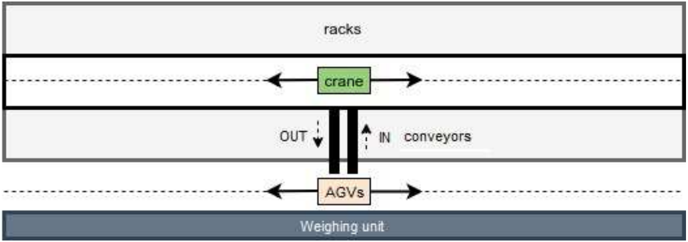
Paths and matchings in an automated warehouse

Michele Barbato, Alberto Ceselli, Giovanni Righini

ODS 2019, Genova

Università degli Studi
di Milano

The automated warehouse



Each side of the warehouse is a matrix of locations of size $V \times H$.

On one side, two adjacent locations are taken by the endpoint of the conveyor (“origin”).

Each location hosts two sites: front and rear.

Hence, there are $2(2 \times V \times H - 2)$ sites.

The automated warehouse

Each site contains a box.
Each box can contain several components.

The crane can access any location by moving vertically and horizontally.
We assume the knowledge of a complete distance (or time or energy consumption) matrix.

The crane capacity is 2.

To access a site on the rear layer, the crane must pickup the box on the front layer (if any) first and reinsert it afterwards.

The automated warehouse

Order list:

- pick-up orders: a box must be carried from its site to the conveyor;
- delivery orders: a box must be taken from the conveyor and reinserted into its site.

Pick-up operations can be executed in any order.
Delivery operations must be executed in a given order (the arrival order on the conveyor).

The same component can be located in many boxes (pick-up site selection sub-problem).
The assignment between boxes and sites can be given or variable (delivery site selection sub-problem).

Instance size

Instances given by the company under study:

- size of the warehouse: $H = 53$, $V = 8$, yielding 1692 sites;
- 4 weighing units simultaneously active;
- about 13 orders per day with an average of 6 components per order, yielding 78 pick-up and delivery operations on the order list;
- the speed of the crane corresponds to about 2 horizontal sites per second and 0.4 vertical sites per second.

Classification

Problem variations:

1. Capacity q of the crane: [1], [2], [$q > 2$].
2. Dimensions l:
   - 1 : a line or two lines stemming from the origin,
   - 2 : a $V \times H$ matrix of locations,
   - 3 : a matrix with double layer (front and rear).
3. Order type o:
   - P : pickup only,
   - D : delivery only,
   - PD : mixed pickups and deliveries.
4. Pickup sites s: fixed [F] or variable [V] pickup sites.

Four fields notation $q/l/o/s$:

- the letters indicate "any"
- the values in square brackets indicate specific cases.

Objective: minimize the overall traveling distance (or time).

Capacity 1: $(1/I/o/s)$

The only non-trivial case is with mixed pickups and deliveries $(1/I/PD/s)$.

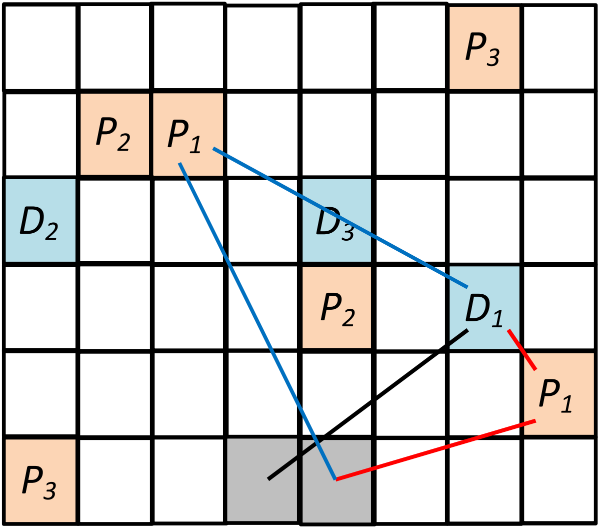
Let P_i be the set of sites corresponding to pickup i.
Let d_{jk} be the distance between the site of j and that of k.

The problem can be transformed into a minimum cost **bipartite matching** problem:

- balance the graph with dummy deliveries or dummy pickups at the origin, if needed;
- define the cost of matching a pickup i with a delivery j as $d_{0j} + \min_{k \in P_i}\{d_{jk} + d_{k0}\}$;
- compute a minimum cost bipartite matching between pickups and deliveries.

Complexity: $O(n^3)$, being n the number of orders.

Capacity 1: $(1/l/o/s)$



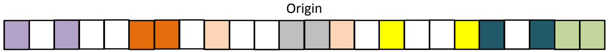
Capacity 2, basic variation ($2/1/P/F$)

The problem is easily solved by sorting.

- Sort the sites to be visited by non-increasing distance from the origin (separately for each line, in the case of two lines).
- Pair the orders, so that each trip of the crane visits the two farthest sites (on the same side of the origin) not yet visited.

In the case with two lines:

- if the number of sites is odd on one of the two lines, the last trip visits a single site;
- if the number of orders is odd on both lines, the last trip visits the two sites closest to the origin, one on each line.



Complexity: $O(n \log n)$.

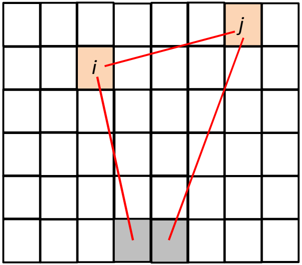
One complicating feature: $(2/2/P/F)$

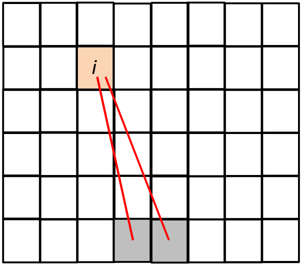
For each (unordered) pair of sites $[i,j]$ in N we define a matching cost

$$c_{[i,j]} = \min\{d_{0i} + d_{ij} + d_{j0}, d_{0j} + d_{ji} + d_{i0}\}.$$

For each site $i \in N$ we define a non-matching cost

$$c_{ii} = d_{0i} + d_{i0}.$$





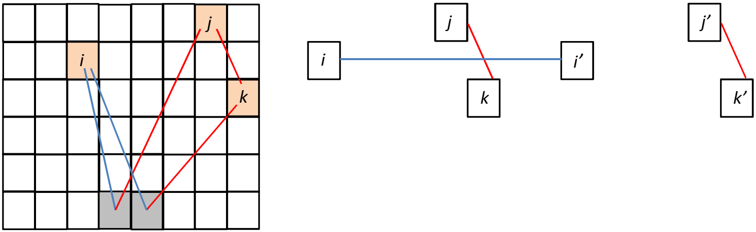
A similar construction holds also for the case $(2/3/P/F)$, with some additional technicalities in the definition of distances if rear layer sites are involved.

One complicating feature: $(2/2/P/F)$

Consider a graph $G = (N \cup N', E \cup E' \cup E'')$, where

- N is the set of pickup sites and N' is a copy of the same set;
- E includes edges $[i, j]$ $\forall i, j \in N$ with weight $c_{[i,j]}/2$;
- E' includes edges $[i', j']$ $\forall i', j' \in N'$ with weight $c_{[i,j]}/2$;
- E'' includes edges $[i, i']$ $\forall i \in N, i' \in N'$ with weight c_{ii}.

A perfect matching in G is made by edges of E'', corresponding to 1-site trips, and edges of E and E' corresponding to 2-sites trips. There exists a perfect matching of minimum cost where edges in E and edges in E' form twice the same matching.



Complexity: $O(n^3)$.

A challenging variation: $(2/1/PD/F)$.

Also the other variations with only one complicating feature are easily solvable, with just one exception.

Problem $(2/1/PD/F)$ cannot be transformed into a matching problem, because the cost of a trip visiting two pickup and delivery pairs is not given by the sum of two terms each one depending on a single pair.

We could devise no straightforward reformulation of this problem into a polynomially solvable graph optimization problem.

Complexity: open.

Two complicating features: $(q/2/P/F)$ and $(q/3/P/F)$.

The problem with generic capacity $q > 2$ and 2 or 3 dimensions is a Capacitated Vehicle Routing Problem with unit demands, i.e. the capacity limits the number of vertices that can be visited in each route.

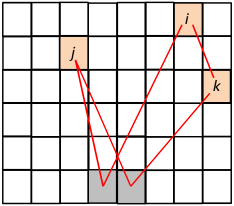
Complexity. The problem is known to be *NP*-hard on general graphs (see Toth and Vigo, 2002).

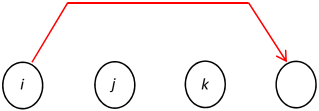
Two complicating features: $(2/2/D/F)$ and $(2/3/D/F)$

If order i is matched with order $j > i + 1$ in a same trip, then all orders between i and j in the input sequence must remain unmatched.
The problem can be reformulated as a shortest path problem on an acyclic digraph, with

- a node for each order in the input sequences;
- an arc $(i, j + 1)$ $\forall i < j$ with weight

$$c_{i,j+1} = \begin{cases} d_{0,i} + d_{i,0} & \text{if } j = i \\ \min\{d_{0,i} + d_{i,j} + d_{j,0}, d_{0,j} + d_{j,i} + d_{i,0}\} & \text{if } j = i + 1 \\ \min\{d_{0,i} + d_{i,j} + d_{j,0}, d_{0,j} + d_{j,i} + d_{i,0}\}+ & \\ + \sum_{k=i+1}^{j-1}(d_{0,k} + d_{k,0}) & \text{if } j > i + 1 \end{cases}$$





Complexity: $O(n^2)$.

Two complicating features: $(2/2/P/V)$ and $(2/3/P/V)$

Let P_i be the subset of sites from which the pickup order i can be satisfied.

The problem can be reformulated as a minimum cost perfect matching problem on a suitable complete graph, where

- there are n nodes, one for each order; if n is odd, add a dummy pickup at the origin.
- the weight of each edge $[i, j]$ is the cost of the most convenient trip among those that visit a site in P_i and a site in P_j.

Complexity. If each pickup order can be satisfied in p sites, the weight of each edge is the minimum among p^2 trip costs.
Defining the weighted complete graph: $O(n^2p^2)$.
Computing a minimum cost perfect matching: $O(n^3)$.
Worst-case time complexity: $O(n^2p^2 + n^3)$.

Another challenging variation: $(q/1/P/V)$.

On a single line, the problem is trivial: it is always optimal to select the site closest to the origin for each pickup order, which makes the problem equivalent to the variation with fixed sites $(q/1/P/F)$.

On two lines, it is optimal to keep only the site closest to the origin on each line for each pickup order. This reduces p to 2.

Complexity.
For $q = 2$, the complexity $O(n^2p^2 + n^3)$ reduces to (at most) $O(n^3)$ (likely to be improvable).
For $q > 2$, open.

Two complicating features: ($q/1/D/F$)

This variation can be efficiently solved with dynamic programming.

- Consider an order at a time, according to the given sequence.
- For each order u one has to decide whether to keep it on the crane or to deliver it.
- Case 1: the crane remains at the origin, because moving the crane without delivering the last loaded order u is always dominated.
- Case 2: the crane goes at least up to the site of u and along its way it serves all delivery orders previously accumulated that are closer than u. The crane can also go further to possibly serve more delivery orders previously accumulated. Each of these possible decisions generates a new dynamic programming state.

Two complicating features: $(q/1/D/F)$

Complexity. The number of iterations is n.

After each iteration u, the number of possible states is bounded by $u + u^2 + u^3 + \ldots + u^{q-1}$, which grows as $O(n^{q-1})$.

Therefore the number of states grows as $O(n^q)$.

From each state at most $q + 1$ possible extensions must be considered.

This yields a time complexity $O(qn^q)$, that is polynomial for each q fixed.

Three complicating features: $(2/2/D/V)$ and $(2/3/D/V)$.

By enumeration it is possible to select the most convenient sites for each order served alone and the most convenient pairs of sites for each pair of orders served in the same trip.

After that, one can still use the transformation of variation $(2/1/D/F)$ leading to a shortest path problem.

The pre-processing step takes polynomial time as in the previous cases (for instance $(2/2/P/V)$ and $(2/3/P/V)$).

Complexity: $O(n^2p^2 + n^3)$.

Conclusions

For several variations, we could find polynomial-time transformations that allow the problem to be efficiently solved with existing polynomial-time graph optimization algorithms such as

- shortest paths,
- min cost bipartite matchings,
- min cost perfect matchings,
- dynamic programming.

Establishing the complexity of two variations, namely $(2/1/PD/F)$ and $(q/1/P/V)$, remains open.

Other variations are already *NP*-hard even in this strongly simplified version of the original problem.

Conclusions

The knowledge about what features are complicating and how the others can be efficiently dealt with paves the way for solving the complete warehouse optimization problem with exact optimization or approximation algorithms based on suitable decompositions or relaxations.

The model can be further enriched by considering additional features such as, for instance,

- deadlines
- energy consumption minimization
- delivery sites selection
- overlaps between orders
- uncertainty
- real-time optimization.

AD-COM
Advanced
Cosmetic
Manufacturing

OptLab