

注文商品の共起ネットワークを利用した 商品配置方法の提案

南 賢一^{1,a)} 古川 正志² 山本 雅人²

概要: 物流センター内の作業の中で、注文された商品を保管場所から取り出してくるピッキング作業に最もコストを要している。ピッキング作業を効率化する方法の一つに商品の配置の改善が挙げられる。本研究では、注文される商品同士の依存関係を共起ネットワークで表し、生成された共起ネットワークを自己組織化マップにより棚の位置に写像することによる商品配置の方法を提案する。

キーワード: 物流センター, 共起ネットワーク, 自己組織化マップ

A Proposal of Storage Location Assignment Using Co-occurrence Network of Order Items

MINAMI KENICHI^{1,a)} FURUKAWA MASASHI² YAMAMOTO MASAHITO²

Abstract: Order picking has a great affect with efficiently controlling a large-scale logistic center. It is important for order picking not only to give workers a good tour plan but also to assign products to the storage shelves in the center. This study proposes a new method how to assign products to the storage using an ordered product relation network and Self-Organizing Map.

Keywords: distribution center, co-occurrence network, Self-Organizing Map,

1. 緒言

1.1 背景

近年、インターネットの普及によって、個人がインターネット上でさまざまな商品のを手軽に購入できるようになった。それによって扱かれる商品の多品種・多様化が進み、納期も厳密になったために、物流センターの重要性がより注目されるようになった。

多くの企業にとって物流センターでのコストの削減や生産性の向上が大きな課題となっており、多くの研究が行われている。物流センター内では、商品の保管、伝票に応じて商品を棚から運び出すピッキング作業、仕分け、配送、流通加工などが行われ、中でもピッキング作業は物流センターの全作業時間の大部分を占めている。そのため、ピッキング

作業の効率化を図ることによって物流コストを下げる事が可能である。ピッキング作業の効率化の一つに棚への商品配置の最適化が挙げられるが、棚への商品配置の方法は経験則で行われている場合が多い。また、商品配置の最適化によるピッキング作業の効率化に関する研究は少なく、その内容もクラス分けによるものが多い。

1.2 目的

物流センターでは、顧客からの注文の伝票に基づいて商品のピッキングを行う。このときある特定の2商品が同じ伝票内で同時に注文されるような依存関係があることが知られている。このため、物流センターの倉庫に商品を保管する際に同時に注文される商品同士をお互い近くの棚に配置することで、商品のピッキング作業時の移動コストを削減することが出来き、ピッキング作業を効率化することが可能である。

本研究では、伝票から商品の共起ネットワークを生成し、

¹ 北海道大学 工学部 情報エレクトロニクス学科

² 北海道大学 情報科学研究科

^{a)} minami@complex.ist.hokudai.ac.jp

生成された共起ネットワークに対して自己組織化マップを適用することで商品配置を行う方法を提案し、商品同士の依存関係を商品配置に反映させピッキング作業の効率化を図ることを目的とする。

2. 物流センター

2.1 物流センターとは

2.1.1 物流システム

物流センターとは、多種多量の商品の保管、仕分け、流通加工などを行い、顧客の注文に応じて商品を配送する役割を持った重要な物流拠点であり、その目的は顧客からの注文にしたがって商品を顧客に届ける、物流の流れを円滑に行うことである。物流センター内での作業工程は大きく分けて「入荷処理」と「出荷処理」があり、入荷処理は「入荷」、「格納」、「保管」などを指し、出荷処理は「ピッキング」、「流通加工」、「集積」、「仕分け」、「梱包」、「積み込み」などを指す。

2.1.2 オーダーピッキング

オーダーピッキングとは、発注されたオーダー（伝票）に基づいて商品を集荷することである。オーダーピッキングは、物流センターの全作業時間の約60%を占めるといわれており、オーダーピッキングの作業効率を向上させることで、物流センターにかかるコストを削減する出来る。オーダーピッキングの作業内容は、棚から商品を探し、棚から商品を取り出すというステップを繰り返す。ピッキング経路が長くなる理由として、ピッキング経路の決定は作業員それぞれに任されているので、どのような経路で商品を集めれば効率的であるかは経験上の知識に大きく左右され、作業員の経験量によってピッキング時間に差が出てしまうことが挙げられる。ピッキング経路を短くする方法として、伝票を複数まとめて作業員に割り当てて、一度に複数の伝票を集配する方法がある。これにより一度に集配する商品の数が増えるため、ピッキング作業の回数を減らすことができる。伝票の割り当て方法としては、伝票の到着順に一定数割り当てていく方法や、一定時間で区切り、その時間内で来た伝票を割り当てていく方法などが用いられている。しかし、これらの方法は伝票の内容を考慮せずに受付順で決まるため、似た伝票が異なる作業員に割り当てられ、作業に無駄が発生するという問題が生じる。伝票割り当ての最適化では、似た伝票を同じ伝票のまとまり（バッチ）になるように分割する。また、商品を保管する場所を改善することでピッキング経路を短くする方法がある。一回のピッキング作業で頻繁に同時に集配されるような商品を近くに集めることで棚から棚への移動距離が短くなる。本研究ではこの問題について取り上げる。

2.2 商品配置問題

商品配置とピッキング作業効率の関係は非常に深く、商

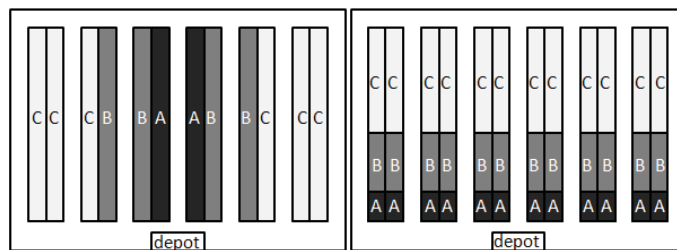


図1 クラスごとの配置例

品の配置の仕方により作業時の移動距離、待ち時間、探索時間などが変化し、ピッキング作業に大きな影響を与える。

2.2.1

2.3 注文頻度による配置

注文頻度による配置の手法に Class-based storage(Graves, 1977)[1]が挙げられる。Class-based storageとは、注文頻度などの基準によって商品をクラス分けを行い、クラスごとに商品を配置していく手法である。商品の分類方法は、需要量の多い約20%の商品のみで粗利益の約80%を占めるというパレートの法則に基づいて分類されることが多い。このとき、注文頻度の最も高いクラスをA-itemsと呼び、以下順にB-items, C-itemsと呼ぶ。分類されたクラスの割り当て方法として、図1のような1つの通路に対して1つのクラスの商品だけを配置する方法[2]、出入口に近い順に上位のクラスの商品を配置する方法[3]などがある。

2.4 商品同士の依存関係による配置

上に挙げた Class-based storageは、商品同士の依存関係を考慮していない。もしもある商品と同時に注文されることが多い別の商品が存在する場合、その2商品は近くに配置する方が効率的である、そのような考え方を取り入れた手法に Family groupingが挙げられる。Family groupingは、商品同士の依存関係が強い商品同士を保管エリアの同じ領域内に配置する。この手法は他の手法と組み合わせることが可能で、上に挙げた Class-based storageと組み合わせることが可能である。しかし、商品が割り当てられるクラスの決定は、各商品の持つ性質の組み合わせに依存する。

3. 共起ネットワーク

3.1 共起ネットワークとは

共起とは、ある任意の文・文書において、ある語とある語が同時に出現することである。文書からその文書を構成する語をノードとし、語同士の共起関係を辺としてネットワーク図にしたものが共起ネットワークである。ネットワーク図に表すことにより単語同士の関係が視覚的に理解しやすくなるメリットがある。共起ネットワークの関連性の高い単語同士の結び付きが密になり、関連性の低い単語同士の

伝票番号	商品名	伝票番号	商品名	伝票番号	商品名
1	商品A	2	商品C	3	商品C
	商品B		商品D		商品D
					商品E

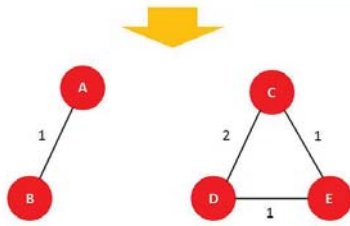


図 2 注文商品の共起ネットワーク

結び付きが疎になる性質を利用して、Web 上の情報を用いて、研究者の情報をキーワードとして自動的に抽出する手法が提案 [4] されている。

3.2 伝票における共起ネットワーク

3.2.1 商品の共起

言語解析における共起とは、文書内に含まれる単語同士の関係のことをいうが、注文商品における共起とは、注文データにおける商品の共起を伝票内に含まれる商品同士の関係のことをいう。ある二つの商品について、一つの伝票内に同時に注文された回数が多いほど共起関係が強く、同時に注文される商品であるといえる。

3.2.2 共起ネットワークの生成方法

注文データの共起ネットワークの生成について、商品をノード、商品同士の共起関係を辺として表す。商品同士の共起の重みは、一つの伝票内で同時に注文された回数である。データにおける商品の共起ネットワークの生成方法の例を図 2 に示す。三つの伝票データより、商品 A と商品 B、商品 C と商品 D と商品 E は同一伝票内で同時に注文されているので、それぞれノードの間に辺が引かれる。共起の重みは、商品 C と商品 D が 2 つ伝票で同時に注文されているので辺の重みは 2 と定義し、商品 A と商品 B、商品 C と商品 E、商品 D と商品 E は一つの伝票で注文されているので辺の重みは 1 と定義する。また、辺が繋がっていないノード間の重みは 0 と定義する。

3.3 共起ネットワークの実験

実際の商品の注文データを注文頻度によって 3 つクラス A, B, C に分類し、その中でもより重要度の高いクラス A と B について商品同士の関係を共起ネットワークで表す。生成された共起ネットワークを解析し、どのような特徴があるかを考察する。

3.3.1 実験条件

本実験では、注文商品の実データを用いる。実験に用いた伝票のデータを表 1 に示す。表 2 に注文頻度によって分類されたクラスに含まれる商品の割合を示す。ノードの色分けは、共起関係の強さによって発生するコミュニティごと

表 1 伝票の設定

全商品の注文数	50000
商品の種類数	1474
伝票数	11930

表 2 クラスの設定

クラス	商品の種類数	全体の割合
A	74	5.02%
B	150	10.18%
C	1250	84.80%

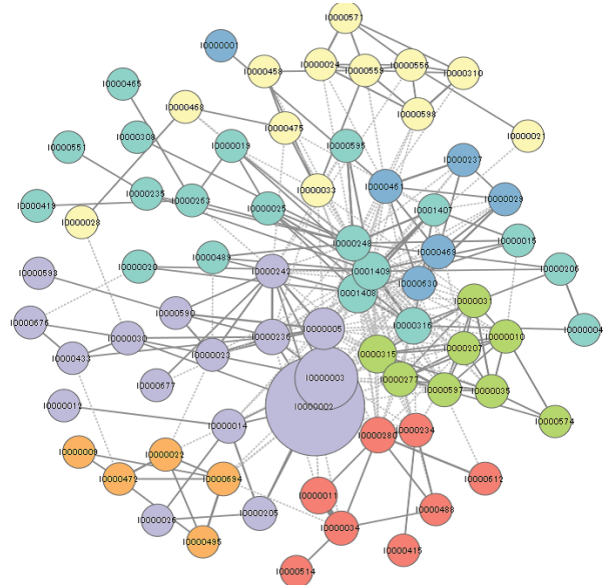


図 3 クラス A の共起ネットワーク

に行う。辺について、共起関係の強いものから順に描画していき、全てのノードに 1 本以上辺が繋がったときに描画を終了する。これは辺を全てを描画した場合、辺同士が重なってしまい、その繋がりや太さ（共起の重み）が見えなくなるのを防ぐためである。

3.3.2 実験結果

図 3 より、クラス A の商品同士の共起関係を共起ネットワークに表すことが出来た。クラス A の共起ネットワークは、複数のコミュニティで構成されており、1 つのコミュニティ内では、ノードはより太いエッジで繋がれている。このコミュニティはノード同士の強い共起関係を表しているため、コミュニティを考慮した商品の配置により、ピッキング作業を効率化出来る。また、異なるコミュニティが細かいエッジで繋がっている場合があり、少なからず共起関係があることが分かる。よって、商品配置の際に、共起関係があるコミュニティ同士を近くに配置することで、ピッキング作業をより効率化出来る。

図 4 より、クラス B 同士においても共起関係を共起ネットワークに表すことが出来、クラス A と同様のことがいえるが、ノードの数がクラス A に比べて多いため、その共起関係はより複雑なものとなっている。

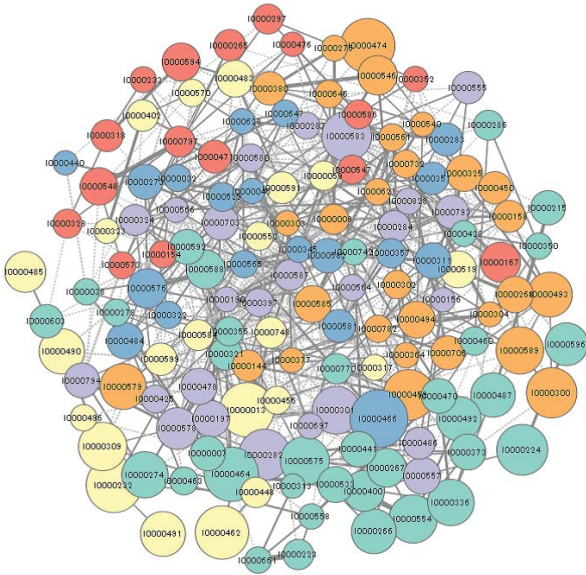


図 4 クラス B の共起ネットワーク

4. 共起ネットワークを用いた商品配置方法の提案

本研究では、伝票から生成された注文商品の共起ネットワークに自己組織化マップを用いて商品配置を行う方法を提案する。

4.1 自己組織化マップ

4.1.1 自己組織化マップとは

自己組織化マップ (SOM: Self-Organizing Map) とは、Kohonen によって提唱された与えられた入力データをその関係性を保ったまま任意の次元上にマッピングする手法で、教師なし学習のニューラルネットワークの一種である。自己組織化マップは、入力層と出力層の二層のネットワークから構成される。出力層は競合層とも呼ばれる。入力層では n 次元の入力データを受け取り、出力層では m 次元上に配置されたノードが入力データを学習する。出力層は主に 1~3 次元へのマッピングに用いられ、多次元のデータ間の関係を可視化できる。自己組織化マップは、プロセス解析、制御、検索システム、経営のための情報分析など実社会において重要な分野へ応用されている。

4.1.2 アルゴリズム

ここでは最も基本的な自己組織化マップについて取り上げる。自己組織化マップは以下の要素から構成される。

- ◆ 入力データの集合 $\{x_t \mid x_t \in \mathbf{R}^n, t = 0, 1, 2, \dots\}$: 学習させたいベクトルの集合
- ◆ ニューロンの集合: 学習させるマップの構成要素, 各々参照ベクトル $w \in \mathbf{R}^n$ を持つ
- ◆ トポロジー: ニューロンの位相関係
- ◆ 近傍関数 h_{ci} : 学習係数率を定義, 時間と距離の関数

◆ 類似度 $d(x, w_i)$: 勝者を決める基準

w_i はニューロン i の参照ベクトル, h_{ci} はニューロン c, i 間の近傍関数, x は信号ベクトル, t は時刻である。自己組織化マップの学習アルゴリズムは以下の通りである。

1. 各入力ベクトルを生成し, その入力ベクトルの集合を \mathbf{X} とする。
2. 出力層の全ニューロンに重みベクトルを与える。
3. 入力ベクトルの集合から入力ベクトル x_j (勝者ニューロン) を選択する。
4. 全ニューロンの集合から, x_j に最も近いベクトル値 m_c を持つニューロン m_c を選択する。つまり, $\|x_j - m_c(t)\|$ が最小となるニューロンを求める。
5. 選択されたニューロンの近傍領域 N_c を定義する。
6. 重みベクトルを更新し, さらに N_c も x_j に近づける。

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x_j(t) - m_i(t)] \quad (1)$$

7. 終了条件を満たしていなければ 3 に戻る。

近傍領域 N_c は学習の初期段階では大きく, 学習ステップが進むにつれて減少させていく。近傍関数 h_{ci} はいくつかの種類があり, 最も簡単な例は N_c 内のニューロンは $h_{ci} = \alpha(t)$, N_c 外のニューロンは $h_{ci} = 0$ で学習を行う。このとき $\alpha(t) \in (0, 1)$ は「学習率」を表している。学習率は近傍領域と同様に学習ステップが進むにつれて減少する。

$$\alpha(t) = \alpha_0(1 - t/T) \quad (2)$$

4.2 問題設定

本研究では、オーダーピッキングを行う倉庫を想定し, その条件の下でピッキング作業の移動距離が最小となる商品を最適な商品配置であると定義する。商品配置において, 一つの棚に一種類の商品を配置するものとする。

4.3 商品配置方法の提案

商品配置に用いる自己組織化マップの各要素は以下である。

- ◆ 入力データの集合: 棚のノードの位置 $\{x_j \mid x_j \in \mathbf{R}^n, j = 0, 1, 2, \dots\}$
- ◆ ニューロンの集合: 商品のノード位置 $\{m_i \mid m_i \in \mathbf{R}^n, i = 0, 1, 2, \dots\}$
- ◆ トポロジー: 注文商品の共起ネットワーク
- ◆ 近傍関数:

$$h_{ci} = \begin{cases} \alpha_1(t) & : i = c \\ \alpha_2(t)(2\sigma(w) - 1) & : i \neq c \end{cases} \quad (3)$$
- ◆ 類似度 $d(x, m_i)$: マンハッタン距離

表 3 伝票の設定

全商品の注文数	60026
商品の種類数	1536
伝票数	15165

表 4 クラスの設定

クラス	商品の種類数	全体の割合
A	81	5.02%
B	280	18.22%
C	1175	76.51%

勝者ノードが選ばれたとき, その他の全てのノードを共起の重みに応じて近づける. $\sigma(w)$ はシグモイド関数を表す. このとき, β はゲインと呼ばれ, この値によって関数の傾きを調節する.

$$\sigma(w) = \frac{1}{1 + e^{-\beta w}} \quad (4)$$

以上に基づいた商品配置の自己組織化マップのアルゴリズムは以下のとなる.

1. 入力ベクトルを設定し, その集合を X .
2. 出力層のニューロンを初期化する.
3. 入力ベクトルの集合から入力ベクトル x_j を選択する.
4. 選択された x_j 最も近いニューロン m_c 決定する.
5. 重みベクトルを式 (1) で更新する.
6. 終了条件を満たしていなければ 3 に戻る.
7. 終了条件を満たしたとき, 注文頻度の高い商品から順に最も近い棚の位置に配置する.

4.4 配置結果の評価

自己組織化マップによって求められた配置が商品同士の共起関係をきちんと反映しているかを以下の評価式により評価する.

$$E = \sum_{i=1}^{I-1} \sum_{j=i+1}^I f_{ij} d_{ij} \quad (5)$$

ここで f_{ij} は商品 i の j 間の共起回数を表し, d_{ij} は商品 i, j の配置された棚間の距離, I は商品の種類数を表している. この評価式は, 共起回数が多い商品同士が互いに近くに配置されたときに小さくなる.

5. 数値計算実験

実際の商品の注文データを用いた商品配置の最適化実験とその考察を行う. 商品を配置する倉庫の形状は, 作業者はピッキング作業時に棚間を格子状に移動できるものとする. 実験に使用する商品の注文データを表 3 に示す.

5.1 出入口を考慮しない配置問題

注文商品の共起ネットワークに自己組織化マップを適用

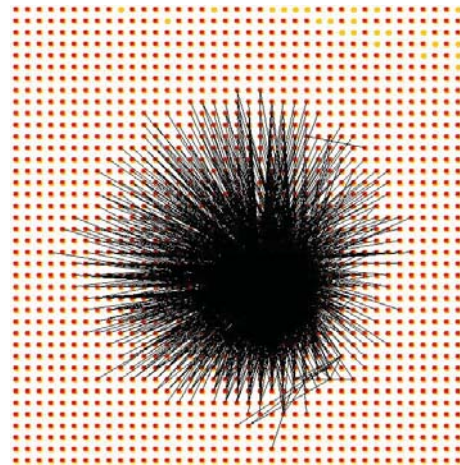


図 5 自己組織化マップでの配置

することによって, 配置結果に共起関係が考慮されているかを検証する.

5.1.1 実験条件

ランダムに商品を配置した初期配置を 5 個作成して実験を行う. 終了条件は 30000 ステップ終了後とする. 配置結果は共起回数が 5 以上の辺のみ表示する. 学習率の初期値は勝者ノードが 0.5, 近傍ノードが 0.65 で 300 ステップごとに 0.999 倍に減少させる. シグモイド関数の β の値は 1/5 である.

5.1.2 実験結果

配置結果を図 5 示す. この結果より, 共起頻度の高い商品ほど倉庫の中央に引き寄せられることがわかった. この配置の評価をするために, 注文頻度の高さによって 3 つにクラス分けを行った Class-based storage との比較を行った. 各クラスの割り当てるエリアは, 倉庫の中央のエリアに最も注文頻度の高い A クラスを割り当て, 以下外側に向かって B クラス, C クラスと割り当てる. 各クラスに分類された商品割合を表 4 に示す. 各クラス内では商品はランダム配置したものを 5 個作成し, 評価を行う. 配置結果を図 6 示す. 自己組織化マップと評価式の値を比較した結果, 提案手法は Class-based storage と比べて評価値が約 10.3% 下がった. これにより, 共起関係の強いもの同士が互いに近くに配置されていることがいえる.

5.2 出入口を考慮した配置問題

前述より, 自己組織化マップを用いて共起関係を考慮した配置を可能とした. しかし, 現実の商品配置問題において, 注文頻度の高い商品を出入口に配置することが望ましい. よって, 本節では, 共起関係に加えて出入口を考慮した配置の実験を行う.

入力ベクトル x_j に加えて出入口のノード d を作り, 一定ステップごとにニューロンを d に近づける. 近づける強さは式 (3) を用いて商品の注文頻度によって調節する. 配置後, 実際の伝票を用いてピッキング経路長を求める. ピッ

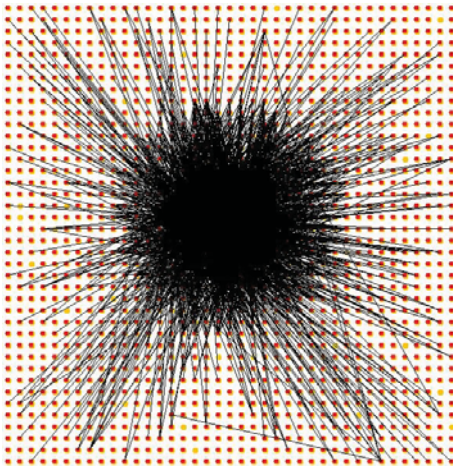


図 6 Class-based storage での配置

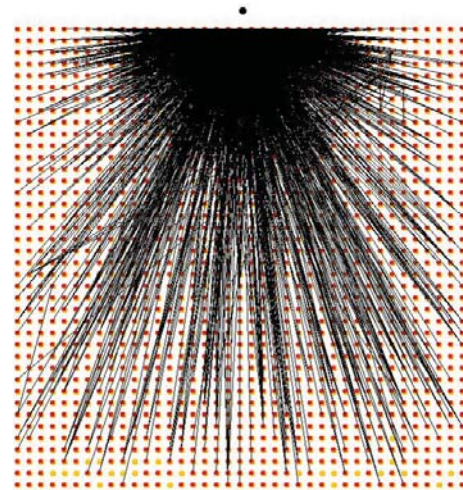


図 7 提案手法での配置

キング経路は巡回セールスマン問題の解法によって求める。同様に Calss-based storage での配置も行い、ピッキング経路を求め比較する。Calss-based storage は、表 4 にしたがって出入口に最も近いエリアに A クラスを割り当て、離れていく順に B クラス、C クラスと割り当てる。

5.2.1 実験条件

5.1 節の実験と同様にランダムに商品を配置した初期配置を 5 個作成して実験を行う。終了条件は 30000 ステップ終了後とし、共起回数 5 以上の辺を表示する。学習率について、 d へ近づける更新式の学習率の初期値は 0.55 で 300 ステップごとに 0.99 倍に減少させ、500 ステップごとに d へ近づける。 x_j に近づける更新式の学習率の設定は 5.1 節と同様である。シグモイド関数の β の値は、 d へ近づける更新式は $1/40$ 、 x_j に近づける更新式は $1/15$ である。

5.2.2 実験結果

提案手法での配置結果を図 7 に、Class-based storage での配置結果を図 8 示す。提案手法での配置は、Class-based storage での配置に比べて辺のクロスが少なく、総ピッキング経路長は、提案手法での配置が Class-based storage での配置に比べて約 43.3%減少した。また、式 (5) の評価値も約 9.8%減少した。これにより、提案手法での配置は商品の共起関係と注文頻度の両方を考慮していることがわかる。

6. 結言

本研究では、注文商品の共起ネットワークに自己組織化マップを適用して共起関係を反映させる商品配置の方法を提案した。また、出入口を考慮した自己組織化マップにより、共起関係だけでなく注文頻度も反映させた方法を提案した。この結果、提案手法による商品配置によってピッキング経路長が短縮されることが確認された。

本研究では棚が格子状に置かれた倉庫を想定したが、一般的な物流センターにおいて棚のネットワークは格子状ではなく異なるネットワーク構造をしているため、それに考慮した配置方法が必要である。

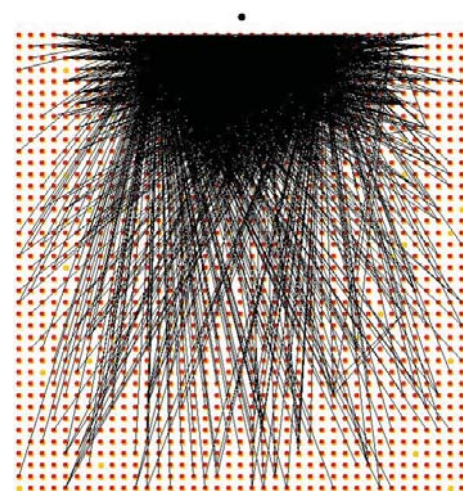


図 8 Class-based storage での配置

参考文献

- [1] Stephen C. Graves, Warren H. Hausman and Leroy B. Schwarz.: *Storage-Retrieval Interleaving in Automatic Warehousing Systems*, Management Science May 1977 vol. 23 no. 9 935-945 (1977).
- [2] Rene de Koster, Tho Le-Duc, Kees Jam Roodbergen.: *Design and Control of Warehouse Order Picking : a literature review*, European Journal of Operational Research, pp.481-501 (2007).
- [3] Shin Y. Chin, Jose H. C. G. Junior: *Modeling and simulation of retrieving process*, IEE Press Piscataway, NJ, pp.1682-1690 (2007).
- [4] 森純一郎, 松尾豊, 石塚満: *Web からの人物に関するキーワード抽出*, 人工知能学会論文誌 巻:20 ページ, pp.337-345 (2005).