Splitting shopping and delivery requests in an on-demand personal shopper service

Alp Arslan, Niels Agatz and Mathias A. Klapp

Departamento de Ingeniería de Transporte y Logística,

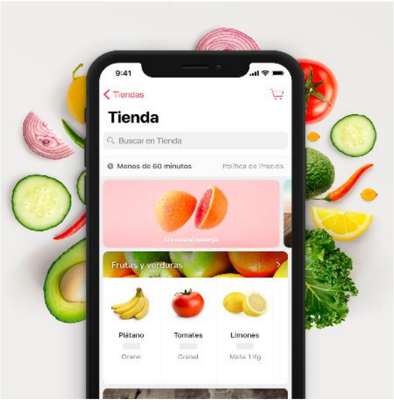
Departamento de Ingeniería Industrial y de Sistemas,

Pontificia Universidad Católica de Chile

maklapp@ing.puc.cl

- **Motivation: online personal shopper services**
- Our strategy
- Solution method
- Experiments
- Takeaways

Motivation: online personal shopper services



PS services are <u>intermediaries</u> who:

1. **receive** online customer requests (a shopping list),
2. **shop and pick-up** items available at local retailer stores,
3. **and deliver** these to the customer within a short deadline (e.g. 2 hours).

- Idea:

  convenience of online shopping + product availability at stores.

Increasingly popular for grocery delivery

- Biggest PS service provider: **Instacart** (US)
  - ≈$8 billion market value, 300 retailes partners, operates in 50 US states
- **Postmates** (US), **Deliv** (US), **Rappi** (Colombia), **Cornershop** (Chile - México), **Glovo** (Spain), ….

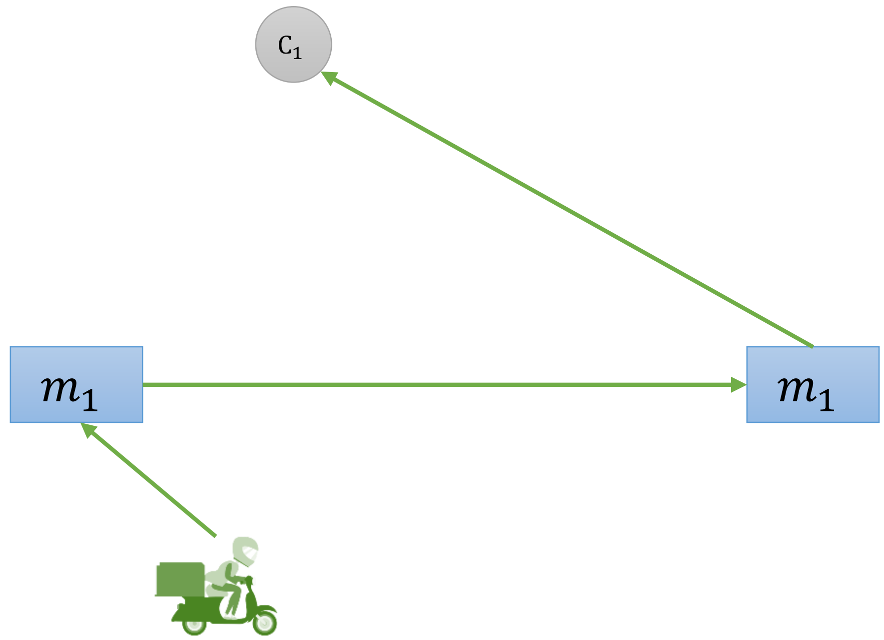
- Similarity to meal delivery services (**Grubhub**, **UberEats**, **Foodora**)

PS service business model

- A PS service is a **store aggregator**:
  - Offers products of affiliated Brick & Mortar stores.
  - Google Shopping: 50 merchants: Costco, Target, Walgreens…
  - Asset-light business (no inventory).

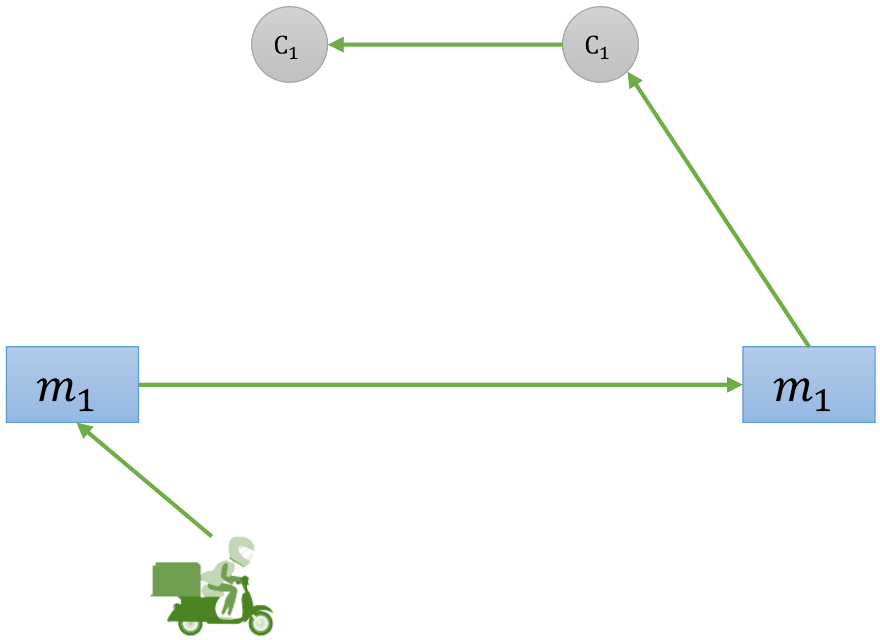
- Also, it is a **logistics service provider**:
  - Online platform accepts customers' shopping requests.
  - Automatic dispatcher assigns accepted requests shoppers.
  - **Shopper:** Shops and delivers items to customers.

Simple strategy: One request per shopper at a time



Customer c_1 orders from stores m_1 and m_2.

An improvement: consolidate if possible

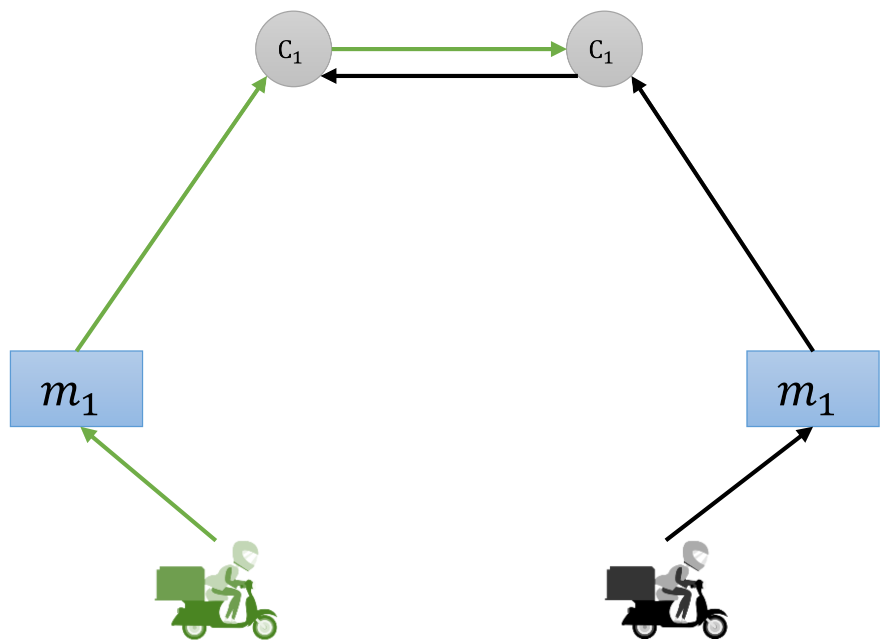


Customers c_1 and c_2 order from stores m_1 and m_2.

But: tight delivery deadlines → limited consolidation options.

- **Motivation: online personal shopper services**
- **Our strategy**
- Solution method
- Experiments
- Takeaways

Our strategy: split requests & deliver in parallel



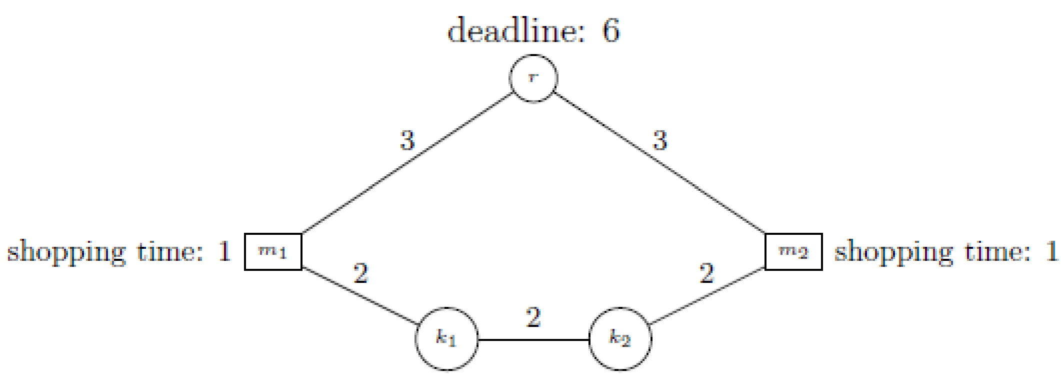
- We study splitting the service of requests involving shopping at multiple stores into separate **tasks** served by different shoppers.

Less granularity & more flexibility

Dividing requests into smaller task may create:

Packing benefits: increased fleet time utilization & capacity.

- Particularly relevant when delivery deadlines are tight.

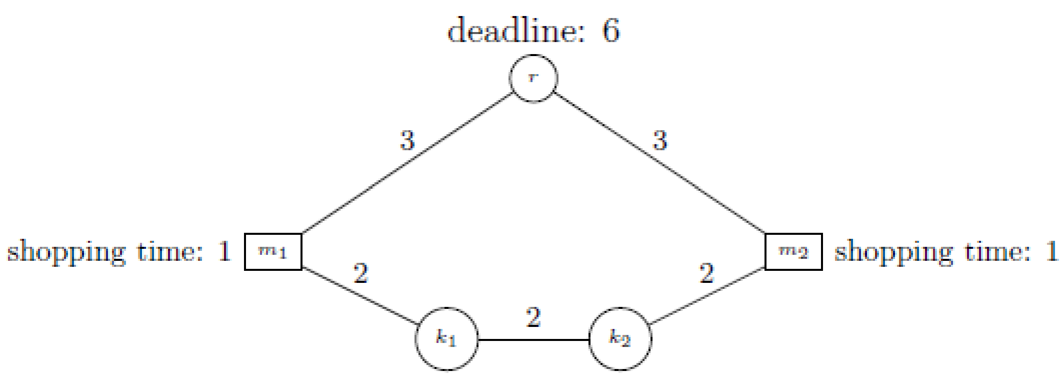


- An on-time service is infeasible without splitting ($t \geq 13$).
- If the request is split into two tasks, then two shoppers can deliver by $t = 6$.

Less granularity & more flexibility

Dividing requests into smaller task may create:

Routing benefits: a larger set of routing options may require less travel time.



- Single shopper total travel 11 time units
- Two shoppers total travel 10 time units

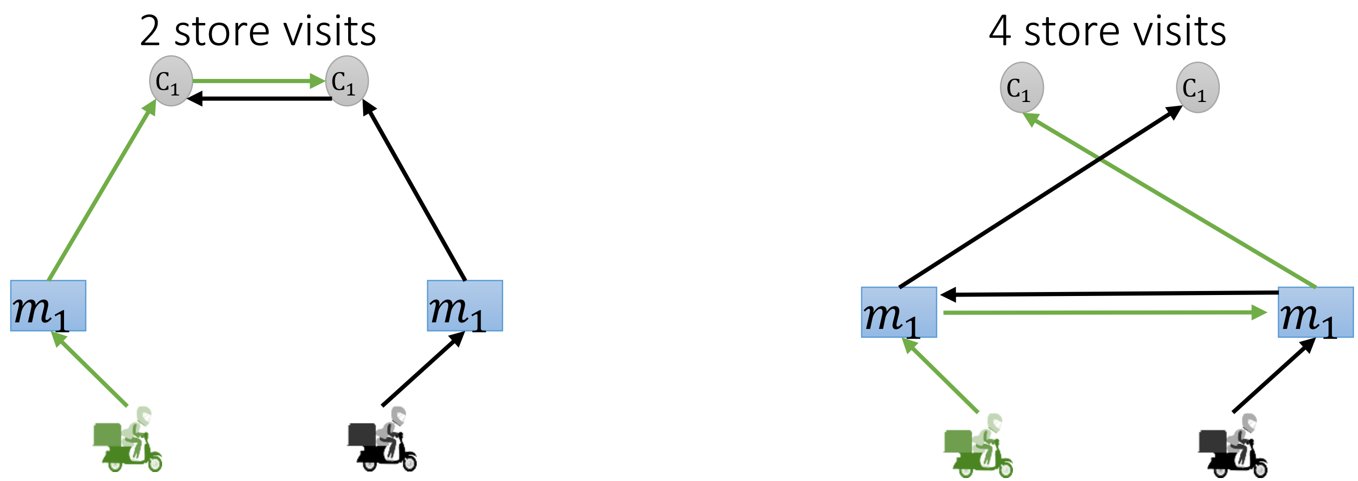
Less granularity & more flexibility

Dividing requests into smaller task may create:

Shopping benefits: save fixed shopping times by consolidating in one shopper multiple tasks originated in a common store.

$$\text{Shopping time} = f + \sum_{s \in S} v_s$$

- Variable **shopping time** is unavoidable, but we could save store visits (parking, queuing, walking to store).



Related Literature

- **Pick-up and delivery:** Salvelsbergh and Sol, (1995), but dynamic & multiple pick-ups per delivery.
- **Split delivery routing problem:** Archetti et al. (2008), Nowak et al. (2009): similar flexibility principle, different problem.
- **Same-day delivery**: Arlsan et al. (2019), Klapp et al. (2018), Voccia et al. (2017), Ulmer (2018): same-day delivery with multiple pickup locations.
- **Meal delivery problem:** Reyes et al. (2018); Ulmer et al. (2017); Yildiz and Savelsbergh (2017); Steever et al. (2019).
    - Relatively more constrained, different objectives

Problem Statemet

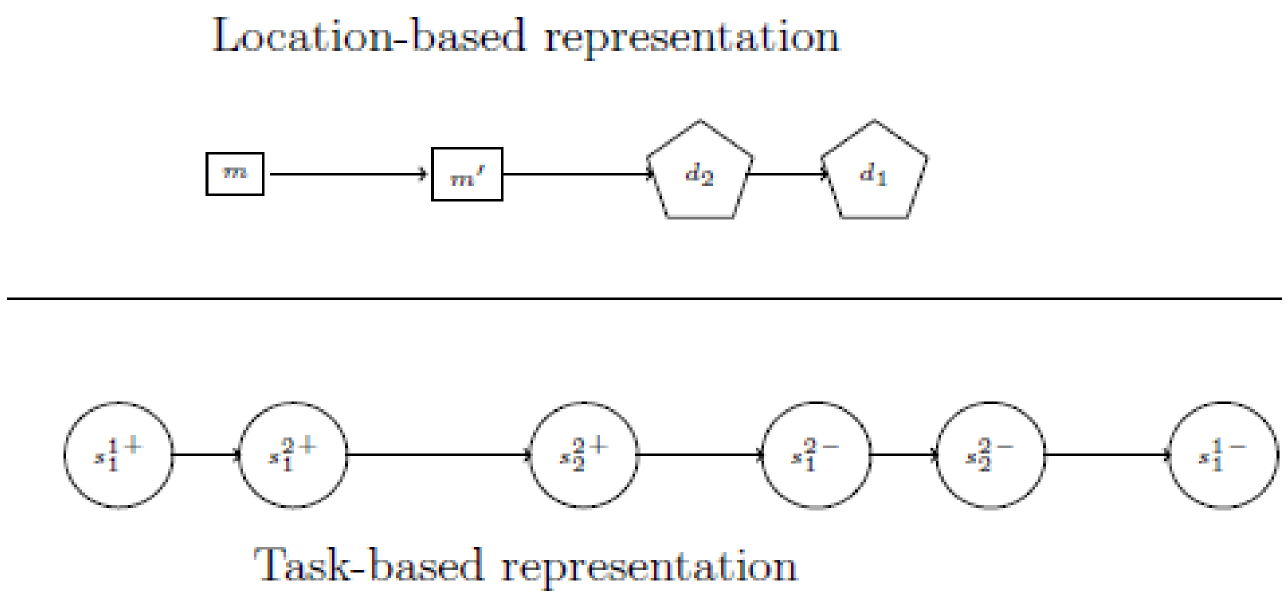
- A service period T in which customers place requests.
- Set of partner retailer stores M.
- Dynamically arriving customer requests $r \in \{1, \dots, n_R\}$ with:
  - Required delivery location
  - Shopping list from one or more stores in M
  - Order placement time e_r
- System-wide delivery deadline L. Latest delivery time $e_r + L$
- Fleet of shoppers K

- Motivation: online personal shopper services
- Our strategy
- Solution method
- Experiments
- Takeaways

Tasks and graph representation

- A customer's shopping list is a collection S_r of tasks with common delivery location.
- A **task** is a pair of shopping s^+ and delivery s^- nodes.

- **Task based network representation:**
  - can model fixed and task-dependent store times as arc costs.



Model

- Assume event-based sequential decisions triggered by request arrivals with no prior future knowledge (pure online problem).

- State at decision time t:
  - set of active tasks S_t, *i.e.* accepted but not yet served,
  - shopper status: location, earliest departure and load info,
  - delivery plan: a pick-up and delivery trip per shopper.

- Decisions at time t:
  - accept or not: we accept when it is feasible,
  - Update and execute delivery plan until next decision time.

- Objective: Maximize number of requests served on-time.

A rolling horizon framework

Our solution:

- solve **routing problem (PsDPd)** before each acceptance decision to identify a feasible plan covering new and active tasks.
- If such a plan is found, then accept and update plan.

- **PsDPd**: pick-up and delivery routing problem, but
  - Multiple pick-ups per request and split delivery,
  - service deadlines,
  - considers current state of shoppers & en-route assignments.
  - Minimize total shopping and travel time.

PsDPd solution approach #1: Exact approach

1. Partition all active tasks among the shoppers
2. Test feasibility with DP labeling algorithm similar to Tilk a

   - Ham eadlines per s

- Algorit
  - **Lowe** bent.
  - **Specific label domination** rules acknowledging that distances between pick-ups in a common store have no path dependency.

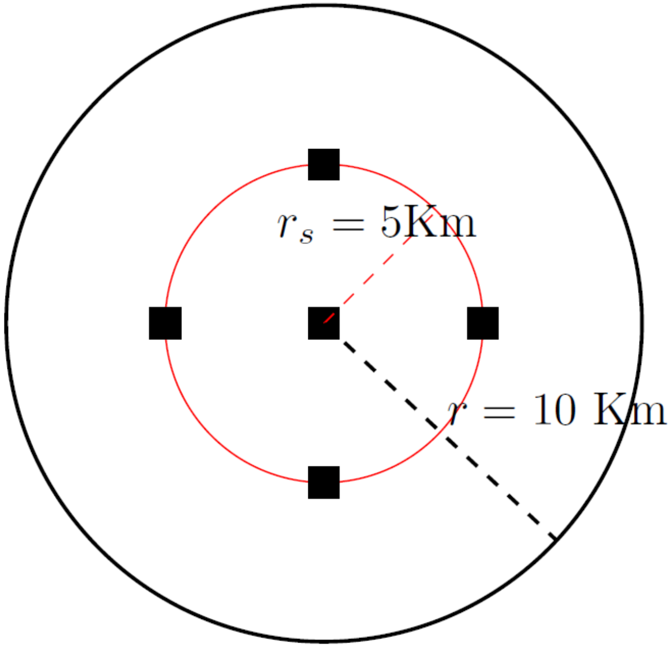
PsDPd solution approach #2: PlanMaker heuristic

- Split new request into tasks.
- Sequential cheapest insertion by task.
- Adaptive large neiborhood search (ALNS):
  - Removal operators: partial destruction of solution by removing certain tasks.
  - Repair operators: reinsert the removed tasks.
- Choose repair operators according to dynamically updated weights based on success.

- Motivation: online personal shopper services
- Our strategy
- Solution method
- Experiments
- Takeaways

Computational Experiments

- 10km radius area, five stores located within a 5Km radius.
- Uniformly distributed customers over space and time.
- 10 hours service period.
- Each request demands shopping from three randomly chosen stores
- Five shoppers, max capacity 10 tasks.



Tested operational policies

- **One by one (1b1):** Each shopper serves one single customer request at a time.

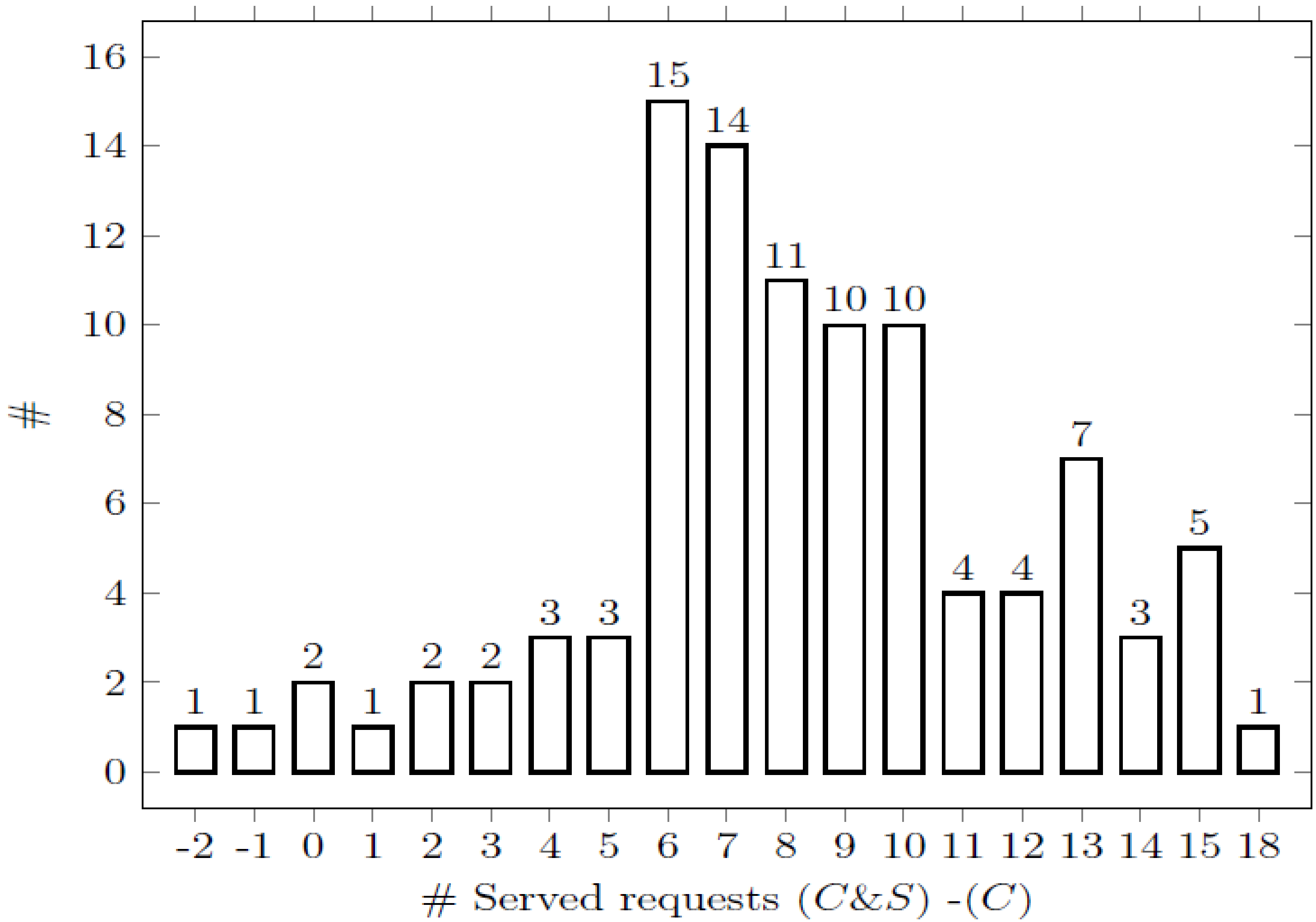
- **Consolidation (C):** A shopper can simultaneously serve multiple requests, but all tasks of a single request are served by one single shopper.

- **Consolidation & Splitting (C&S)**: Requests can be split into different tasks that can be served by multiple shoppers in parallel. Also, shoppers can simultaneously serve tasks of multiple requests.

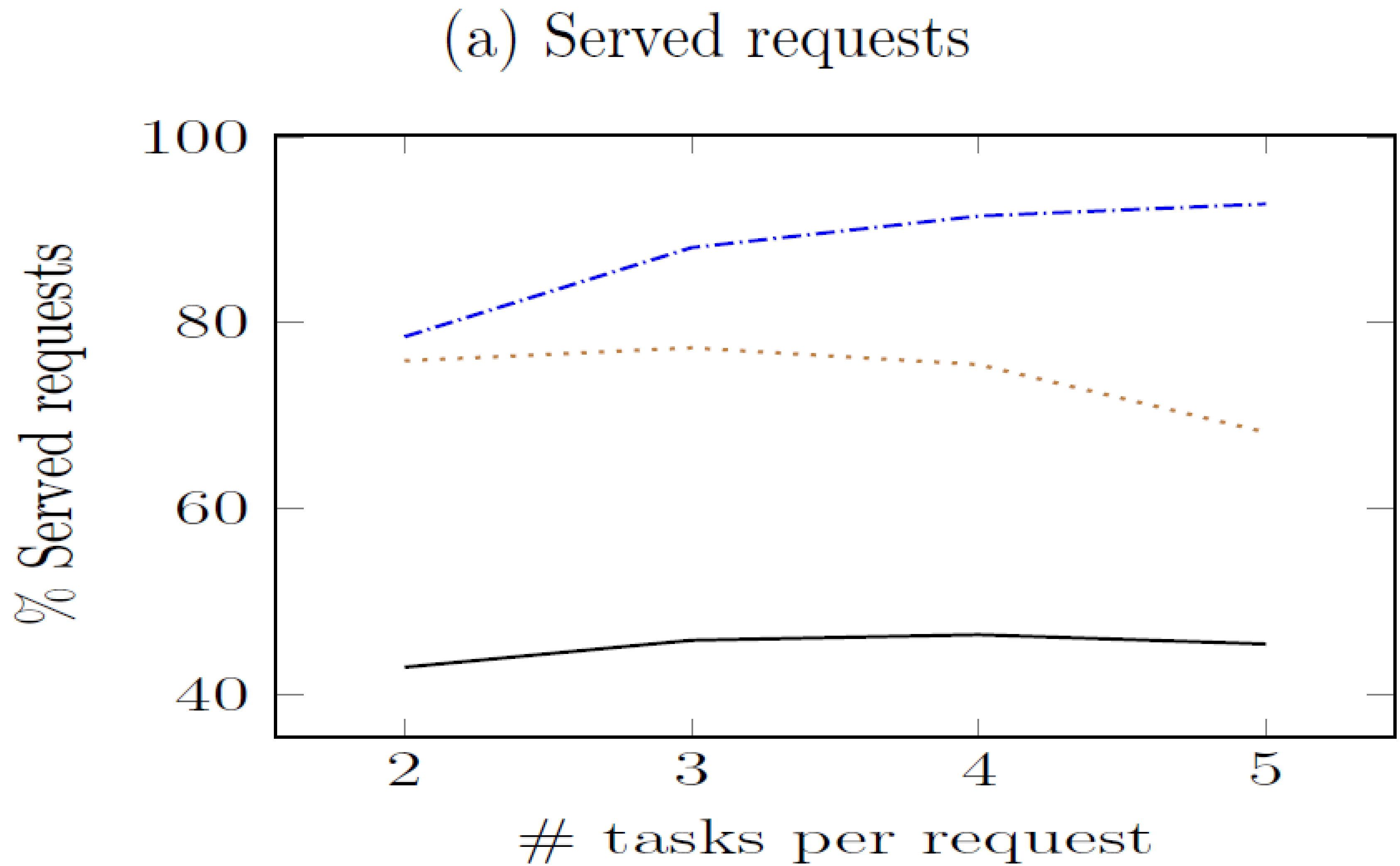
Base Case Results ($L = 90$min)

	$1b1$	C	$C\&S$
Served requests (%)	45.8	77.3	88.1
Request split. (%)	0	0	69.2
Delivery interval (min.)	0	0	23.6
Time per req (min.).	51.2	28.9	25.1
Shopping time per req (min.)	30.0	15.0	10.3
Travel time per req (min.).	21.2	13.9	14.8
# locations visited per req.	4.0	2.3	2.7
CtD (min.)	77.1	78.2	77.3

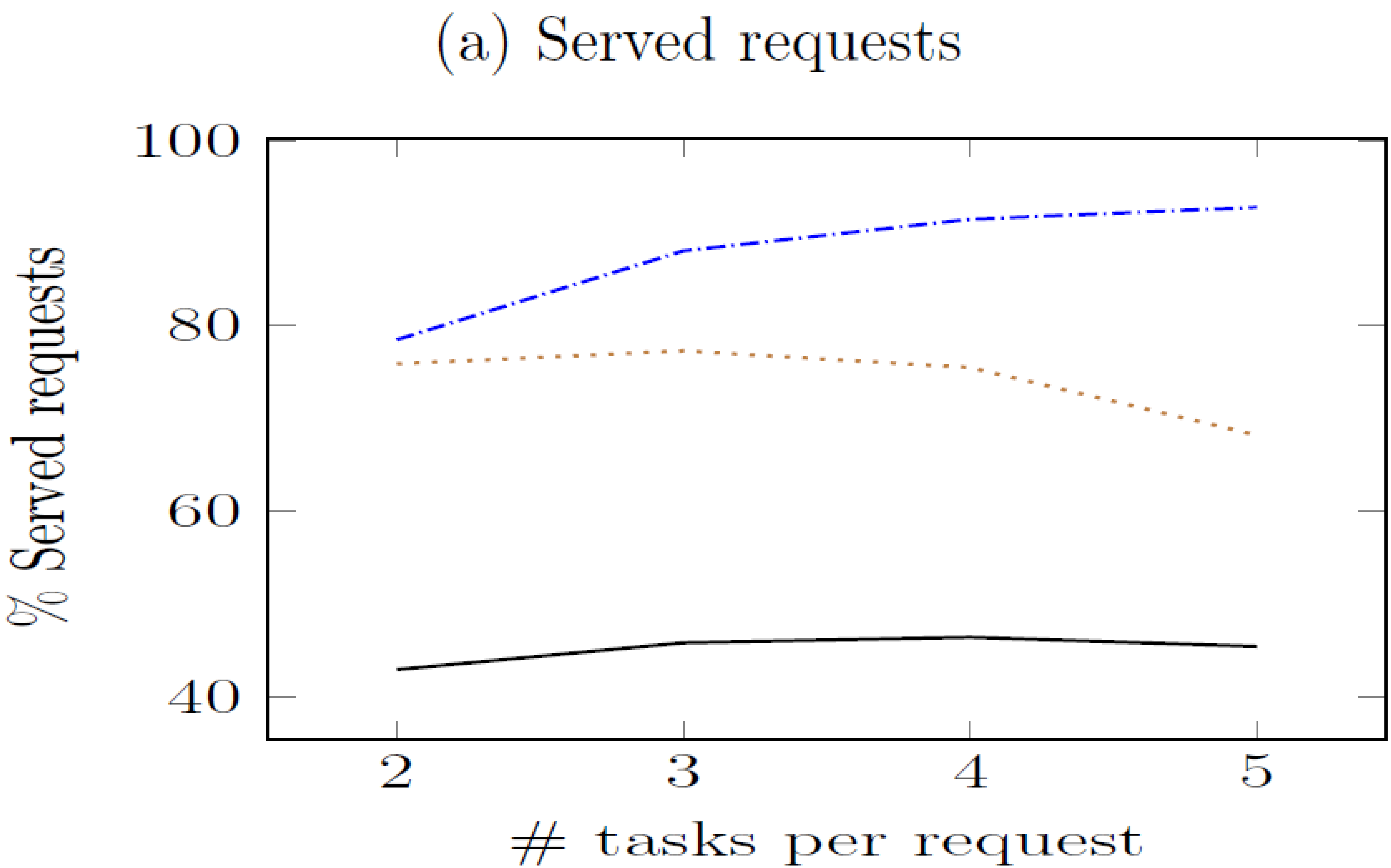
Base Case Results ($L = 90\text{min}$)



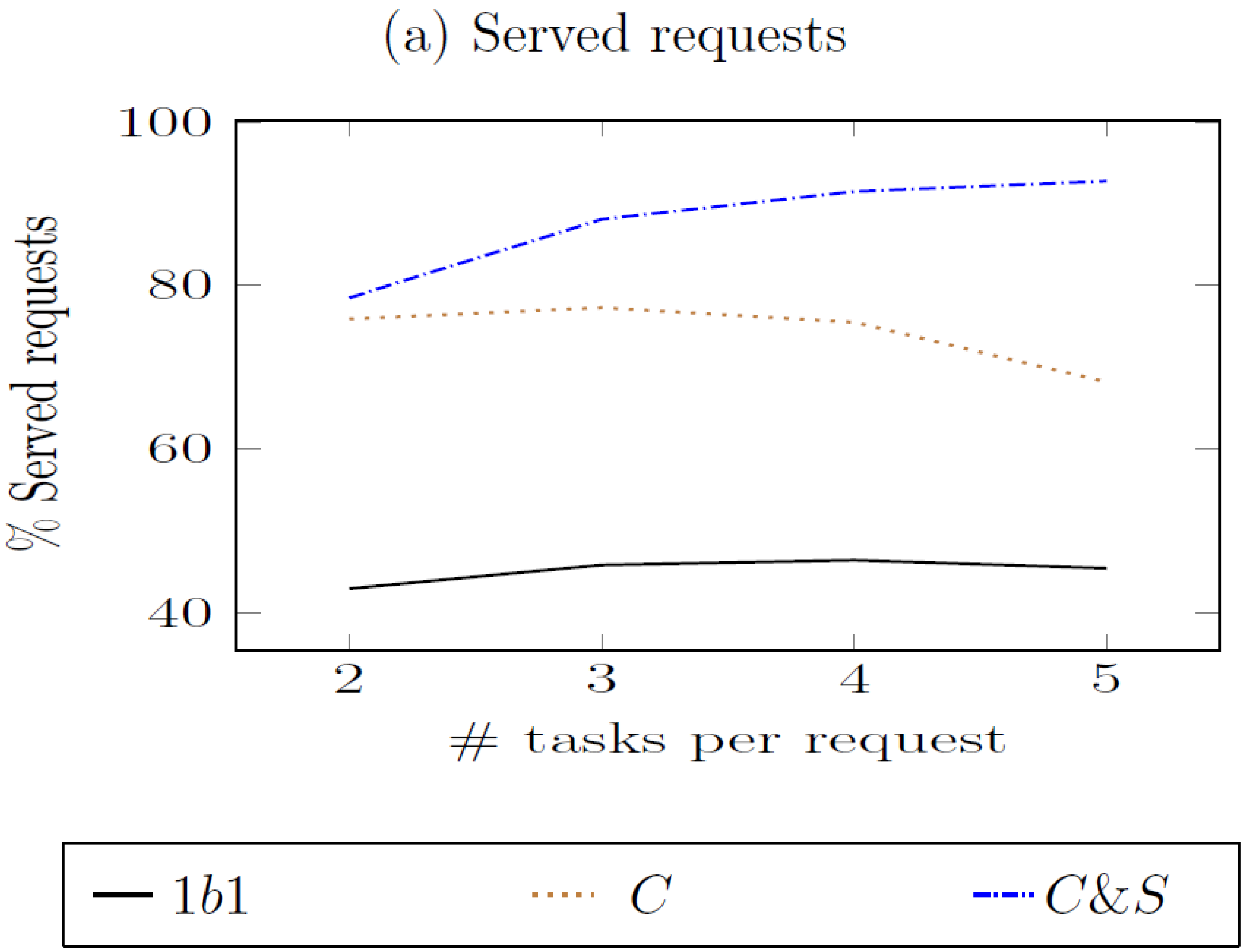
Sensitivity: # tasks per request



Sensitivity: # tasks per request

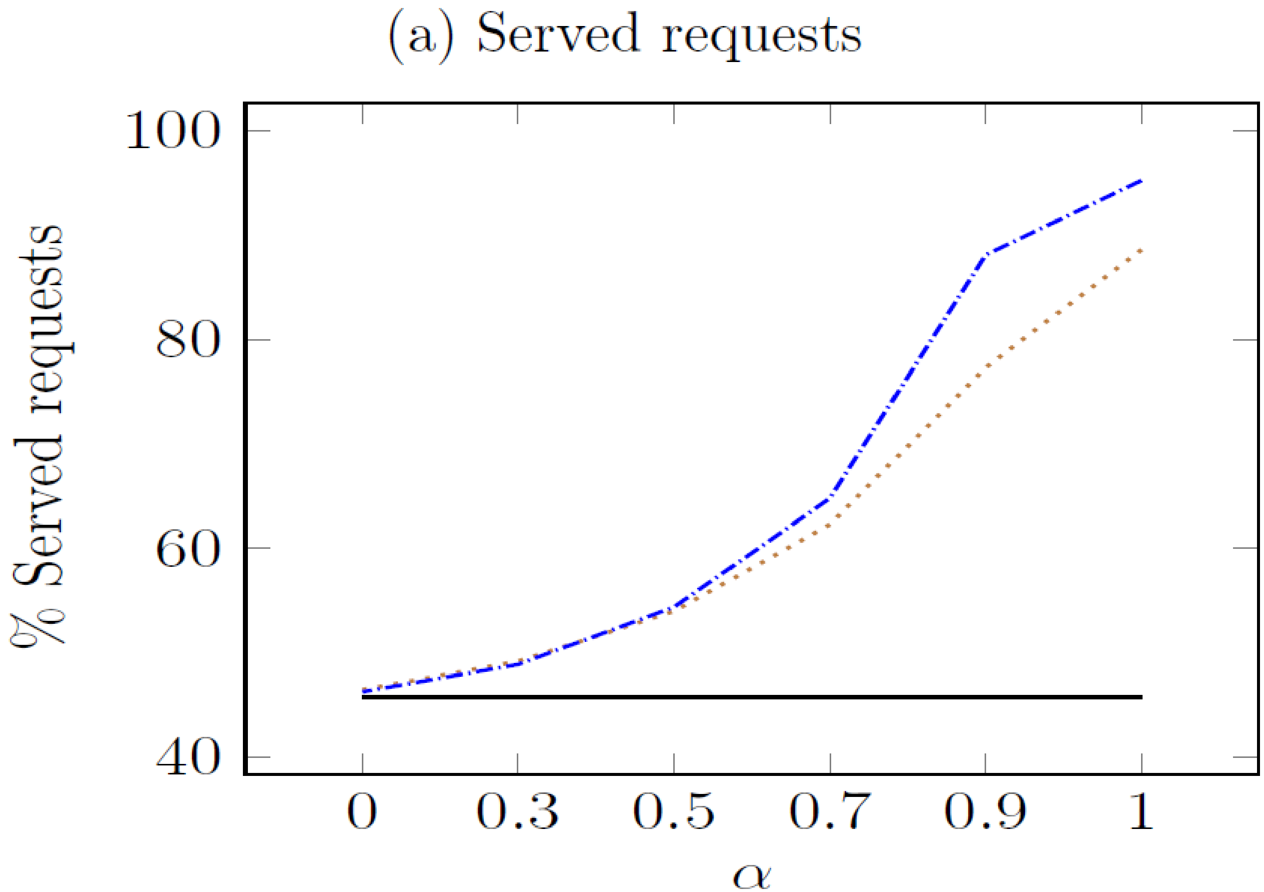


Sensitivity: # tasks per request



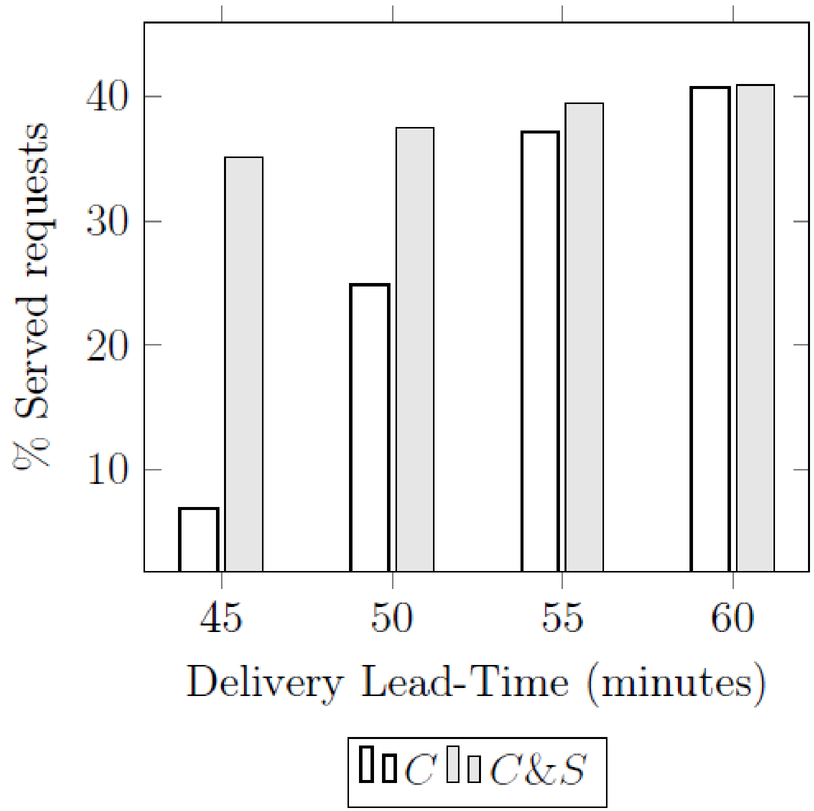
Sensitivity: shopping economies at stores (α)

- $\alpha = 0$: shopping time proportional to tasks collected.
- $\alpha = 1$: fixed shopping time per store visit.



Sensitivity: delivery deadline (and packing benefits)

Figure 7: Packing Benefits of Request Split, $\alpha = 0$, **80 Requests, 3 Shoppers**

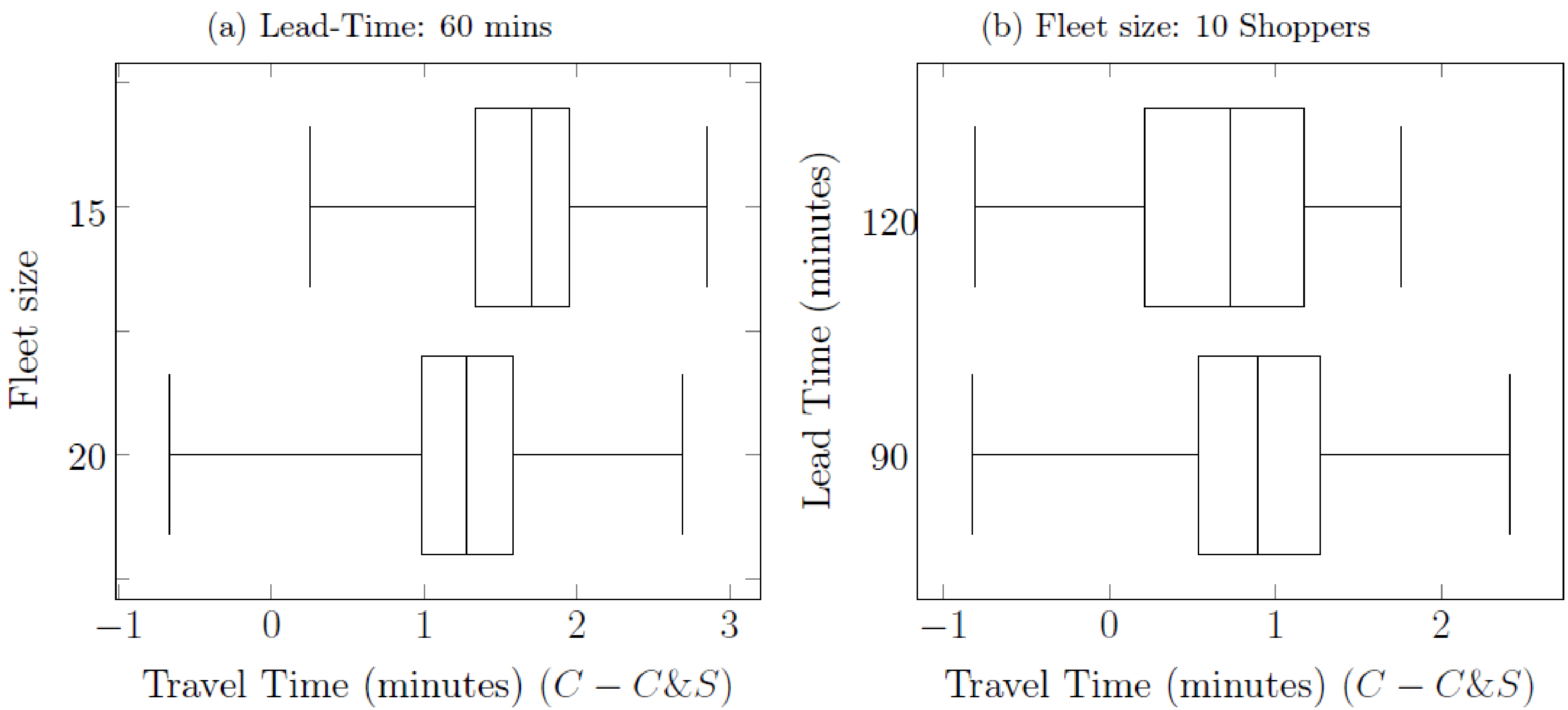


- We focus specifically on packing and routing benefits (no shopping economies)

Routing Benefits

- We focus specifically on routing benefits (no shopping economies, enough capacity to serve 100%)

Figure 8: Routing Benefits of Request Splits When No Packing Benefits: $\alpha = 0$, **80 Requests**



- Motivation: online personal shopper services
- Our strategy
- Solution method
- Experiments
- Takeaways

Key Takeaways

- Request splits increase the percentage of served requests.
- On average customers also receive faster delivery.
- Benefit mostly obtained due to an increased shopper utilization, reduced shopping times, and cheaper routing options available.
- Benefits of splitting increase for relatively more time constrained systems with stronger shopping economies.

Future Work:

- Probabilistic information about the future and proactivity.
- Splits and transfers?
- Separating shopping and delivery?

Questions?

``Splitting Shopping and Delivery Tasks in an On-Demand Personal Shopper Service''

Draft available at https://dx.doi.org/10.2139/ssrn.3428912

maklapp@ing.puc.cl

Splitting shopping and delivery requests in an on-demand personal shopper service

Alp Arslan, Niels Agatz and Mathias A. Klapp

Departamento de Ingeniería de Transporte y Logística,

Departamento de Ingeniería Industrial y de Sistemas,

Pontificia Universidad Católica de Chile

maklapp@ing.puc.cl

PlanMaker validation in small instances:

n_s	Av. Opt. gap (%)	# Optimal	$\triangle$ Feasibility
8	1.3	4/5	0
9	1.5	4/5	0
10	4.0	3/5	0
11	0.8	4/5	0
12	2.1	3/5	0
13	2.8	4/5	0
14	3.7	1/5	0
15	3.1[1]	0/5	1

Para responder a su pregunta:

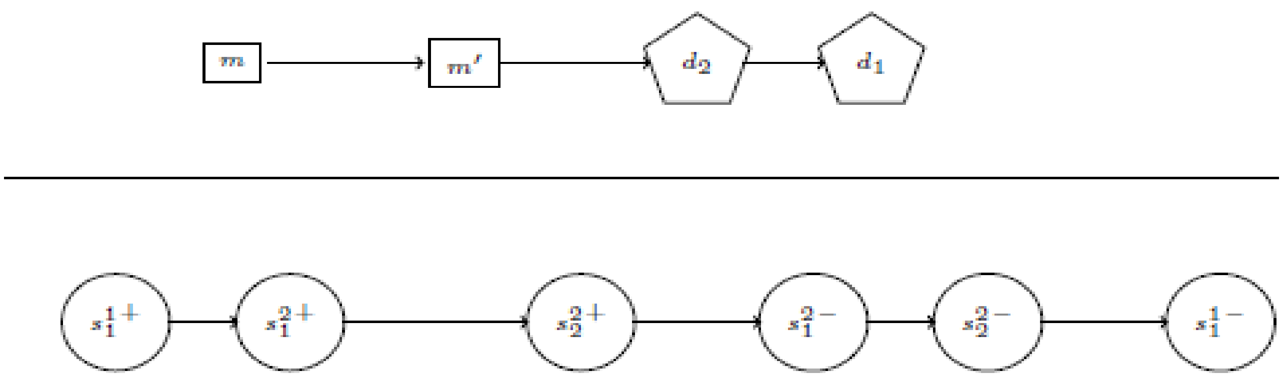
Table 4: Empirical Distribution of click-to-door time and delivery Interval for the *C&S* policy.

	click to door (CtD) non-split	split	All	delivery interval
minimum	15.0	37.8	15.0	0.01
Q1	71.4	72.9	71.7	7.3
Q2	81.7	82.2	81.8	18.3
Q3	87.2	86.8	87.3	34.9
maximum	89.9	89.9	89.9	85.7
average	77.2	78.1	77.3	23.6

Para responder a su pregunta:

$$c_{ij} := \mathbb{1}_{\left(i \overset{\mathrm{L}}{\neq} j\right)} t_{ij} + \mathbb{1}_{(j \in S^{+})} \left(c_j^p + \mathbb{1}_{\left(i \overset{\mathrm{L}}{\neq} j\right)} c_j^f \right),$$

Location-based representation



Task-based representation