A pick-up and delivery problem with drones.

The problem was presented at ECCO 2023 by Emmanouil Giannoulakis, who proposed heuristics.

A set N of origin/destination pairs are given in known positions. Each pair (o_j, d_j) corresponds to an order, consisting of transporting a package of given weight w_j from the origin o_j to the destination d_j. The distance between locations (origins and destinations) is given.

A drone has a given payload capacity Q and a given weight U. Its battery has a given capacity B. The drone starts from a base station with full charge and it must return to the base station before the battery charge is exhausted.

The energy consumption of a drone when traveling from location i to location j is given by $e_{ij} = d_{ij} * (U + f_{ij})$, where d_{ij} is the distance between i and j and f_{ij} indicates the payload transported from i to j.

Possible objectives:

- maximize the total weight transported;
- maximize the total number of orders satisfied.

An exact optimization algorithm.

The proposed algorithm is based on dynamic programming.

Dynamic programming state:

- u: the last location reached;
- r: the amount of residual energy available;
- $S \subseteq N$: the subset of satisfied orders (both the origin and the destination have been visited);
- $O \subseteq N$: the subset of open orders (only the origin has been visited).

Extension: feasibility test. An extension from location i to location j is feasible only if the following conditions hold:

1. i is the last reached location;
2. $d_{ij} * (U + f_{ij}) \leq r$ (there is enough energy to travel from i to j);
3. j has not been already visited;
4. - either j is a pickup vertex and $w_j + f_{ij} \leq Q$ (there is enough capacity to pickup the package at j),

- or j is the delivery vertex of one of the open orders in O.

Some of these conditions can be strengthened, to early detect infeasible states.

Extension: state update. The extension rules are the following.

If j is the pickup vertex of order $k \in N$, then

- $u \leftarrow j$
- $S \leftarrow S$
- $O \leftarrow O \cup \{k\}$
- $r \leftarrow r - d_{ij}(U + \sum_{i \in O} w_i)$

If j is the delivery vertex of order $k \in N$, then

- $u \leftarrow j$
- $r \leftarrow r - d_{ij}(U + \sum_{i \in O} w_i)$
- $S \leftarrow S \cup \{k\}$
- $O \leftarrow O \backslash \{k\}$

Dominance. A state (u', r', S', O') dominates a state (u'', r'', S'', O'') when all these conditions hold

- $u' = u''$
- $r' \geq r''$
- $S' = S''$
- $O' = O''$

Since conditions $S' = S''$ and $O' = O''$ are very restrictive, dominance is unlike to allow deleting many states. However, owing to the battery capacity limit, it is also unlikely that a single feasible path visits many locations. Hence the combinatorial explosion could be kept under control.

In case of long paths, bi-directional dynamic programming can be used, because the payload (and therefore the energy consumption along each arc) can be computed exactly also along backward paths.

Possible extensions. Variants of interest include the possibility of battery replacement/recharge, multi-drone optimization, and the use of more realistic non-linear functions to describe the energy consumption (possibly depending also on the state of the battery, weather conditions and so on).