자율주행카트를 활용한 디지털 어소팅 시스템의 주문묶음 알고리즘

Operations Analytics Laboratory
Department of Industrial Engineering
Pusan National University, Busan, KOREA

이정만[1], 김영주[2], 홍순도[1]

[1] 부산대학교 공과대학 산업공학과

[2] 철도기술연구원

2019년 한국SCM학회 2019 춘계컨퍼런스
서울특별시 중구 대한상공회의소

목차

- 연구 배경 및 동기
- 문헌연구
- 문제 정의
- 자율주행카트를 활용한 주문묶음 모델
- 휴리스틱 알고리즘
  - Modified Seed algorithm
  - Shortest Route First with First-fit Decreasing
- 실험 결과
- 결론 및 향후 연구

연구 배경 및 동기

- 로봇 및 자율주행 시스템에 대한 지속적인 수요 증가
  - 저출산 및 초고령화 사회의 노동력 문제를 해결하기 위한 대안으로 부상
  - 제조 및 물류에서 활발히 연구됨

- 로봇 및 자율주행 시스템에 대한 기업들의 적극적인 도입
  - 아마존의 Kiva 시스템
  - UPS와 FedEX (미국)
  - CJ 대한통운 (한국)

Figure 1. 물류 시스템에 활용되는 로봇 및 자율주행 시스템

연구 배경 및 동기

- **디지털 어소팅 시스템 (Digital Assorting System)의 역할과 중요성**
  - 편의점 물류를 담당하는 물류센터는 복수 제품에 대한 각 점포의 주문을 신속하게 처리해야함
  - 디지털 어소팅 시스템 활용하여 각 점포별 셀(cell)에 제품을 저장하지 않고 직접 분류 작업을 수행
  - 본 연구에서 자율주행카트를 활용하는 디지털 어소팅 시스템을 위한 주문묶음 알고리즘을 제안하고자 함

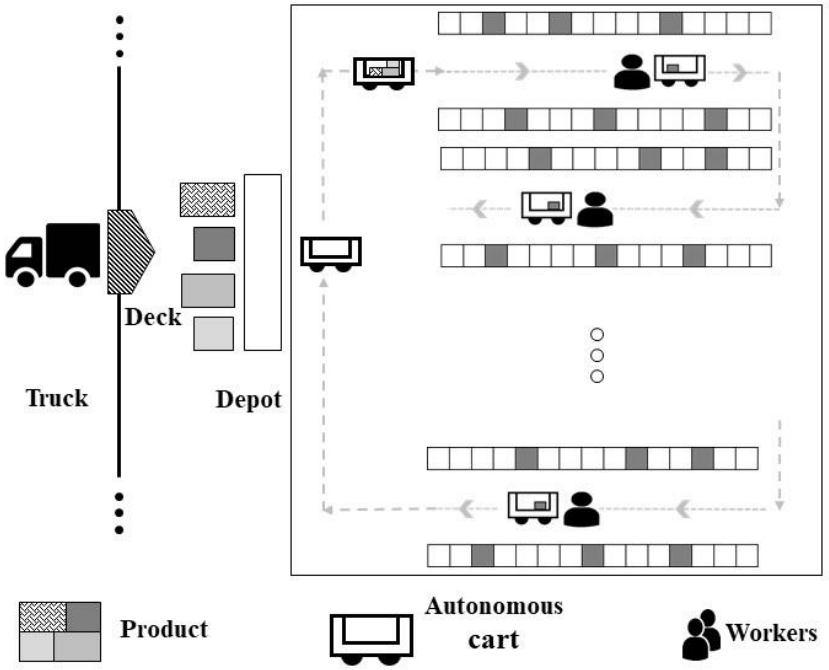

Figure 2. 자율주행카트를 활용한 디지털 어소팅 시스템에서의 분류 작업

문헌연구

물류센터에서의 분류 작업

- 컨베이어 시스템이 도입된 물류센터 내 분류 작업에 대한 연구
  ① **Johnson and Meller (2002)**: 자동화된 분류 시스템의 생산성 분석
  ② **Lee et al. (2015)**: 컨베이어 기반 분류 시스템의 생산성 분석

- 크로스 도킹 (Cross-docking)은 분류 작업과 관련된 물류센터의 중요한 기능
  ① **Nassief et al. (2018)**: 크로스 도킹 시스템 내 도크(dock) 할당 문제를 해결하기 위해 혼합정수계획법 기반 모델 제시
  ② **Yu et al. (2016)**: 크로스 도킹 시스템 내 인바운드와 아웃바운드 간 VRP (vehicle routing problem) 을 위한 모델 제시
  ③ **Enderer et al. (2017)**: 크로스 도킹 시스템 내 자재 취급 비용과 운반 비용 최소화를 위한 수리 모델 제시

문헌연구

▪ 주문묶음 알고리즘 연구

- 주문묶음 형성과 관련된 연구는 주문피킹 시스템 내 주요한 연구 주제임
- 주문묶음과 관련된 연구는 크게 4가지 종류로 분류:

Seed heuristics:
- **De Koster *et al.* (1999)**

Metaheuristics:
- Genetic algorithm: **Hsu *et al.* (2005)**, **Pan *et al.* (2015)**
- Simulated annealing: **Matusiak *et al.* (2014)**
- Tabu search: **Kulak *et al.* (2012)**
- Ant coloy optimization: **Li *et al.* (2017)**

Saving heuristics:
- **Clarke and Wright (1964)**
- **De Koster *et al.* (1999)**

Optimal approaches:
- **Gademann *et al.* (2001)**
- **Gademann and de Velde (2005)**
- **Hong *et al.* (2012b)**
- **Hong *et al.* (2017)**

문헌연구

- **자율주행카트 시스템**
  - 최근 물류센터 및 창고에서 로봇 시스템을 기반한 주문피킹 및 저장정책에 대한 연구가 대두

  - 물류센터 내 대표적인 자율주행 시스템으로 아마존 사의 Kiva 시스템이 존재
    ① **Lamballais et al. (2017)**: Kiva 시스템 내에서 Semi-open Queueing Networks (SOQN)을 활용한 물류센터 내 생산성 측정 모델 개발
    ② **Zou et al. (2018)**: Kiva 시스템 내 로봇의 배터리 교체를 반영한 생산성 측정 모델 제시

문제 정의

- **평행 통로형 디지털 어소팅 시스템에서의 분류 작업**
  - 자율주행카트(autonomous carts)를 활용한 평행 통로형 디지털 어소팅 시스템을 가정
  - 자율주행카트는 데포(depot)에서 제품을 싣고 운반함
  - 카트를 따르는 작업자는 카트 내 제품을 각 고객에게 할당된 셀의 상자에 분류

문제 정의

- **주문묶음을 활용한 분류 작업**
  - 시스템에 도착하는 대용량의 제품은 각각 점포 별로 분류하기 위해 통로 별, 점포 별로 나눠져야 할 필요성이 존재
  - 동일 고객을 대상으로 한 제품들을 묶어서 한 번에 분류한다면 카트의 이동거리 및 횟수를 줄일 수 있음
  - 제품을 통로 별로 나눈다면 제품을 통로 별 주문묶음으로 구성하여 자율주행카트의 이동거리를 줄일 수 있음

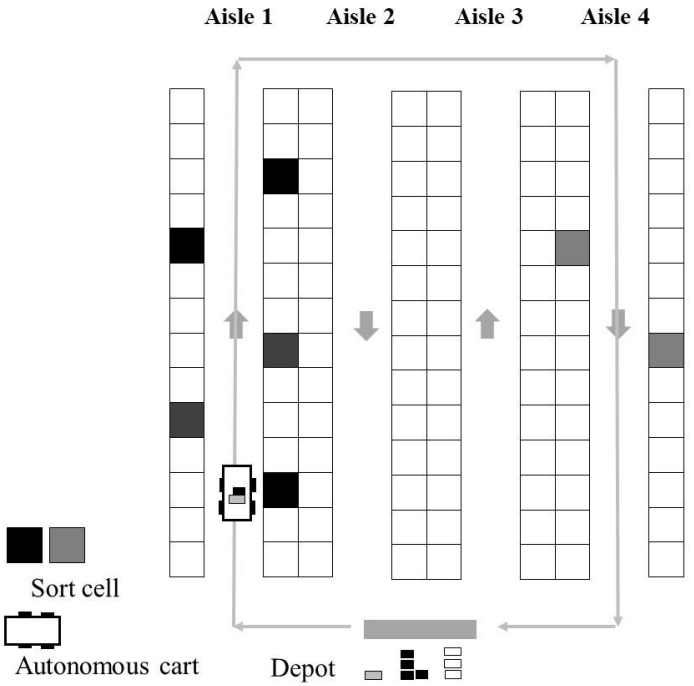

Figure 3. 주문묶음을 통한 디지털 어소팅 시스템 내 분류 작업

자율주행카트를 활용한 주문묶음 모델

- **Indices and sets**

P, p = the set of products and its index $p \in P$.

A, a = the set of aisles and its index $a \in A$.

B, b = the set of batches and its index $b \in B$.

R, r = the set of routes and its index $r \in R$.

- **Parameters**

$CAPA$ = the volume capacity of a cart.

$$RA_{ra} = \begin{cases} 1 & \text{if route } r \text{ passes through aisle } a, \\ 0 & \text{otherwise.} \end{cases}$$

LT_r = the length of route r.

V_{pa} = the volume of product p assigned to aisle a.

$$PA_{pa} = \begin{cases} 1 & \text{if product } p \text{ passes through aisle } a, \\ 0 & \text{otherwise.} \end{cases}$$

자율주행카트를 활용한 주문묶음 모델

- **Decision variable**

$$X_{pab} = \begin{cases} 1 & \text{if product } p \text{ 's part that should visit aisle } a \text{ is assigned to batch } b, \\ 0 & \text{otherwise.} \end{cases}$$

$$Z_{pa} = \begin{cases} 1 & \text{if at least one part of product } p \text{ is assigned to batch } b, \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_{br} = \begin{cases} 1 & \text{if batch } p \text{ takes route } r, \\ 0 & \text{otherwise.} \end{cases}$$

자율주행카트를 활용한 주문묶음 모델

- **Formulation**

$$\min \sum_{b \in B} \sum_{r \in R} LT_r \cdot Y_{br}, \tag{1}$$

subject to

$$X_{pab} \le Z_{pb}, \quad \forall p \in P, \forall a \in A, \forall b \in B, \tag{2}$$

$$\sum_{b \in B} X_{pab} \ge 1, \quad \forall p \in P, \forall a \in A, \tag{3}$$

$$\sum_{p \in P} \sum_{a \in A} V_{pa} \cdot X_{pab} \le CAPA, \quad \forall b \in B, \tag{4}$$

$$\sum_{r \in R} Y_{br} = 1, \quad \forall b \in B, \tag{5}$$

$$X_{pab} \cdot PA_{pa} \le \sum_{r \in R} RA_{ra} \cdot Y_{br}, \quad \forall p \in P, \forall a \in A, \forall b \in B, \tag{6}$$

$$X_{pab} \in \{0,1\},\ Y_{br} \in \{0,1\},\ Z_{pb} \in \{0,1\}. \quad \forall p \in P, \forall a \in A, \forall b \in B,$$

휴리스틱 알고리즘

- **Modified Seed algorithm**
  - 주문피킹(order picking)에서 활용되는 Seed algorithm 이용
  - 모든 제품은 통로 수에 맞게 나눠져 있다고 가정

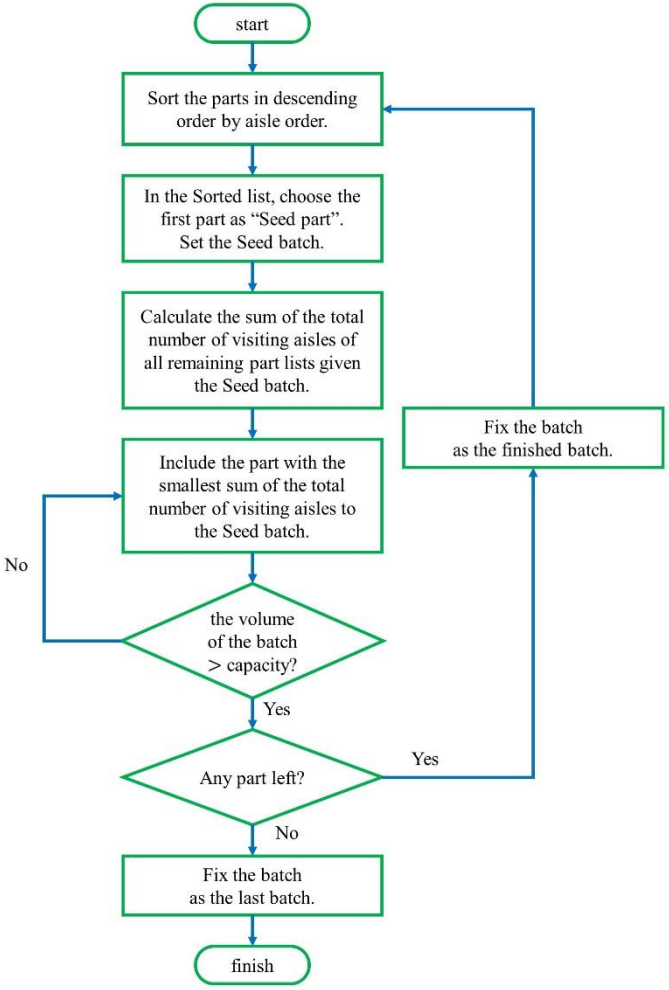

Figure 4. Modified Seed algorithm의 순서도

휴리스틱 알고리즘

- **Shortest Route First with First-fit Decreasing**
  - 모든 제품은 통로 수에 맞게 나눠져 있다고 가정
  - Shortest Route First with First-fit Decreasing 알고리즘은 두 단계로 구성
    ① 1 단계: 각 이동경로 별 작업자가 이동하면서 제품을 분류할 수 있는 제품 후보 리스트 작성
    ② 2단계: 각 이동경로 별 제품 후보 리스트를 이용하여 주문묶음 형성

휴리스틱 알고리즘

- **Shortest Route First with First-fit Decreasing**
  - 1 단계: 각 이동경로 별 작업자가 이동하면서 제품을 분류할 수 있는 제품 후보 리스트 작성
  - 2단계: 각 이동경로 별 제품 후보 리스트를 이용하여 주문묶음 형성

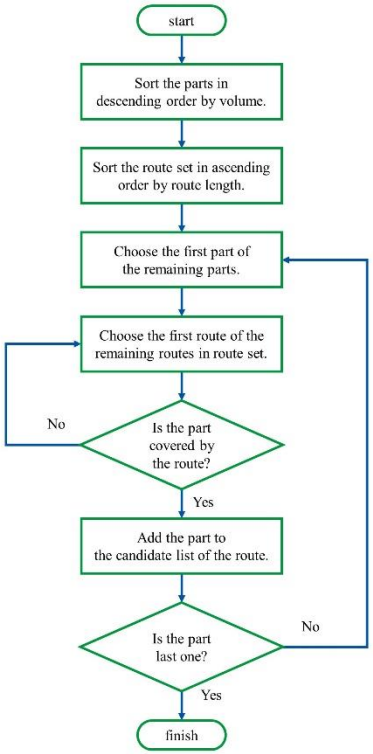

Figure 5. Shortest Route First with First-fit Decreasing의 1단계

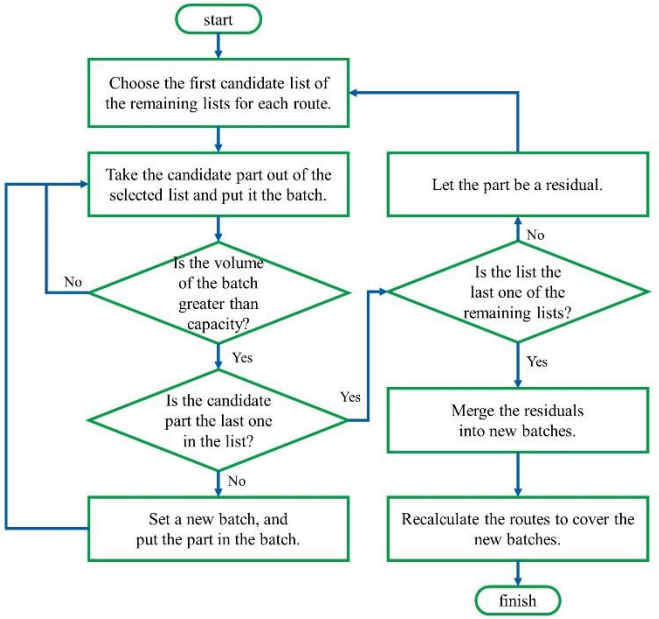

Figure 6. Shortest Route First with First-fit Decreasing의 2단계

실험 결과

실험 설계

Table 1. 실험 가정사항

항 목	내 용
제품 수 ($\|P\|$)	20, 40, 60, 100, 500, 1000
제품의 부피	Uniform[0.5, 5.0]
이동경로 규칙	One-way traversal routing method
통로 수 ($\|A\|$)	4 통로, 6 통로
통로 길이 (LT)	15 m
자율주행카트의 용량 ($CAPA$)	20, 30, 40 리터

- 실험 환경
  - ① IBM CPLEX 12.6 (Concert Technology in IBM JAVA)
  - ② Tecnomatix Plant Simulation 12
  - ③ AMD Ryzen5 2400G @ 3.60GHz / RAM 16.0GB
  - ④ 실험 인스턴스 수: 실험 당 10 회

- 측정 지표 및 표기

Opt	: 자율주행카트를 활용한 주문묶음 모델의 해
SRFwithFFD	: Shortest Route First with First-fit Decreasing
Seed	: Modified Seed algorithm
Obj.	: 카트의 총 이동거리 (m)
# of batches	: 형성된 주문묶음 수 (EA)
CPU	: 계산 시간 (sec.)

실험 결과

계산 결과

Table 2. 소규모/중규모 주문에 대한 실험 결과

|A|	|P|	CAPA	Opt			SRFwithFFD			Seed		
			Obj.	# of batches	CPU	Obj.	# of batches	CPU	Obj.	# of batches	CPU
4	20	20	398.4	11.6	2.2	425.6	11.4	0.00	575.6	13.8	0.00
		30	269.3	7.9	1.7	294.7	7.6	0.00	460.1	9.9	0.00
		40	208.1	6.1	1.9	236.9	5.9	0.00	423.2	8.2	0.00
	40	20	761.4	21.0	13.8	808.1	22.7	0.00	975.8	25.9	0.00
		30	528.4	15.0	11.3	556.5	15.3	0.00	729.5	17.6	0.00
		40	392.4	11.2	10.4	423.9	11.4	0.00	627.4	14.2	0.00
	60	20	1163.6	34.2	809.2	1188.9	33.9	0.00	1387.6	37.9	0.01
		30	770	22.6	501.3	825.1	23.2	0.01	990.3	25.6	0.00
		40	578.6	16.9	326.4	631.3	17.5	0.00	818.9	20.0	0.00
6	20	20	598.8	17.4	7.6	630.2	16.3	0.00	882.2	19.8	0.00
		30	400.6	11.5	3.8	444.9	10.9	0.00	714.5	13.8	0.00
		40	301.6	8.8	4.2	354.8	8.2	0.01	649.3	11.1	0.00
	40	20	1159.8	33.9	2019.7	1206.5	33.3	0.01	1491.5	38.0	0.01
		30	761.8	22.1	1032.4	837.6	22.4	0.01	1112.3	25.5	0.00
		40	591	17.1	205.1	648.9	16.9	0.01	956.3	19.8	0.00
	60	20	1729.8	50.5	3618.3	1772.6	49.9	0.01	2104.7	55.7	0.01
		30	1157.2	33.8	2840.7	1233.7	34.1	0.01	1516.0	37.1	0.01
		40	859.8	25.1	1228.6	944.7	25.6	0.01	1255.9	28.5	0.01

- 4 통로 물류센터에서 40개까지, 6 통로 물류센터에서 20개 제품에 대해서 최적 해 도출 확인
- 취급하는 제품 수가 중규모 이상부터는 최적 해를 도출하는데 계산 상 많은 시간이 소요
- SRFwithFFD 의 경우 40개/60개 제품에서 최적 해 대비 2.2 % ~ 10.0 % 의 적은 차이로 주문묶음 구성

실험 결과

- **대규모 주문에 대한 계산 결과**

Table 3. 대규모 주문에 대한 실험 결과

|A|	|P|	CAPA	SRFwithFFD			Seed		
			Obj.	# of batches	CPU	Obj.	# of batches	CPU
4	100	20	1806.0	52.0	0.01	2188.0	61.0	0.02
		30	1262.0	36.0	0.01	1508.0	41.0	0.00
		40	956.0	27.0	0.01	1206.0	31.0	0.02
	500	20	8810.0	258.0	0.16	10416.0	303.0	0.39
		30	6090.0	178.0	0.16	6880.0	199.0	0.26
		40	4594.0	134.0	0.16	5150.0	147.0	0.20
	1000	20	17548.0	515.0	0.62	20684.0	605.0	2.33
		30	12074.0	354.0	0.60	13582.0	395.0	1.53
		40	9116.0	267.0	0.60	10114.0	293.0	1.13
6	100	20	1810.0	51.0	0.03	2268.0	60.0	0.02
		30	1368.0	38.0	0.02	1868.0	46.0	0.01
		40	1164.0	32.0	0.02	1562.0	37.0	0.01
	500	20	9052.0	264.0	0.38	10326.0	297.0	0.59
		30	6910.0	201.0	0.38	7746.0	220.0	0.44
		40	5550.0	161.0	0.37	6254.0	175.0	0.36
	1000	20	18062.0	529.0	1.40	20500.0	594.0	3.51
		30	13710.0	401.0	1.39	15264.0	440.0	2.58
		40	11126.0	325.0	1.39	12208.0	349.0	2.06

- SRFwithFFD 이 Seed에 비해서 총 카트 이동거리가 약 9.7 % ~ 36.5 % 짧은 결과를 보임
- SRFwithFFD 은 또한, 주문묶음의 경우 7.4 % ~ 21.1 % 더 적은 수를 생성

결론 및 향후 연구

결론

- 자율주행카트를 도입한 디지털 어소팅 시스템의 주문묶음 모델 및 알고리즘 제안
- 혼합 정수 계획법을 활용하여 디지털 어소팅 시스템의 주문묶음을 설명
  ① 일반적인 Bin-packing problem을 해결하기 위한 First-fit Decreasing 알고리즘을 이용한 휴리스틱 제안
  ② 주문피킹에서 활용되는 Seed algorithm에 기반한 휴리스틱 알고리즘 제안
- 자율주행카트를 도입할 경우 주문묶음을 형성하여 카트의 총 이동거리를 줄일 수 있으며, 필요에 따라서 제품을 통로 별로 나누어 보다 효과적인 주문묶음을 형성할 수 있음

향후 연구

- 본 연구에서는 자율주행카트 간 정체 (blocking) 현상을 고려하지 않음
- 자율주행카트 간 정체(blocking)을 고려한 주문묶음 알고리즘 개발 및 검증 필요

THANK YOU

Jeongman Lee
Tel. : (+82) 51-510-1539
Email : dbdbpqpq11@naver.com

Soondo Hong
Tel. : (+82) 51-510-2331
Email : soondo.hong@pusan.ac.kr

Operation Analytics Laboratory
Department of Industrial Engineering
Pusan National University, Busan, Korea

Acknowledgement:
This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A09000648) and by a grant from the R&D Program of the Korea Railroad Research Institute, Republic of Korea.