Paths and matchings in an automated warehouse

Michele Barbato, Alberto Ceselli and Giovanni Righini

Abstract We analyze a number of variations of a combinatorial optimization problem arising from the optimization of an automated warehouse. We classify these variations according to four relevant parameters and we analyze which combinations are polynomially solvable, owing to dynamic programming recursions or to reductions to known graph optimization problems such as the shortest path problem and the minimum cost perfect matching problem.

1 The automated warehouse

This study arises from an applied research project on smart manufacturing, involving some companies in the cosmetic industry. In particular, we consider the optimization of automated warehouses where components such as colors, pigments and bulk products are stored in identical boxes. When a production order is processed its bill of materials is sent to the weighing unit, close to the warehouse. The boxes containing the needed components are then brought to the weighing unit through a conveyor and an AGV system; with the same means the boxes are returned to the warehouse afterwards.

The structure of the automated warehouse under study is illustrated in Figure 1. The boxes are stored on the two sides of a rail, carrying a crane. In each of the H available positions along the rail, V vertically aligned locations are available and the crane can reach them by moving up and down. In addition, each location can contain two boxes: one directly facing the corridor with the crane (front layer) and

Michele Barbato
University of Milan, Department of Computer Science, e-mail: michele.barbato@unimi.it

Alberto Ceselli
University of Milan, Department of Computer Science, e-mail: alberto.ceselli@unimi.it

Giovanni Righini
University of Milan, Department of Computer Science, e-mail: giovanni.righini@unimi.it

the second one just behind it (rear layer). The crane has capacity two, i.e. it can carry up to two boxes simultaneously. To access a box in the rear layer the crane first extracts the box in the front layer, then it extracts the box in the rear layer and finally it reinserts the box in the front layer. Hence, to perform this sequence of operations the crane must be initially empty. Two adjacent locations of the automated warehouse are taken by the input and the output conveyors, running in opposite directions. In the remainder the locations of the conveyors are identified as the *origin*, since they are the position from which the crane starts and to which the crane returns. A *trip* is the sequence of movements of the crane between two consecutive visits to the origin. In the remainder we call *site* the position where a box can be stored in the warehouse. So, the warehouse is assumed to have $V \times H$ locations on one side, $V \times H - 2$ locations on the other side (where the conveyors are) and 2 sites (front and rear) for each location.

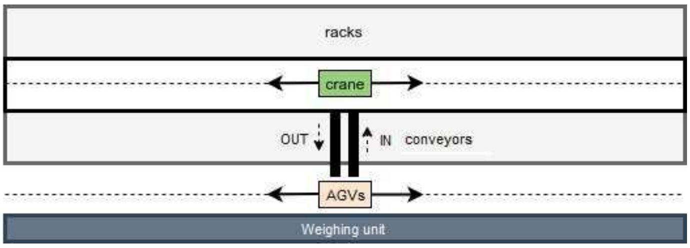

Fig. 1 The automated warehouse

The crane must execute a set of pickup operations to send the required boxes to the weighing unit and a set of delivery operations needed to put the used boxes back into their locations. The same component may be stored in several boxes: therefore the same pickup order may be satisfied by visiting any site in a given subset. On the contrary, delivery operations are assumed to be constrained to store the boxes into the same sites from which they had been extracted: therefore each delivery operation is satisfied by visiting a given site. Another asymmetry between pickup and delivery operations is their sequence. Pickup operations can be satisfied in any order, because there is no precedence between weighing different components in a bill of materials. On the contrary, when used boxes are returned to the warehouse, the order in which they arrive depends on the weighing unit and it cannot be chosen in the warehouse: the crane must process the incoming boxes in the order they arrive on the input conveyor. Therefore the input of the optimization algorithm may include a set of pickup orders (each corresponding to a subset of sites), a sequence of deliveries (each corresponding to a single site) or both.

The optimization of the crane movements is a hard combinatorial problem that, in addition, may need to be solved both off-line and on-line (i.e. re-optimized in real time). The study originates from the collaboration with a manufacturing company whose warehouse has $H = 53$, $V = 8$ and 4 weighing units. In average, a typical instance to be solved may have 13 orders per day with 6 ingredients per order, while the speed of the crane corresponds to about 2 horizontal sites per second and 0.4 vertical sites per second.

In [2] we evaluate the applicability of state-of-the-art on-line algorithms in a similar context. In this short paper we concentrate on the off-line version, in which we have considered the simplest variations of the problem for three purposes: (a) to possibly establish a starting point for a decomposition algorithm allowing for the exact off-line optimization of the most difficult variations of the problem; (b) to pave the way for the design of on-line optimization strategies; (c) to have a term of comparison against which an on-line strategy can be compared.

In particular, we classified the problem variations according to the following features:

1. Capacity q of the crane: [1], [2], [$q > 2$];
2. Dimensions l: [1] (a line or two lines originating from the origin, i.e. the I/O position), [2] (a $H \times V$ matrix of locations), [3] (double layer);
3. Operations o: pickup only [P], delivery only [D], mixed pickups and deliveries [PD];
4. Sites s: fixed [F] or variable [V] sites;

In the remainder we use a four fields notation $q/l/o/s$, where the letters indicate "any case" while the values listed above in square brackets indicate specific cases.

Here we assume that the objective to be minimized is the overall traveling time taken by the crane to perform all its duties. Other objectives may be also relevant in an industrial context, such as the minimization of energy consumption.

2 Problem variations

2.0.1 Problem variations with capacity 1 $(1/l/o/s)$.

The case with capacity 1 (which implies no double layer) is trivial in almost all variations, since the crane can visit only one of the sites for each trip. The only non-trivial case is with mixed pickups and deliveries $(1/l/PD/s)$. In this case the problem can be transformed into a minimum cost bipartite matching problem. First, dummy deliveries or dummy pickups at the origin site are generated, in order to have the same number of both operations. Then a minimum cost matching between pickups and deliveries is computed, where the cost of matching a pickup i with a delivery j is $\min_{k \in P_i}\{d_{jk}\}$, where P_i is the set of sites corresponding to pickup i and d_{jk} is the distance (or travel time) between the site of j and that of k.

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

In the remainder we only consider variations with crane capacity equal to 2 or larger.

2.0.2 Basic problem variation: $(2/1/P/F)$.

This variation corresponds to the easiest combination of the four fields. The problem can be solved by initially sorting the sites to be visited by non-increasing distance from the origin (separately for each line, in the case of two lines). Then they are paired 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 of two lines if the number of sites is odd on one of the two lines, the last trip visits a single trip, that is the closest to the origin on the line with an odd number of sites. 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: $O(n \log n)$.

2.1 One complicating feature

Here we examine the problem variations in which only one of the four fields takes a different value with respect to the basic variation.

2.1.1 Capacity $q > 2$ $(q/1/P/F)$.

Sites are sorted as in the basic variation. Each trip visits the q farthest sites (on the same line) not yet visited. If there are two lines with a number of sites that is not a multiple of q, let $r_1 < q$ and $r_2 < q$ the remaining sites to be visited, Then, if $r_1 + r_2 > q$ two final trips, one on each line, visit the remaining sites; else, $r_1 + r_2 \leq q$ a single final trip is enough to visit all the remaining sites.

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

2.1.2 Two dimensions $(2/2/P/F)$.

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

$$c_{[i,j]} = \min\{d_{0i} + d_{ij} + d_{j0}, d_{0j} + d_{ji} + d_{i0}\} \tag{1}$$

and for each site i we define a non-matching cost

$$c_{ii} = d_{0i} + d_{i0}, \tag{2}$$

where d are the (possibly asymmetric) distances or traveling times to be optimized. If the traveling times $d(w)$ also depend on the transported weight, w, then we have

$$c_{[i,j]} = \min\{d_{0i}(w_i + w_j) + d_{ij}(w_j) + d_{j0}(0), d_{0j}(w_i + w_j) + d_{ji}(w_i) + d_{i0}(0)\} \tag{3}$$

and

$$c_{ii} = d_{0i}(w_i) + d_{i0}(0), \tag{4}$$

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

- N is the set of orders (pickups);
- N' is a copy of the same set;
- E includes edges $[i, j]$ with weight equal to half of $c_{[i,j]}$ defined as above for each pairs of elements of N;
- E' includes edges $[i', j']$ with weight equal to half of $c_{[i,j]}$ defined as above for all pairs of elements of N';
- E'' includes edges (i, i') with weight c_{ii} defined as above, linking the two copies of each vertex in N and in N'.

A perfect matching in G is made by some edges of E'', corresponding to trips visiting a single site, and by edges $[i, j]$ and $[i', j']$ corresponding to trips visiting two sites. In particular, it is trivial to prove that there exists a perfect matching of minimum cost where edges in E and edges in E' form twice the same matching.

Complexity: $O(n^3)$.

2.1.3 Three dimensions $(2/3/P/F)$.

The same definitions given above still apply, but now the traveling times d also take into account the time needed to access the rear layer. Furthermore, if two sites i and j are both in the rear layer, the pair $[i, j]$ is infeasible and $c_{[i,j]}$ must be set to ∞ in (1) or (3)).

For each pair of sites it is necessary to select the optimal sequence of visits, with the constraint that sites in the rear layer must be visited first. Then $d_{ij}(w)$ is set to ∞ for all i and w in (3) if j is in the rear layer.

A polynomially solvable matching problem is generated as in the previous case.

Complexity: $O(n^3)$.

2.1.4 Deliveries $(2/1/D/F)$.

Deliveries imply an additional constraint with respect to pickups: if box i is matched with box $j > i+1$ in a same trip, then all boxes between i and j in the input sequence must remain unmatched: the crane takes box i from the input conveyor without delivering it, then it delivers all boxes between i and j one by one and finally it takes box j and delivers both i and j in a single trip.

This problem can be reformulated as a shortest path problem on an acyclic digraph whose nodes are numbered according to the input sequence and the weight of each arc (i,j) with $j > i$ is defined as follows:

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

where $d(w)$ are the traveling costs on the line, possibly dependent on the transported weight w.

Complexity: $O(n^2)$.

2.1.5 Mixed pickups and deliveries $(2/1/PD/F)$.

In this case the problem 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. Therefore this remains an open question.

2.1.6 Variable sites $(2/1/P/V)$.

In this case we indicate with n the set of requested pickup operations and by P_i 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 graph. If n is odd, add a dummy pickup at the origin. The graph has a vertex for each pickup operation, it is complete and the weight of each edge $[i,j]$ is the cost (traveling time) 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. Hence, to define the weighted graph costs $O(n^2p^2)$ time. The computation of the minimum cost perfect matching costs $O(n^3)$ time. Therefore the worst-case time complexity is $O(n^2p^2+n^3)$.

Remark. The same construction holds also in two and three dimensions, i.e. for variations $(2/2/P/V)$ and $(2/3/P/V)$. In one dimension it is optimal to keep only the site closest to the origin on each line for each pickup order. This reduces p to 2 and the time complexity to $O(n^3)$.

2.2 Two complicating features

Now we consider the six variations arising from considering two complicating features at a time.

2.2.1 Capacity and dimensions $(q/2/P/F)$ and $(q/3/P/F)$.

The problem is now a Capacitated Vehicle Routing Problem with unit demands, i.e. the capacity limits the number of vertices that can be visited in each route. The problem is known to be NP-hard [1].

2.2.2 Capacity and deliveries $(q/1/D/F)$.

This variation can be efficiently solved with dynamic programming. Consider a delivery order at a time, according to the given input sequence. For each order u one has to decide whether to keep it on the crane or to deliver it. In the former case the crane remains at the origin; Moving the crane without delivering the last loaded order u is always dominated by another policy, i.e. moving the crane to do perform the same operations before loading u; but this is already considered in the dynamic programming states generated before considering u. In the second case the crane goes at least up to the site of u and along its way it serves all delivery orders kept in it up to that moment, if they are closer to the origin than u. Moreover 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.

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.

2.2.3 Capacity and variable sites $(q/1/P/V)$.

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 easy to prove that for each line we can keep only the site closest to the origin for each pickup order. Hence each pickup order can be satisfied in either of two sites, on opposite sides with respect to the origin. However, in spite of this simplification we could not find any polynomial time algorithm to solve the resulting model.

Therefore this problem remains open.

2.2.4 Dimensions and deliveries $(2/2/D/F)$ and $(2/3/D/F)$.

The same construction of variation $(2/1/D/F)$ holds. The only change is in how arc costs are computed from traveling times. Hence, the problem can be transformed into a shortest path problem on an acyclic weighted digraph.

Complexity: $O(n^2)$.

2.2.5 Dimensions and variable sites $(2/2/P/V)$ and $(2/3/P/V)$.

In two dimensions we can re-use the same transformation of variation $(2/2/P/F)$, leading to a minimum cost perfect matching problem, after selecting:

- the most convenient site, for each single pickup order;
- the most convenient pair of sites, for each pair of pickup orders.

In three dimensions the same construction holds again, but there is no guarantee that all trips visiting two sites are feasible: any two sites in the rear layer are incompatible, because both require the crane be empty to execute the pickup operation. However the pre-processing step needed to check for incompatibilities has polynomial complexity: with n orders and p sites for each order the computation of the arc costs takes $O(n^2 p^2)$. Solving the resulting minimum cost perfect matching problem requires $O(n^3)$.

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

2.2.6 Deliveries and variable sites $(2/1/D/V)$.

On a single line it is always optimal to select the site closest to the origin for each delivery order. Hence, the problem is equivalent to $(2/1/D/F)$ and can be solved in polynomial time as a shortest path problem.

On two lines it is always optimal to select on each line the site closest to the origin for each delivery order. Hence, the same transformation of variation $(2/1/D/F)$ holds again with the only difference that the quantity $\min\{d_{0i}(w_i + w_j) + d_{ij}(w_j) + d_{j0}(0), d_{0j}(w_i + w_j) + d_{ji}(w_i) + d_{i0}(0)\}$ must be chosen in an optimal way among four possibilities, corresponding to the four combinations of the two sites for each of the two orders i and j.

Complexity: $O(n^2)$.

2.3 *Three complicating features*

There is only one triplet of features that generate polynomial problems when they are taken two at a time.

2.3.1 Dimensions, deliveries and variable sites $(2/2/D/V)$ 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: $O(n^2 p^2 + n^3)$.

3 Conclusions

We have analyzed several variations of a problem of optimizing the traveling time of a crane in an automated warehouse. For many variations we have shown polynomial-time transformations that allow the problem to be efficiently solved with existing algorithms for computing shortest paths or perfect matchings or with 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 simplified version of the original problem.

The knowledge about what features are complicating and how the others can be efficiently dealt with paves the way for solving the original problem with exact optimization algorithms based on suitable decompositions or relaxations. The model can be further enriched by considering additional features such as deadlines and energy consumption minimization as well as uncertainty and real-time decision policies. This is the subject of an ongoing research program.

Acknowledgments

This research was partially funded by Regione Lombardia, grant agreement n. E97F17000000009, project AD-COM.

References

1. P. Toth, D. Vigo (eds.), The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
2. M. Barbato, A. Ceselli, F. Mosconi, A Computational Evaluation of Online ATSP Algorithms, Technical Report of University of Milan, submitted to ODS 2019.