まず1つ目は、情報不足でモデルが収束しない問題。そして2つ目は、対応するデータが存在しないという問題である。以降ではこれら2つの問題について、それぞれの問題を詳細に説明したうえで、解決方法を提案する。

#### 5.1.1 区分けによるフィッティング

少ないサンプルで今回のような多くのパラメータを含むモデルをフィッティングする場合には、大量のサンプルが得られるときにはみられなかった問題が発生する。それは、制約条件が少ないため解空間が広くなり過ぎ、各パラメータが収束しないというものである。大量の正解データがあれば、それらのどのデータでも正しい出力ができるよう各パラメータが影響しあって値を変えていく。その一方で少量の正解データからパラメータをフィッティングしようとする、正解を導くことのできるパラメータの範囲が広がってしまい、値が収束しない。本研究ではこの問題を、データの区分けによるフィッティングを用いることで改善を試み

た。

前述のような問題は、1つの正解に対して大量のパラメータが存在する場合において、より顕著にみられる。これは、1つの正解に対してパラメータが多いと、パラメータの解空間も広がるためである。そこで本研究では、正解データを、モデルの出力としている”全工程での総処理時間”ではなく、”各工程ごとの処理時間”とした。例えばピッキングリストが与えられたときの正解データは、前者だと、そのピッキングリストを処理するのにかかる合計の処理時間である。それに対し後者では、正解データは、準備工程・ピック工程・梱包…それぞれで各x時間、という形になる。この場合、各工程でのデータが対応するパラメータの数は工程の分だけ少なくなるため、パラメータが収束しやすくなると考えられる。なお、このときの区分けは、Fig. 2, 4, 6で示されるような、その倉庫におけるオペレーションの区分けに従う。

#### 5.1.2 データの比率化

前述のように、サンプル数が少ないことによる収束の困難さを緩和するため、本研究では正解データの各工程での区分けを行う。物流倉庫においては業務管理の観点からこうした各工程での所要時間が保持されていることが多く、追加のデータ収集が必要ないという利点がある。ただ一方で、こうした各工程での所要時間データを得た日のピッキングリスト群が存在するかは異なる問題である。

一般の倉庫で蓄積されているデータは、研究目的ではなく業務改善の一環で取得されたものであり、別々の日、きっかけで別個に取得されている場合が多い。そして、そのような別個に取得されたデータをパラメータフィッティングでそのまま用いることはできない。例えばピッキングリストと所要時間のデータがあったとしても、それが別個に取得するものであれば、そのセットを正解データとするのは明らかに誤りである。そこで今回、正解データとなる所要時間を絶対的な時間ではなく、各工程の相対的な比にすることによってこの問題を解決した。このときのイメージをFig. 7に示す。



Fig. 7: 所要時間の比率への変換

### 5.2 ピッキングモデルのフィッティング

この節では、前節で前処理を行ったデータを用いたモデルのフィッティングを行う。ただ、少量のデータからフィッティングを行うことから、従来のようにすべてのデータをモデルに入力することによる探索的なフィッティングのみでは望んだ結果が得られない。このことから本研究では、トップダウンとボトムアップの、2種類のフィッティングアプローチを導入し用いた。以降ではこれら2つのフィッティングアプローチについて解説を行う。またその上で、今回のモデルを構成

するピックモデルと全体モデルが、それぞれどちらのアプローチに適しているのかを示していく。

### 5.2.1 トップダウンなフィッティング

トップダウンなフィッティングとは、従来の用いられている、正解データを与えることによる探索的な求解のことである。今回の場合正解データとは、前述の工程別の時間比である。正解データの工程別時間比と、モデルから得られる工程別時間比が一致すれば、モデルが倉庫内の作業を十分表現できていると言える。

トップダウンなパラメータフィッティングは、性質の異なるデータが複数得られる場合において用いるべき手法である。いま、正解データに含まれている1組のピッキングリスト（入力）と所要時間比（出力）のセットを考える。今回のように多数のパラメータを含むモデルのフィッティングでは、このセットを再現できるようなパラメータの組み合わせはいくつか存在する場合が多い。ここから正しいパラメータに更新を収束するには、性質の異なるデータが必要である。似たデータが複数あってもパラメータの組み合わせは絞られず、正しいフィッティングは望めない。反対に性質の異なるデータが存在すれば、そのデータにおいても正しい出力ができるようパラメータを更新するため、組み合わせが絞られていく。本研究では、全体モデルにおけるデータは注文の内容により性質が大きく異なるため、トップダウンなアプローチを採用した。これについては後述する。

### 5.2.2 ボトムアップなフィッティング

ボトムアップなフィッティングとは、1つの正解データから数理的にパラメータを求める手法のことである。今回のように実際の作業とパラメータが結び付けているモデルでは、現場でパラメータに対応する作業を実測値として計測することが可能である。これは過去のデータではなく改めてのデータ収集が必要であるが、作業時間を1回ストップウォッチ等の機器で計測するのみで済むため、容易に実行が可能である。

ボトムアップな手法を用いるのは、トップダウンの場合とは反対に、似たような性質のデータしか得られない場合である。例えばピッキングモデルにおいて、各商品は多くの場合1個しか注文されず、必然的に収集されるデータは類似する。このような場合モデルの探索的なパラメータチューンは上手く収束しないため、ボトムアップなパラメータチューンが必要となる。

## 6. フィッティング結果

本研究では、これまで説明した加工データ・フィッティング手法を用いて実際にモデルのフィッティングを行った。なお倉庫で得られるデータの制約から、摘み取りアプローチのフィッティングにはオーダー方式のデータ、種まきアプローチのフィッティングでは1PCS方式のデータを用いた。以下ではその結果を、全体モデルでのフィッティングと、ピックモデルのフィッティングに分けて示す。

### 6.1 全体モデルのフィッティング

前述のように全体モデルには、トップダウンなフィッティングを用いた。その結果をFig. 8に示す。なお内輪がデータから算出した各工程ごとの作業時間の割合

（正解データ）で、外輪がそれに合うようフィッティングを行ったモデルの出力する工程別作業時間比（モデルの出力）である。

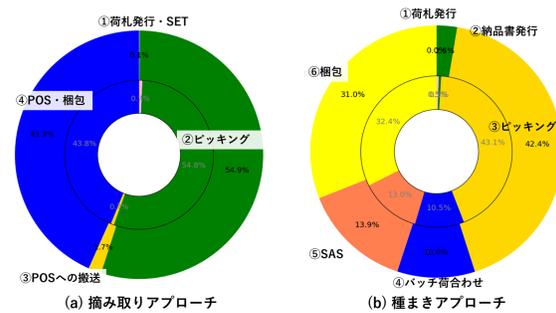


Fig. 8: 全体モデルのフィッティング結果

これより、モデルは一定の精度でフィッティング出来ていることが分かる。一方で種まきアプローチでの②荷札発行のように誤差がみられる工程も存在する。これは、摘み取りアプローチと種まきアプローチ間で共通する作業では同じパラメータを用いており、これによってパラメータの値がどちらかのアプローチに寄ってしまったことによると思われる。このことより、作業の能率を表すパラメータには一定の振れ幅があり、特に異なるアプローチ法間では同じ作業であっても、商品の形状特性など違いから処理時間に違いがでるものと思われる。

### 6.2 ピッキングモデルのフィッティング

ピックモデルには、ボトムアップなフィッティングを用いた。ボトムアップでは実測値と理論値を用いてパラメータの値を決定するが、その結果が以下のFig. 9である。なお、表のデータには仮想のものを使用している。

計測時間 リスト内容	ピッキング場所	PCS数	商品選別	移動時間	他	合計・算出		I move_conf		find_param_pick_param	
						合計(実測値)	算出(理論値)	算出(理論値)	算出(理論値)	固定値	差分
商品 4F A	15	142	250	10		250	506.42	0.49	9.53	2.10	7.43
商品 2F B	16	165	250	11							
商品 2F C	15	72	150	9		400	539.75	0.74	7.65	2.10	5.55
SAS 4F D	18	300	450	15		450	823.19	0.55	16.67	2.10	14.57
SAS 2F E	12	310	300	9							
SAS 2F F	20	401	530	20		830	1432.87	0.58	22.22	2.10	20.12

Fig. 9: ピッキングモデルのフィッティング結果

## 7 シナリオ比較

本研究の目的は、少量の現場データからフィッティングした倉庫モデルを用いて最適なオペレーションを検討することであった。そこで本章では、6章で解説したデータの前処理・フィッティング法を適用したモデルを用いて、様々なシチュエーションのシミュレーションを行った。

### 7.1 実行方式間の比較

モデルのシミュレーションを用いて、ある注文群に対して種まきアプローチを用いるべきなのか種まきアプローチを用いるべきなのかを考える。モデルに入力する注文群は、現場では種まきアプローチを適用している注文群とする。なおこの注文群の、各注文ごとのSKUの性質を、Fig. 10及びFig. 11に示す。

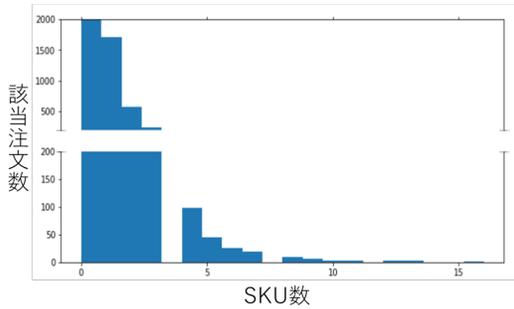


Fig. 10: 注文1つに含まれるSKU数

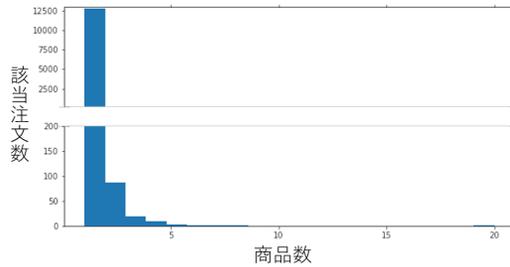


Fig. 11: SKU1つに含まれる商品数

まずこの注文群を、種まきアプローチを採用したモデルに入力し、各工程での所要時間を得た。なおこのとき、オペレーションの実行方式は、商品のSKUが1以上であるのでトータル方式とする。また同様に、注文群を、摘み取りアプローチを採用したモデルに入力し、各工程での所要時間を得た。このときのオペレーションの実行方式は、オーダー方式である。今回注文群に採用しているのは現場で種まき方式を採用しているものであるから、前者の出力する所要時間の方が短ければ、モデルの妥当性が示される。結果をFig. 12及びFig. 13に示す。なお、Fig. 12はピック工程内での所要時間の割合、Fig. 13は全行程での所要時間割合である。

ピック工程での所要時間については、種まきオペレーションがより短かった。この差は主に、“歩行”と“折りコンをパレットに置く”という工程により発生している。まず“歩行”について考える。3章でも説明したように、種まきオペレーションは複数の注文をまとめることによって歩行にかかる時間を削減していた。これに対しオーダーオペレーションでは、注文ごとにスタート地点に戻る必要があるため、必然的に歩行の所要時間が延びる。こうした特性の違いにより、種まきオペレーションの方が歩行時間が短くなっていると考えられる。次に“折りコンをパレットに置く”という工程についてだが、この工程はパレットをゴール地点に置くための歩行移動の時間も含んでいる。この作業の1回あたり時間はピッキングを終了した地点からゴールまでの距離に依存するため、種まきオペレーションと摘み取りオペレーションの間に差異はない。しかし、この作業を行う回数は、商品をまとめてピッキングを行う種まきオペレーションより、注文ごとにピッキングを行う摘み取りオペレーションの方が多い。この発生回数の差異が、“折りコンをパレットに置く”

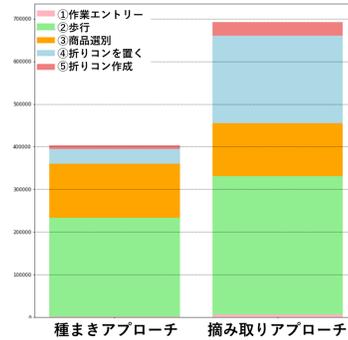


Fig. 12: ピックモデルによる出力

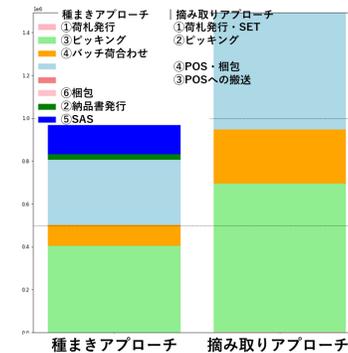


Fig. 13: 全体モデルによる出力

工程での所要時間の差異に反映されていると考えられる。

次に全行程での所要時間を見ると、やはり種まきオペレーションの方が摘み取りオペレーションより短かった。これは、前述のピッキング工程における差異、そして“POS (梱包場所) への搬送”工程における差異が影響していると考えられる。摘み取りオペレーションを実現するオーダー方式は本来、1つの注文に対するSKUが多いときに用いる方式である。よってピッキングを完了した折りコンは、1つ1つの注文ごとに別個にフォークリフト等で搬送される。これに対し種まきオペレーションを実現するトータル方式は、1つの注文に含まれるSKUは少なく、また再分配が必要なため、複数の注文に含まれる商品をまとめて搬送する。この違いには優劣があるわけではなくSKUに応じて切り替えるべきものであるが、今回はFig. 10の通りSKUが少ない注文が多いため、種まきオペレーションの方が“POSへの搬送”工程においては時間が削減されるという結果になった。

これらの結果から、この注文群を処理するには種まきオペレーションの方が優れているという、現場の判断と一致する結果となった。

## 7.2 シャッフルロケーションの導入

前節の検証において、商品のロケーション (格納場所) は実際の倉庫で採用されているものを用いた。このロケーションは注文の性質を鑑みて現場の責任者が決定したものであるが、この配置が妥当な物かは明らかになっていない。よって本節では、実際のロケーションで業務を実行した場合と、ランダムにロケーショ

ンを設定し業務を実行した場合とでのモデルの出力を比較する。その結果を、Fig. 14及びFig. 15に示す。なお、モデルの入力はある日の倉庫が受注した注文であり、処理を行う実行方式は実際に用いられたオペレーションを反映するものとした。7.1節同様、ピックモデルの出力をFig14, また全体モデルの出力をFig. 13で示す。

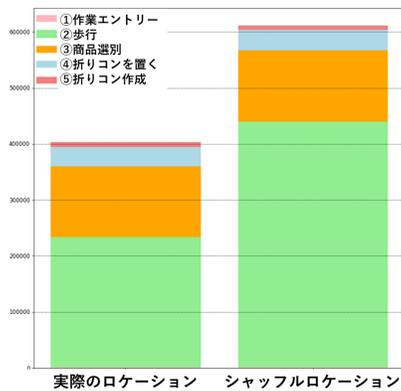


Fig. 14: ロケーションに関するピックモデルの出力

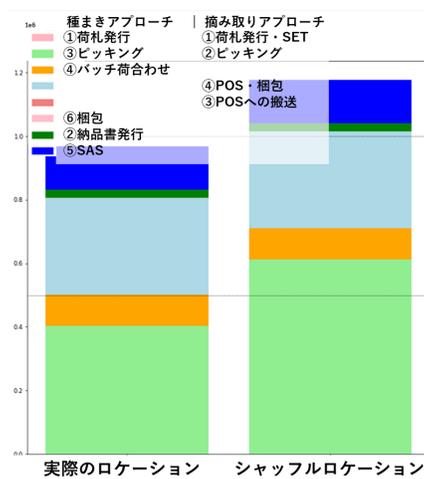


Fig. 15: ロケーションに関する全体モデルの出力

同じ注文を同じ方式で処理しているため、差異が発生するのは”歩行”工程のみである。このことから歩行工程に注目すると、シャッフルロケーションで実行した場合は実際のロケーションで実行した場合の約1.896倍の処理時間を要した。一方で、全行程を通した時には、この差は約1.305倍に縮まった。これは歩行工程はあくまで全行程の一部でしかなく、この工程での所要時間の悪化が全体工程に与える影響は、比較的小さいためである。このことから、ロケーションを整理するのにかかる時間が長い場合には、シャッフルロケーションを検討する必要があることが分かった。

## 8 結論

本研究では、物流倉庫のオペレーション最適化にエージェントベースモデルを用いて取り組んだ。その過程において、あらゆる倉庫に手軽にモデルを適用できるようにするためには、少量の限定的なデータからモ

デルをフィッティング出来る必要性が明らかになった。本研究ではその方法として、データの前処理法とフィッティング方法を提案した。その結果は7章の中で示されているように、一定の精度がみられ、また物流現場における判断と同じ結果がみられた。

一方でこうした主観での検証ではなく、客観的なモデルの精度の検証は行えていない。これは、データが不十分であることを前提としており、真に正解となるデータを用意できていないためである。モデルを適用する各倉庫において正解データを用いたモデルの精度検証は必要ないが、モデルを構築する段階においてはそのモデル、およびパラメータの妥当性を検証することは不可欠である。こうした検証は前述のようにデータの制約から課題が多く、今後の課題と言える。

また、今回はモデルによってあるオペレーションを行った場合にかかる各工程での所要時間を出力することにより、最適なオペレーションを議論可能にしたが、最適なオペレーションを自動で導き出すことはできていない。これは受注した注文に対しあらゆるオペレーションの組み合わせを仮定しモデルを実行することで得られると考えられる。一方で実際のオペレーションは早く終わればよいのではなく、期限や人員数、優先順位など今回のモデルには含まれていない様々な背景が評価関数に含まれる。今後の研究では、こうした多様な背景を考慮した上で、最適なオペレーションを提案する仕組みを考えていく必要がある。

## 参考文献

- 1) Minfang Huang, Qiong Guo, Jing Liu, Xiaoxu Huang: Mixed Model Assembly Line Scheduling Approach to Order Picking Problem in Online Supermarkets, Sustainability 2018, 10(11), 3931 (2018)
- 2) Mustafa Guller, Tobias Hegmanns: Simulation-based Performance Analysis of a Miniload Multishuttle Order Picking System, Procedia CIRP, Volume 17, 2014, 475/480 (2014)
- 3) Hiroyuki Masuda, Shinya Kikuchi, Takeshi Arai: Performance Analysis of Retail Store Layout Using Agent-Based Simulation, J Jpn Manage Assoc 60, 128/144 (2009)