Multi-robot task allocation

The company Guber provides automatic pick-up and delivery services for goods. At any time, customers can send to a control center their requests for receiving selected goods at their homes. For instance, customer c can ask for a bundle of goods $G^c = \{g_1^c, g_2^c, \ldots, g_n^c\}$. Goods are available at known locations, where the robot can pick them up. Each good i has a capacity requirement q_i and a value v_i. The delivery of the goods in a bundle G^c requires the pick-up of the associated goods. While strictly related, these should be seen as different types of tasks.

Periodically, customer requests are organized into a batch and dispatched from the control center to a team of n mobile robots that perform the pick-up and delivery tasks. Each robot k has a limited capacity Q_k. The goods bundle G^c requested by a customer c is assigned to a single robot (no split of requests), respecting robot's capacity constraint.

Based on the knowledge of the capacity requirements of the goods, of robots' capacity limitations and mobility skills (e.g., wheeled vs. legged), and of the locations of goods and customers, Guber's control center computes the set R_k of all routes that can be feasibly assigned to a robot k for picking-up and deliver the goods for each one of the customers. For each robot k, each feasible route r has known traveling cost c_r^k.

Given the number of requests in the batch and robots' capacity constraints, it might not be possible to satisfy all customer requests. Therefore, Guber wants to assign routes (i.e., pick-up and delivery subtask bundles) to the available robots with the goal of maximizing the difference between the overall value obtained from the delivered goods and the costs incurred for traveling, with the additional constraint that for any robot traveling over the assigned routes doesn't exceed a maximum cost (i.e., time) T_{max}.

1. Define an optimization model for Guber's task assignment problem.

   Solution:

   - The set of goods to pick-up plus the set of delivery actions define the set $T = \{1, 2, \ldots, t\}$ of the subtasks in the batch. Each subtask $t \in T$ has a value v_t that corresponds to the actual value of the good for a pick-up subtask, and to a constant (e.g., zero) value for a delivery subtask.
   - $K = \{1, \ldots, n\}$ is the set of the robots in the team.
   - $x_r^k \in \{0, 1\}, \forall\ r \in R_k,\ k \in K$: binary variables indicating whether robot k performs (feasible) route k or not in the assignment.
   - $c_r^k \in \mathbb{R}, \forall\ r \in R_k, k \in K$: real-valued constants specifying the cost of traveling on route r for robot k.
   - $\tau_{tr}^k \in \{0, 1\}, \forall\ t \in T,\ r \in R_k,\ k \in K$: parameter saying whether task t is served on route r that belongs to k's feasible routes.

   The resulting IP model:

$$
\begin{aligned}
\max \; & \sum_{t\in T}\sum_{k\in K}\sum_{r\in R_k} v_t \tau^k_{tr} x^k_r - \sum_{k\in K}\sum_{r\in R_k} c^k_r x^k_r \\
s.t. \; & \sum_{r\in R_k} c^k_r x^k_r \le T_{max} && \forall k \in K \\
& \sum_{k\in K}\sum_{r\in R_k} \tau^k_{tr} x^k_r \le 1 && \forall t \in T \\
& \sum_{r\in R_k} x^k_r \ge 0 && \forall k \in K \\
& x^k_r \in \{0,1\} && \forall\; r \in R_k,\; k \in K
\end{aligned}
$$

The first constraints say that each robot must be assigned a number of routes whose cumulative cost doesn't exceed T_{max}. The second constraint says that each subtask must be performed by at most one robot. The third constraint (not strictly necessary) says that a robot can have 0 or more tasks assigned ot it. Any robot that is not assigned to a route is not used, as well as, a task may be assigned to no robot (e.g., when the reward is not sufficient to balance the cost).

2. Can the resulting optimization problem be solved in polynomial time? (Yes/No, why)

   Solution: No, the above model has the form of a set-partitioning problem, which is NP-hard.

3. Based on Gerkey and Mataric taxonomy, how can the above problem can be classified?

   Solution: It is $\{ST, SR, TA\}$. The set partitioning formulation (instead of the generalized assignment) results from having available R_k, the set of feasible routes for a robot.

4. Define, with a pseudo-code, an alternative strategy for addressing Gruber's problem following a distributed approach that doesn't rely on an optimization model.