

物流倉庫におけるオーダーピッキングシミュレータの開発

佐久間 拓人^{1,a)} 渡邊 南美² 伊原 滉也^{1,3} 加藤 昇平^{1,3,b)}

概要: 本研究は物流センターにおけるオーダーピッキングシミュレータを開発し、現場の実績データに近づけることを目的としている。このシミュレータを用いることで、商品配置最適化アルゴリズムが出力した商品配置の評価を実際に現場で商品配置を変更する前に検証することで、より効果的な商品配置を提案することができるようになる。シミュレータの性能は商品伝票が作成されてから伝票内の商品が全てピッキングされるまでの時間（生存時間）によって評価した。実倉庫における2020年9月から12月の実績データを用い、それぞれの月において適したパラメータを選定することで実績の生存時間に近いシミュレーションが可能であることを確認した。

1. はじめに

物流センターでは製品を生産者から需要者へと届けるために必要な保管、荷役、流通加工、包装・梱包の四つの作業が主に行われている [1]。特に荷役作業のうち、作業者が倉庫内の棚を巡回して商品の収集を行う出荷ピッキング作業は物流倉庫内の業務のコストの約 55% という大きな割合を占めている。この出荷ピッキング作業を効率化することが物流業務全体のコストダウンにつながるため、現在様々な研究が行われている [2], [3], [4]。

渡邊らは粒子群最適化手法の一つである BLPSO [5] を用いた商品配置最適化システムを提案している [3]。渡邊らは実験により実倉庫の現配置や、Class-based storage [2]、その他の粒子群最適化手法が算出した商品配置よりも、BLPSO は優れた商品配置を算出できることを示している。しかし、商品配置の良さは簡易的に移動距離に基づく計算式によって評価されており、実際にピッキング作業を行った際に良い商品配置であるかは検討の余地がある。例えば、注文頻度の高い商品を発送作業を行うピッキングステーショ

ンの近くに固めることで移動距離は削減できるが、多くのカートがその場所に集まることで渋滞が発生し、かえってピッキング作業に時間がかかってしまう可能性が高い。実際に商品の配置を変更するのは非常にコストがかかり安易にシステムが提案した商品配置は実倉庫で試せないため、このような渋滞問題なども加味した上で最適化後の配置をより正確に評価する必要がある。

本研究は商品配置をより正確に評価するために、カートが商品をピッキングして周る動きを再現したオーダーピッキングシミュレータの構築を目的とする。図 1 に商品配置最適化アルゴリズムと本シミュレータの関係を示す。最適化アルゴリズムが出力した商品配置を基にシミュレーションを行い、ピッキングにかかるコストを算出する。このコストが現在の配置からどれだけ改善されたかが、再度最適化をやり直すかどうかや、実際に現場に最適化後の商品配置を適用するか否かの意思決定の材料となる。そのため、本稿で提案するシミュレータはたとえ最終的な評価の算出に時間がかかっても実倉庫に商品配置を適用した場合に近い評価を下せることが求められており、最適化アルゴリズム内で用いられる適応度関数にとってもかわるものではないことに注意していただきたい。

本稿では株式会社八幡ねじより提供して頂いた倉庫に関するデータと作業者に課している目標作業時間などからオーダーピッキングシミュレータを構築し、出荷実績情報により近づくよう実験を行った結果を報告する。

表記法

1 から任意の自然数 n の集合を $[n] := \{1, \dots, n\}$ とする。 n 次元ベクトル v に対して、インデックス集合 $I \subseteq [n]$ に

¹ 名古屋工業大学大学院工学研究科工学専攻情報工学系プログラム Computer Science Program, Dept. of Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya-city, Aichi 466-8555, Japan

^{a)} sakuma@katolab.nitech.ac.jp

^{b)} shohey@katolab.nitech.ac.jp

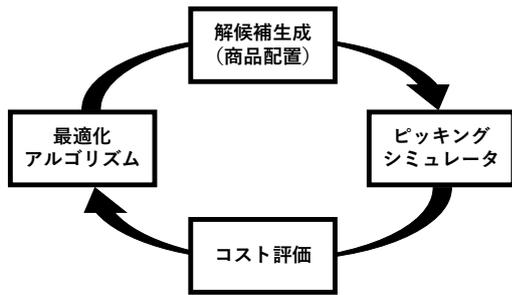


図 1 商品配置最適化アルゴリズムと本シミュレータの関係

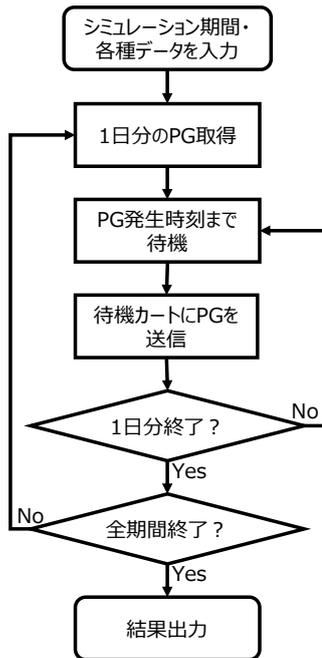


図 2 オーダーピッキングシミュレーションの流れ

対応する部分ベクトルを v_T と表記する。長さ T の系列を (g_1, g_2, \dots, g_T) と表記する。

2. オーダーピッキングシミュレータ

オーダーピッキングでは注文伝票をもとに同じ箱へ梱包する商品のリストとしてピッキンググループ (PG) を作成する。PG が発行されてから PG 中の商品を全てピッキングするまでの時間を生存時間と定義し、生存時間が実績に近づくほど良いシミュレートが出来ていると仮定する。そのため、実績通りの時間に PG を発生させ、シミュレータ内のカートが PG 中の商品全てをピッキングし終わるまでの時間を計測し、実績との誤差を PG ごとに出力できるようにした。図 2 にオーダーピッキングシミュレーション全体の流れを示す。シミュレーションを行う期間と各種データを入力することで、PG ごとの生存時間を指定した期間分まとめて出力する。シミュレーション内では午前 4 時から翌日午前 3 時 59 分までを一日として扱い、一日分 PG を取り出して処理していく。なお、一日ごとにシミュレータはリセットされ、未完了の PG が残っていた場合は破棄

表 1 シミュレータで用いる各種パラメータ

パラメータ	値	備考
移動係数	1.646	1m を移動するのにかかる秒数
共通作業時間	35 秒	PG 受領に関する作業
箱ピッキング作業時間	9.2 秒	ピッキング単位が箱かケースの場合
ばらピッキング作業時間	40.2 秒	ピッキング単位がばらの場合
拠点作業時間	30 秒	拠点到達後の荷降ろし作業など

される。実績ではその日に処理できた PG データを用いたシミュレーションであるため、破棄された PG 数が多いほど実績との乖離が大きいシミュレータであると言える。

PG は待機しているカートに送信され、待機カートがない場合は待機カートが現れるまで保持される。PG を割り振られたカートは次節以降で述べる行動決定アルゴリズムに従って PG 中の商品を収集し、収集にかかった時間を報告する。カートは複数台並列で動いており、その台数は実際の作業人数を一時間ごとに記録したものを基に一時間ごとにカート台数を変化させてシミュレートする。

2.1 シミュレータ設定

全棚集合 $s := \{s_i\}_{i \in [|s|]}$ およびピッキングステーションをノード、異なる二つのノード間のマンハッタン距離をエッジの重みとして持つ無向グラフ G を構成し、カートの移動はノードを遷移することで表現する。ピッキングステーションとは棚からピッキングした商品を梱包・発送作業を行う場所であり、カートは一定条件ごとにピッキングステーションへ戻る必要がある。本稿では 4PG 収集し終わった時点か、カート内の商品重量が 100kg を超えたら現在収集している PG が終わった時点でピッキングステーションへ戻るものとした。カートは実績に基づいて一時間毎に台数を変化させる必要があるため、台数を増加させる際は増加台数分の待機カートを増やし、減少させる場合は台数分ピッキングステーションへの帰還命令を出し、現在とりかかっている PG の処理が終了次第ピッキングステーションへ帰還させ、拠点作業時間経過後にカートを停止することとした。表 1 にシミュレータで用いる各種パラメータを示す。これらはすべて実環境における測定結果を元に設定されている。なお、「ばらピッキング作業時間」とは商品を本数単位でピッキングする際の作業時間であり、本数が正確であるかの確認などに時間を要するため「箱ピッキング作業時間」よりも長い時間が設定されている。

2.2 カートの行動決定アルゴリズム

カートはある PG g 内に含まれる商品が配置されている棚の集合 $s^g := \{s_i^g\}_{i \in [|s^g|]}$ をすべて辿る必要があるため、巡回セールスマン問題 (TSP) として捉え、TSP の近似解を求めるアルゴリズムによって最短距離で辿る順番を求め、目標棚系列 $S^g := \langle s_j^g \rangle_{j \in [|s^g|]}$ を作成する。棚集合 s^g の要素 s_i^g をノード、異なる二つのノード間のマンハッタン距

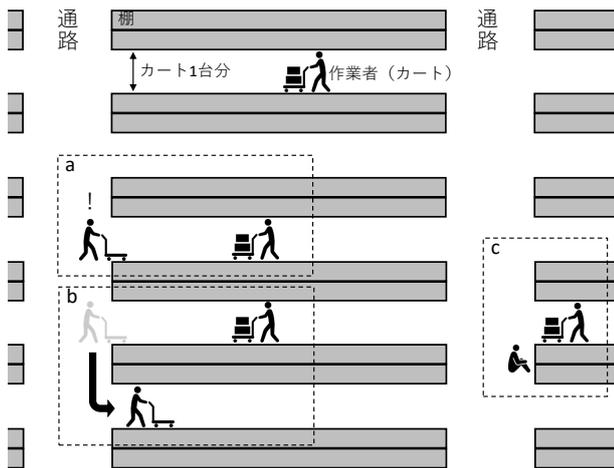


図3 倉庫模式図およびカートの行動決定例

離をエッジの重みとして持つ完全無向グラフを構成する。TSPを解くアルゴリズムはクリストフィードのアルゴリズム [6] とし、得られた目標棚系列 S^g に従いカートは移動する。現在のノードから次の目標ノードまでの最短経路は全棚集合 s をノードとして持つグラフ G と A*アルゴリズムにて求める。A*アルゴリズムにおけるヒューリスティック関数はノード間のユークリッド距離を用いた。カートは A*アルゴリズムにて求めた最短経路に従いノードを遷移する。ノードの遷移にかかる時間はノード間の距離に移動係数を積算して求める。

図3に本稿で対象とする倉庫の模式図を示す。棚と棚の間は狭く、カート1台分の幅しかないため基本的に他のカートとすれ違うことができない。しかし、他のカートとすれ違えるほど広い通路に面している棚も存在する。本稿ではこれを通路ノードと呼び、主に他のカートとのすれ違いや待機場所として使用する。通路ノード以外で他のカートと出会った際(図3a)、カートは別のルートへの回避(図3b)か、通路ノードで待機(図3c)する。人間同士であれば、互いに声を掛け合うことで状況に応じた最適な選択をすることを考える。例えば、塞いでいるカートの作業が長引きそうであれば別の商品のピッキングを優先し回避する。あるいは、塞いでいるカートがすぐにその場を離れるようであれば通路で待機し、動向を伺うなど。本稿で対象としている実倉庫では減多に待機せず、基本的に回避方策を採用していると伺ったため、以下の条件分岐に従い行動を決定するとした。

- (1) 次の目標ノード s_{j+1}^g が S^g 内に存在するなら、現在の目標ノード s_j^g を S^g の最後尾に移動し、現在のノードから目標ノード s_{j+1}^g までの最短ルートを A*アルゴリズムで探索
- (2) 1に当てはまらず、現在のノードから s_j^g までの間に通路ノードが存在するなら、別ルートを A*アルゴリズムで探索

- (3) 上記全てに当てはまらなければ、直近の通路ノードで障害となっているカートがいなくなるまで待機
- 目標ノードへ到達後、その場でピッキングできる商品すべてをピッキングするまでカートは停止する。なお、ピッキングにかかる時間はピッキング単位が箱かケースならば「箱ピッキング作業時間」を、それ以外は「ばらピッキング作業時間」をピッキングする商品数分加算することで求める。

3. シミュレータ性能評価実験

3.1 実験設定

本稿におけるオーダーピッキングシミュレータは実際のピッキング作業者の動きを再現することを目的としている。表1に示したパラメータ類は現場における目標値であり、実際にはそれ以上の時間がかかっていると考えられる。本実験では表1に示したパラメータ全てに対して1.0から2.0まで0.1刻みの係数をそれぞれ積算した条件を用意した。例えば、係数1.5の実験条件では、表1に示したパラメータ全てを1.5倍した上でシミュレーションを行った。実験条件それぞれにおいて実績とどれだけ差があるかをPGの平均生存時間の差を基に調査した。この差が小さい値であるほど、実績との乖離が少ないシミュレーションができていると仮定する。また、シミュレータが処理できたPGのみが平均生存時間の算出および実績との比較に用いられるため、PG数が大きく減っているシミュレーション結果は実績との乖離が大きいと仮定する。なお、PGによっては実績の生存時間が4時間を超えているものも存在する。これはPGの発行だけされたままならかの事情で放置されたものや、優先すべき他のPGが割り込まれたために処理が遅くなったものであり、通常のピッキング作業とは明らかに異なるものである。実績の生存時間4時間以上のPGは外れ値として本シミュレータの評価からは除外した。実験には株式会社八幡ねじから提供を受けた2020年9月から12月の4ヶ月間(76980 PG)のデータを使用し、シミュレーションはそれぞれ1ヶ月分で行った。シミュレーション対象となる倉庫は面積2058m²、棚数635に約22000種類の商品が収められている。

3.2 実験結果

表2, 3, 4, 5に2020年9月から12月のシミュレーション結果として処理完了したPGの平均生存時間および実績における平均生存時間を示す。シミュレーション結果と実績間の平均生存時間の差を示した列の絶対値が最も小さいところを太字で表記する。

表2, 3, 4, 5より、係数1.0条件における平均生存時間が実績よりも短いことがわかる。すなわち、作業目標として設定されている各種パラメータ(表1)が実績では達成できていない可能性が高い。一方で、係数2.0条件にお

Algorithm 1 カートの行動決定アルゴリズム

Require: PG g

```

1: PG 受領後共通作業時間分その場で停止
2:  $g$  内で訪れる必要がある棚集合  $s^g$  をすべて通る最短ルートをクリストフィードのアルゴリズムで求め、辿る順番でソートした目標棚系列  $S^g$  を作成
3: for  $s_j^g \in S^g$  do
4:   現在のノードから次の目標ノード  $s_j^g$  までの最短ルートを A*アルゴリズムで探索
5:   for A*で求めたルートのノード系列 do
6:     if 他のカートがいて移動できない then
7:       if  $s_{j+1}^g \neq \text{NULL}$  then
8:          $s_j^g$  を  $S^g$  の最後尾に移動し、現在のノードから目標ノード  $s_{j+1}^g$  までのルートを A*アルゴリズムで探索
9:       else if 現在のノードから  $s_j^g$  までの間に通路ノードが存在する then
10:        別ルートを A*アルゴリズムで探索
11:      else
12:        直近の通路ノードで障害となっているカートがいなくなるまで待機
13:      end if
14:    end if
15:    次のノードへ移動
16:  end for
17:   ピッキング作業時間分停止
18: end for
19: if 4PG 分収集 or 商品重量が累計 100kg 超 or 帰還命令 then
20:   ピッキングステーションへ移動し拠点作業時間分停止
21: else
22:   直近の通路ノードへ移動
23: end if

```

表 2 2020 年 9 月のシミュレーション結果

係数	生存時間 (秒)		実績生存時間 (秒)		平均の差	PG 数
	平均値	SD	平均値	SD		
1.0	1539	2133	3405	3514	-1865	17577
1.1	1848	2579	3405	3514	-1557	17577
1.2	2270	3130	3405	3514	-1134	17577
1.3	2717	3607	3405	3514	-688	17576
1.4	3354	4087	3403	3513	-49	17550
1.5	3976	4654	3394	3500	581	17388
1.6	4640	5277	3381	3503	1259	17028
1.7	5460	5633	3376	3517	2085	16669
1.8	6306	6090	3390	3532	2916	16120
1.9	7118	6598	3364	3525	3755	15403
2.0	7611	6802	3301	3482	4310	14515

表 4 2020 年 11 月のシミュレーション結果

係数	生存時間 (秒)		実績生存時間 (秒)		平均の差	PG 数
	平均値	SD	平均値	SD		
1.0	2940	3835	4303	3916	-1363	14430
1.1	3551	4328	4304	3916	-753	14429
1.2	4315	5025	4307	3917	8	14401
1.3	5149	5516	4323	3921	826	14314
1.4	6189	6015	4326	3924	1863	14280
1.5	7354	6659	4340	3928	3013	14152
1.6	8178	6835	4337	3921	3841	13829
1.7	9187	7220	4293	3895	4894	13443
1.8	10112	7668	4294	3879	5818	12715
1.9	10751	8048	4284	3866	6467	11925
2.0	11195	8413	4238	3810	6957	11310

表 3 2020 年 10 月のシミュレーション結果

係数	生存時間 (秒)		実績生存時間 (秒)		平均の差	PG 数
	平均値	SD	平均値	SD		
1.0	2505	2529	4264	3946	-1760	17025
1.1	3817	3854	4262	3939	-445	16963
1.2	4282	3705	4265	3946	17	17024
1.3	5382	4221	4264	3946	1117	17025
1.4	6616	4836	4265	3946	2351	17023
1.5	7957	5733	4261	3938	3696	16709
1.6	8928	6240	4265	3929	4664	16067
1.7	9937	6905	4225	3890	5712	15315
1.8	11031	7584	4212	3874	6819	14351
1.9	11834	8198	4220	3888	7614	13449
2.0	12433	8767	4222	3894	8212	12498

表 5 2020 年 12 月のシミュレーション結果

係数	生存時間 (秒)		実績生存時間 (秒)		平均の差	PG 数
	平均値	SD	平均値	SD		
1.0	2730	3317	4330	3969	-1599	13183
1.1	3312	3882	4330	3969	-1018	13183
1.2	4220	4622	4330	3969	-110	13183
1.3	5064	5231	4330	3969	734	13183
1.4	5968	5829	4330	3969	1638	13183
1.5	6917	6525	4332	3973	2586	13095
1.6	8565	7512	4294	3959	4271	12736
1.7	8901	8148	4302	3962	4599	12500
1.8	9741	8396	4283	3939	5459	12036
1.9	10364	8735	4343	3960	6021	11526
2.0	11032	8909	4356	3976	6676	11071

ける平均生存時間は実績よりも長く、処理 PG 数が 1.0 条件よりも大きく減少している。これは各 PG の処理にかかる時間が嵩み、一日が終わる時点で未完了の PG が大量に残っていたことを示しており、係数 2.0 条件はシミュレーションとしてふさわしくないと考える。表 2 より 9 月における最も平均生存時間の差が小さい係数は 1.4 である。表 3, 4, 5 より、10 月, 11 月, 12 月における最も平均生存時間の差が小さい係数は 1.2 である。一方、9 月の係数 1.4 条件および 11 月の係数 1.2 条件の結果は処理 PG 数が係数 1.0 条件と比べて 30 弱程度ではあるが減少しており、実績では処理できていた PG をシミュレータは処理できていないと言える。

4. 考察

実験結果より、当初の仮説通り、現場における目標値であるパラメータ類 (表 1) のままでは実績を再現できないことがわかった。2020 年 9 月はパラメータを 1.4 倍、10 月, 11 月, 12 月はパラメータを 1.2 倍することで最も実績に近い平均生存時間となることがわかった。今回実験対象としていない月に対してもおおよそ 1.2 倍から 1.4 倍を積算することで実績に近づく可能性があるが、本研究の目的は汎用的な係数を見つけることではないため、シミュレーションを行う期間に合わせて最適な係数を都度見つけるようにしたい。実験結果から月ごとに最適な係数を見つけることで実倉庫の実績に近いシミュレーションができることがわかったが、9 月, 11 月のシミュレーションは実績に沿いきれておらず、未処理の PG が発生していたことからわかるように本シミュレータには改良の余地がある。例えばカートの渋滞について、本シミュレータでは単純に通路ノード以外ではカートはすれ違えないものとしている。そのため、他のカートがピッキング作業中などで通路を塞いでいる場合、シミュレータ内のカートは別ルートの探索 (図 3b) や他のカートがいなくなるまで通路ノードで待機する (図 3c) 方策を取る。しかし、実倉庫ではカートを置いて人間だけ商品を取りに行くことや、他の作業者に場所を譲ってもらうなど、臨機応変に渋滞に対応している。さらに作業員間の序列や現場への熟練度に応じてカートの挙動および作業にかかる時間は大きく変わることが予想されるが、これらの要素はシミュレータに組み込まれていない。本実験では一ヶ月間の平均生存時間を実績へ近づけるために係数を用意したが、係数の調整を一日単位で行えば作業員の差やその日特有の受注などを係数で吸収でき、より実績へ近づけることが可能だと考える。係数調整に関しても、すべてのパラメータに一律の係数を積算するのではなく、個別に調整することでより実績へ近づけることが可能である。これらは人手で調整する限界を超えるので、パラメータ最適化手法としてベイズ最適化を用いるなど、効率化を考えていきたい。

さらに別のアプローチとして、PG の生存時間ではなく、PG の処理開始から終了までを計測した処理時間を実績に近づける方針も考えられる。生存時間は前述の通り、発行だけされたまま放置された PG や割り込みによる処理順番の変更の影響が大きく、シミュレーションで近似しづらい面もある。処理時間であれば、純粋にピッキングカートの挙動を模倣したシミュレーションによって十分近似できると考える。残念ながら処理時間の実績データが存在しないため現時点では生存時間を基準とするしかなかったが、データ提供先と今後も提携し、処理時間の計測データが集まった時点で取り組んでいきたい。

5. おわりに

本研究は商品配置最適化アルゴリズムが出力した商品配置の評価をより正確にするため、現場の実績データに近いオーダーピッキングシミュレータの開発を目的としている。本稿では開発したシミュレータを用いて 2020 年 9 月から 12 月の実績データを基に複数条件で実験を行い、実績データに近づくパラメータの発見を試みた。結果として、それぞれの月において PG の平均生存時間が実績に最も近くなるパラメータの発見に至った。このシミュレータを用いることで、実際に現場で商品配置を変更する前に検証することができ、より効果的な商品配置を提案することができるようになると思う。

謝辞 本研究は、一部、文部科学省科学研究費補助金 (課題番号 JP19H01137, JP19H04025, JP20H04018, および、JP20K19905) の助成により行われた。物流業務の実データおよびカートの挙動を決定するためのパラメータを提供いただいた株式会社八幡ねじに謝意を表します。

参考文献

- [1] 鈴木邦成: 図解 物流センターのしくみと実務, 日刊工業新聞社 (2014).
- [2] De Koster, R., Le-Duc, T. and Roodbergen, K. J.: Design and control of warehouse order picking: A literature review, *European journal of operational research*, Vol. 182, No. 2, pp. 481–501 (2007).
- [3] 渡邊南美, 伊原滉也, 佐久間拓人, 加藤昇平: Lévy Flight を導入した Binary PSO による物流倉庫の商品配置最適化, 人工知能学会全国大会論文集 第 35 回全国大会 (2021), 一般社団法人 人工知能学会, p. 2F4GS10h01 (2021).
- [4] Lee, I. G., Chung, S. H. and Yoon, S. W.: Two-stage storage assignment to minimize travel time and congestion for warehouse order picking operations, *Computers & industrial engineering*, Vol. 139, p. 106129 (2020).
- [5] Ihara, K. and Kato, S.: A Novel Sampling Method with Lévy Flight for Distribution-Based Discrete Particle Swarm Optimization, *IEEE Congress on Evolutionary Computation (CEC)*, IEEE, pp. 2281–2288 (2021).
- [6] Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem, Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group (1976).