1 Optimal order picking in automated warehouses

An automated warehouses contains a matrix of locations where identical boxes are stored. A given position in the matrix is the I/O interface. A crane can visit the locations and the I/O position to execute pick-up and delivery operations.

Consider the following taxonomy:

1. Capacity of the crane: 1, 2, $k > 2$.
2. Dimensions of the warehouse: 1 (locations along oone or two lines with an endpoint in the I/O position), 2 (locations organized in a matrix), 3 (every location has double depth and can host two boxes: one in the front position is directly accessible from the rail; the other one, in the rear position, can be accessed only after picking-up the box in the front position);
3. Operations: pick-up only (P): they can be executed in any order; delivery only (D): they must be executed according to the input sequence; both pick-up and delivery (PD);
4. Fixed (F) or variable (V) sites; in the latter case not a single site but a subset of sites is associated with each order; one of themm must be visited to satisfy the order.
5. Deadlines: without deadlines (N) or with deadlines (Y); deadlines are expressed as max number of trips before satisfying the order or as maximum allowed completion time.
6. Objectives: minimize the total time (t), minimize the total energy consumption (e).

Some simple variants have been proven to be polynomially solvable. All complex (NP-hard) variants are still to be solved.

Assuming:

1. Capacity: 2, $q > 2$.
2. Dimensions: 1, 2, 3. A further specification $\times k$ means that there are k rails departing from the same origin.
3. Operazions: pickup (P), delivery (D), mixed (PD).
4. Sites: fixed (F) or variable (V).
5. No deadlines.
6. Objective: minimize the total distance traveled.

we can classify models with a four fields notation.

Tha basic variation $2/1/P/F$ is polynomial.

One complicating feature.

$2/1/P/V$ is polynomial because it is equivalent to $2/1/P/F$.

These variants with one complicating feature are also polynomial (see [1]): $q/1/P/F$, $2/2/P/F$, $2/3/P/F$ and $2/1/D/F$.

$2/1/PD/F$ is NP-hard and it has been solved by branch-and-bound and dynamic programming in [3]. $2/1 \times k/PD/F$ can be solved by adapting the algorithms developed for $2/1/PD/F$, since requests can be matched only if they are located on the same rail.

Two complicating features.

Among the variants with two complicating features, these are polynomial, because they can be reformulated as polynomial problems on graphs, such as maximum weight matching and shortest path (see [1]): $2/2/D/F$, $2/3/D/F$, $2/2 \times k/P/V$, $2/3 \times k/P/V$, $2/1 \times k/D/V$.

$2/1/PD/V$ is the same as $2/1/PD/F$ and $2/1 \times k/PD/V$ is the same as $2/1 \times k/PD/F$.

$2/2/PD/F$ and $2/3/PD/F$ as well as $2/2 \times k/PD/F$, $2/3 \times k/PD/F$ are NP-hard, because they generalize $2/1/PD/F$ and $2/1 \times k/PD/F$. No algorithm is known for them. Idea: generalize the branch-and-bound algorithm devised for $2/1/PD/F$, where

- branching is done in the same way: the number of branches to consider is increased by being in 2 or 2 dimensions and it is possibly decreased by the partition of the requests into k distinct rails;
- lower bound 1 can be generalized, using the polynomial algorithms for $2/2/D/F$ and $2/3/D/F$ (instead of $2/1/D/F$);
- lower bound 2 can be obtained from disrergarding the distinction between pickups and deliveries and computing a lower bound to a $4/2/P/F$ or $4/3/P/F$ instance, which is a VRP with unit demand and fixed capacity.

$q/1/P/V$ is polynomial because it is equivalent to $q/1/P/F$.

$q/1 \times 2/P/V$ is polynomial and it has been solved with dynamic programming [2]. $q/1 \times k/P/V$ with $k > 2$ is still open: this is one of the most interesting versions to investigate. Idea: try to generalize the polynomial-time dynamic programming algorithm devised for $q/1/P/V$ with similar dominance rules.

$q/1/D/F$ is polynomial and it has been solved by dynamic programming (see [4]).

$q/1/PD/F$ and $q/1 \times k/PD/F$ are NP-hard, because they are generalizations of $2/1/PD/F$: no algorithm is known for these variants. Idea 1: design a branch-and-bound algorithm similar to that for $2/1/PD/F$, by a suitable generalization of the branching strategy. Idea 2: design a dynamic programming algorithm similar to that for $q/1/D/F$.

$q/2/P/F$ e $q/3/P/F$ are NP-hard, because they are special cases of the Capacitated VRP with unit demand (that is NP-hard). In particular $q/3/P/F$ has the additional constraint that some pickups (those in the rear layer) must be visited among the first $q - 1$ in their trip, i.e. they cannot be the last visit in a complete trip.

Three complicating features.

Among the variants with three complicating features, $q/1/D/V$ is polynomial because it is equivalent to $q/1/D/F$. However $q/1 \times k/D/V$ is open, while $q/1/PD/V$ and $q/1 \times k/PF/V$ are NP-hard, but no algorithm has been developed for them so far.

$2/2/D/V$, $2/3/D/V$, $2/2 \times k/D/V$ and $2/3 \times k/D/V$ are polynomial, because they can be transformed in a shortest path problem on a suitably weighted acyclic digraph [1].

$2/2/PD/V$, $2/3/PD/V$, $2/2 \times k/PD/V$ and $2/3 \times k/PD/V$ are interesting for two reasons: (a) they correspond to the real application that triggered this research stream on AS/RS optimization; (b) they are NP-hard, being a generalization of $2/1/PD/F$, but they have only one more complicating feature compared to $2/2/D/V$, $2/3/D/V$, $2/2 \times k/D/V$ and $2/3 \times k/D/V$, which are polynomial.

All the other versions are NP-hard (variants of the vehicle routing problem) and no algorithm has been developed so far.

The same holds for the problem with four complicating features.
Deadlines.

The only problem with deadlines solved so far is $2/1/P/F$, but in the special case in which deadlines are expressed in number of trips [5]. No result has been obtained so far with deadlines expressed in units of time.

References

[1] Barbato, Ceselli, Righini, *Paths and matchings in an automated warehouse*, AIRO Springer Series 3 (2019) 151-159.

[2] Barbato, Ceselli, Righini, *A polynomial-time dynamic programming algorithm for an optimal picking problem in automated warehouses*, Journal of Scheduling 27 (2024) 393-407.

[3] Bianchessi, Ostuni, Righini, *Exact optimization algorithms for an order picking problem*, Open Journal on Mathematical Optimization, under review.

[4] Cracco, Ostuni, Righini, Rizzi, *A polynomial-time dynamic programming algorithm for an order picking problem*, in preparation.

[5] Righini, *A dynamic programming algorithm for an order picking problem with deadlines*, AIRO Springer Series, to appear.