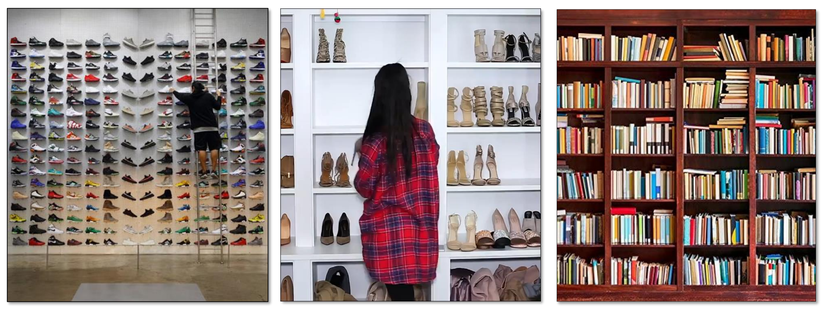
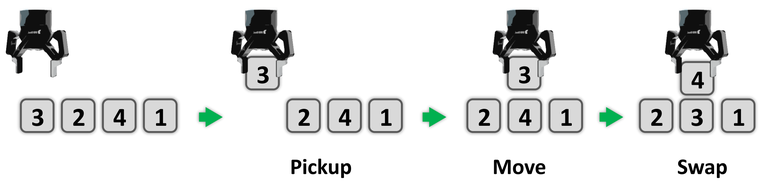
# Rearrangement on Lattices with Pick-n-Swaps: Optimality Structures and Efficient Algorithms

**Jingjin Yu** · Computer Science @ Rutgers University at New Brunswick

## Rearrangement on Lattices with Pick-n-Swaps



Efficient solutions for rearrangement tasks in lattice-like settings find many practical applications, including the rearrangement of products at stores and showroom, the sorting of books on bookshelves, inventory management in autonomous vertical warehouses, to list a few.




In this work, we examine the ensuing structures resulting from the **pick-n-swap manipulation primitive**, where the end effector may pick up, place down, and swap items. To accomplish the task using pick-n-swaps, the robot must carefully plan a sequence with which the items are picked and subsequently placed, to minimize the number of pick-n-swaps $N$ and end-effector travel, which is usually easier to execute than pick-n-swaps and less time-consuming. That is, we perform a **sequential optimization** in this study.

Plan $\quad \boldsymbol{P} = \{p_0, p_1 \cdots, p_N, p_{N+1} = p_0\}$

Objective $\quad \min_{\boldsymbol{P}} J(\boldsymbol{P}) = N\boldsymbol{c_p} + \sum_{i=0}^{N} dist(p_i, p_{i+1})\boldsymbol{c_t}$

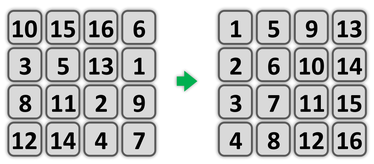
$p_i$: $i$-th pick-n-swap location on a lattice.
$c_p$: single pick-n-swap cost. $c_t$: unit travel cost.

We study four variants in detail:

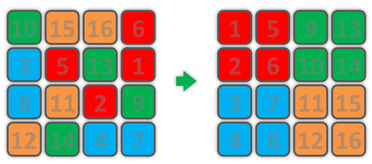
**Labeled One-dim. Rearrangement (LOR)**

**Partially-labeled 1D Rearrangement (POR)**

**Labeled Two-dim. Rearrangement (LTR)**



**Partially-labeled 2D Rearrangement (PTR)**

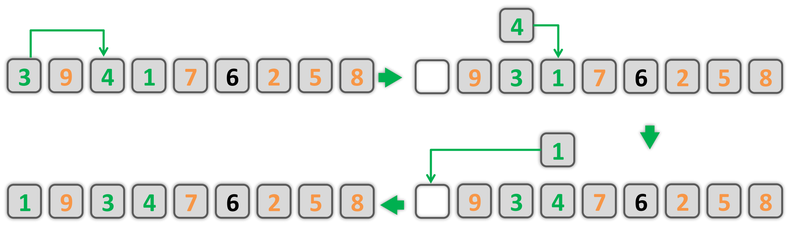


## (Structurally) Related Work


- K. Treleaven, M. Pavone, and E. Frazzoli, "Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems," IEEE Transactions on Automatic Control, vol. 58, no. 9, pp. 2261–2276, 2013.
- S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps," The International Journal of Robotics Research, vol. 37, no. 13-14, pp. 1775–1795, 2018.
- A. Gal and P. B. Miltersen, "The cell probe complexity of succinct data structures," Theoretical computer science, vol. 379, no. 3, pp. 405–417, 2007.
- E. Curtin and M. Warshauer, "The locker puzzle," The Mathematical Intelligencer, vol. 28, no. 1, pp. 28–31, 2006.


## Structure and Optimal Algorithm for LOR

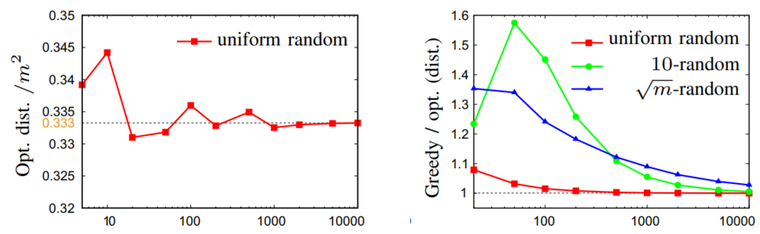
A key structural insight is that lattice rearrangement problem induces **cycles**. For example, in the following LOR problem, there are two cycles, (341) and (98572). The problem may be solved by **cycle-following**: for (341), we pick up 3, swap with 4, then swap with 1, and then put 1 at its goal. Same can be down for (98572). *The cycle-following simple procedure gives us the optimal number of pick-n-swap operations for labeled problems.*



We note that, if we further perform **cycle-switching** operations, end-effector travel distance can sometimes be reduced. In the example below, cycle-following incurs 14 units of end-effector travel. If we switch from the (41) cycle to the (53) cycle in the middle, we can reduce the distance traveled to 10.

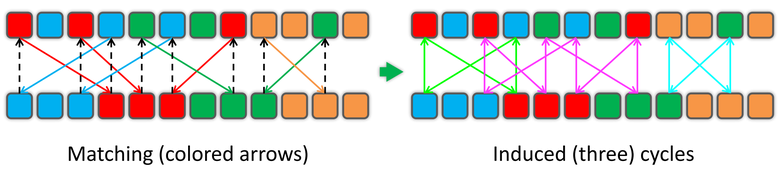
Total distance: **14** Total distance: **10**

Together, **cycle-following** + **cycle-switching** yields an optimal algorithm for LOR. The figures below show the expected distance of restoring items to goals (left) and comparison of end-effector travel distances between optimal (cycle-following + cycle-switching) and greedy (cycle-following only) algorithms. The $x$-axis indicates the number of items.



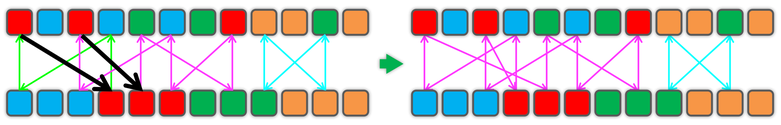

## Optimal Algorithm for POR

For POR, cycle-following does not directly work because items are no longer labeled. We can however perform matchings of items and goals, which induces a set of cycles, over which cycle-following can be performed.

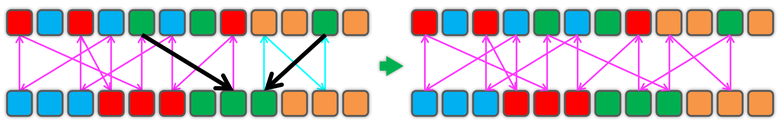


Matching (colored arrows) Induced (three) cycles

We can reduce the total number of pick-n-swaps by performing **cycle-merging**. There are two types: one that does not change end effector travel distance



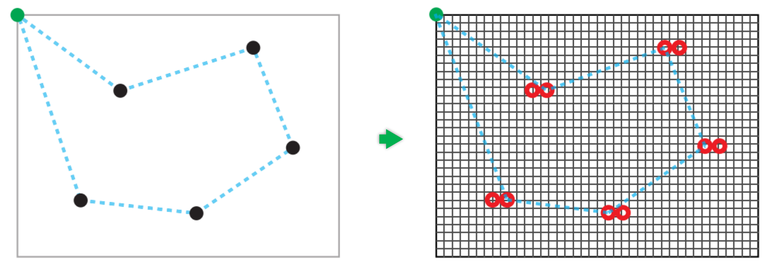
And one that does



Combing the three pieces, cycle-following + cycle-merging + cycle-switching (via computing a minimum spanning tree), yields an optimal algorithm for POR.

## Hardness of LTR and PTR

As we move from one dimension to two dimensions, the problem becomes hard, because computing the shortest end-effector travel embeds a TSP problem.



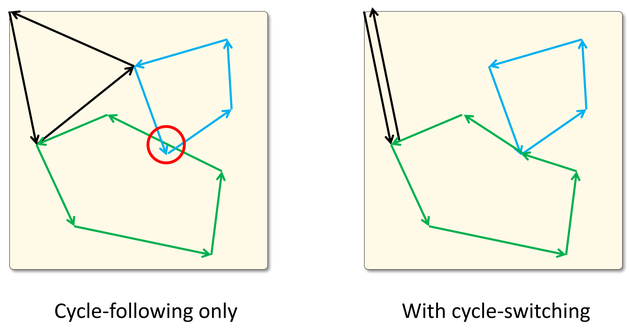
**Theorem.** Optimizing end-effector travel for LTR and PTR, and their higher dimensional versions, is NP-hard.

## Near-Optimal Algorithms of LTR and PTR

But, because there are about $\log n$ cycles for $n$ items, a greedy cycle-following algorithm does well for large $n$.

**Proposition.** The cycle-following algorithm for LTR is asymptotically optimal.

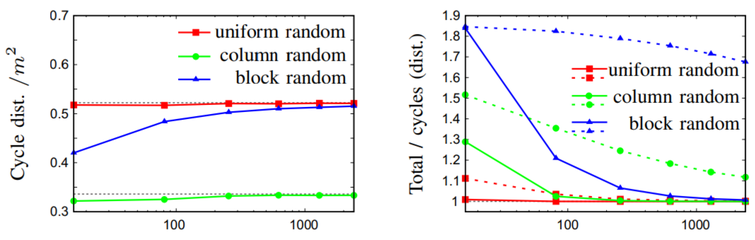
We can also extend the cycle-switching and cycle-merging methods to LTR and PTR, which significantly improve the optimality when the number of items is relatively small. As an example, in the figure below, switching from the green to the cyan cycle can save some end-effector travel.



Cycle-following only With cycle-switching

The algorithm for PTR is essentially a combination of algorithms for POR and LTR.

Similar to LOR, we can compute the expected end-effector travel due to cycle-following for PTR. The left figure below validates our computation. The figure on the right shows the effect of applying cycle-switching. The solid lines are total distance over distance induced by cycle-following with cycle-switching. The dotted lines are the ratio with only cycle-following.




## Conclusions and Source Code

Contributions

- ❑ Proposed lattice rearrangement w pick-n-swap primitive
- ❑ Optimal polynomial-time algorithms for LOR and POR
- ❑ Computational intractability for LTR/PTR
- ❑ Asymptotically optimal algorithms for LTR/PTR
- ❑ Fast $1.x$-optimal algorithms for LTR/PTR, small $n$
- ❑ Extends to arbitrary dimensions

Source code of algorithm implementations

https://github.com/arc-l/lattice-rearrangement

This work is supported in part by NSF awards IIS1734492, IIS-1845888, and CCF-1934924. We sincerely thank the anonymous reviewers for many insightful suggestions and questions.

RUTGERS THE STATE UNIVERSITY OF NEW JERSEY