of the MIP model presented in Kinable et al. [2014] and of two more compact models can be found in the thesis of Hernández López [2020]. In Kinable and Trick [2014], we found an attempt to solve the previous problem with a logic-based Benders' approach. Sulaman et al. [2017] expand upon the SD heuristic proposed in Kinable et al. [2014], proposing a simulated annealing (SA) combined with a time-slot Heuristic (SATH). This method looks for a slot between existing visits of a truck to schedule a new delivery instead of assigning it to the time slot strictly after the truck's latest assigned delivery. The goal is to reduce the large time gaps that can be present in a schedule created with SD due to ignoring the intermediate available time slots. Experimental results indicated that SATH outperforms SD in speed and solution quality. A generalization of the MIP model of Kinable et al. [2014] is addressed in Asbach et al. [2009]. This model simultaneously minimizes the total sum of travel costs and the penalty costs for customers with unfulfilled demand. A customer can request that all concrete deliveries come from the same plant or a subset of plants and that a delivery truck belongs to a subset of the vehicle fleet. The MIP model is used in a local search scheme as a black-box solver to reoptimize an incumbent solution in which a neighborhood operator has unfixed some variables. Tzanetos and Blondin [2023] provide an overview of the various methods used in the literature to address the CDP and categorizes the problem formulations based on the different concepts used in the literature. They also discussed the consistency between industry needs and existing constraints and provided insights into the datasets corresponding to real-world cases, identifying the necessary data for practitioners.

3 Problem description

The focus of this paper is the distribution of RMC from a Canadian company that operates in the greater Montreal area. When a customer places an order, it is received at a central center and assigned to one of the company's batching plants. These plants produce the concrete and then deliver it to the customer. The problem we are examining involves a set of customer orders, a set of concrete-mixer drivers, and a set of batching plants.

A customer i requests one or more types of concrete to be delivered to his construction site on a particular day, starting at time a_i. We call an order o a request for a specific type of concrete. q_i is the sum of the demand q_o of each order o placed, a_i is the desired arrival time of the first concrete mixer, and τ_i^u is the unloading rate for customer i. If the order requires more concrete than a single truck can carry, multiple deliveries are scheduled.

Let O_i be the set of all orders requested by customer i. Each element of O_i must be fully delivered before moving on to another order. Exactly one order o of O_i must have its first delivery start at a_i, while the others can start at any time after o is completed. Once a plant is selected to produce a customer's first delivery, it must be the supplier of all subsequent deliveries from that customer. To avoid cold joint problems with the concrete, the subsequent deliveries must be made in close succession. We define a maximum time delay γ_i after which no more deliveries will be allowed. The customer unloading rate and the quantity to be unloaded give the time required to unload a truckload. Let $\mathcal{C}$ be the set of construction sites (customers) with a planned delivery for the day, and $\mathcal{O} = \{O_i, i \in \mathcal{C}\}$ be the set of all requested orders for all customers.

The company has two types of concrete mixer trucks with capacities of 8 and 12 cubic meters. Each driver k is assigned to a particular batch plant and is responsible for driving a truck with capacity Q_k. The set of drivers is represented by $K = \cup_{b \in \mathcal{B}} K_b$, where K_b is the set of drivers scheduled to start their shift at batch plant b. t_{ij} is the known time to travel between any two locations i and j. A scheduled driver k is required to start his shift at H_k, work a minimum of M_T hours and a maximum of N_T hours during regular working hours, with the possibility of overtime of up to O_T hours. β_3 and β_4 are the penalties incurred if a driver works less than M_t and more than N_t.

A driver typically loads RMC at his assigned batch plant but may be required to drive to and load at other plants if needed. The batch plant produces concrete on demand using recipes specific to each order. This means that a truck can only haul RMC for one order, even if there is spare capacity. To fill multiple orders, a driver must restock at a plant between deliveries. After unloading the RMC, a driver takes β_k minutes to clean the concrete mixer before proceeding.

Let n_o be the number of deliveries needed to fulfill the order o. n_o is not known in advance because we use a fleet of trucks with different capacities. However, we can compute its lower (n_o^{min}) and upper (n_o^{max}) bounds using the capacities of the largest (Q_{max}) and smallest (Q_{min}) available trucks. The lower bound is the number of deliveries required if we only use trucks with capacity Q_{max}, while the upper bound is the number of deliveries required if we only use trucks with capacity Q_{min}.

$$n_o^{min} = \left\lceil \frac{q_o}{Q_{max}} \right\rceil \leq n_o \leq n_o^{max} = \left\lceil \frac{q_o}{Q_{min}} \right\rceil \; \forall o \in \mathcal{O}. \tag{1}$$

Let d_o^j be the j_{th} visit with load q_o^j for order o. We represent the fulfillment of order o by the visits to the ordered set of delivery nodes $\mathcal{D}_o = \left(d_o^0, d_o^1, \cdots, d_o^{n_o}\right)$. The deliveries of customer i are the ordered set $\mathcal{D}_i = (\mathcal{D}_{o_1}, \mathcal{D}_{o_2}, \cdots, \mathcal{D}_{o_{|O_i|}})$, where o_r is the rth delivered order. We will refer to $d \in \mathcal{D}_i$ ($d \in \mathcal{D}_o$) as the dth potential delivery of customer i (order o). $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$ is the union of all delivery nodes.

The company owns a total of eight batching plants located in various geographical regions. Each plant has a loading dock that can only accommodate only one truck at a time, which leads to trucks lining up. Let $\mathcal{B}$ be the set of batching plants. The plants are heterogeneous, as each plant b has its own hourly loading rate, represented by τ_b^l, which affects the duration of the loading process. After loading the concrete, the driver spends α_b minutes adjusting the concrete in the truck before heading to the customer site. Each plant has its own assigned fleet of trucks, but it can borrow trucks from other plants or hire external vehicles if necessary. A batching plant can serve a construction site as long as the travel time between the two is less than the concrete's lifespan, which is represented by Δ. Let $l_{b,j}$ be the loading dock node associated with delivery node j at plant b. We represent each plant b by the set $\mathcal{L}_b = \{l_{b,j}, t_{bj} \leq \Delta, j \in \mathcal{D}\}$ of loading docks nodes. $\mathcal{L} = \bigcup_{b \in B} \mathcal{L}_b$ is the union of all loading docks.

A solution to the problem involves making decisions about truck loading schedules, driver assignments to different deliveries, and truck arrival times at construction sites for unloading. For a batching plant, the decision involves choosing which driver to load, when to load them, and which construction site they should deliver to. For a driver, the decision involves determining the sequence of loading depots and delivery sites. And for a construction site, the decision involves determining the arrival times of all scheduled deliveries for the day.

Each driver leaves and returns to his home plant every day. We represent the home plant of a driver k with a starting depot s_k and an ending depot e_k. S and E are the sets of starting and ending depots, respectively.

We define our problem on a complete directed graph where $V = \{S \cup \mathcal{L} \cup \mathcal{D} \cup E\}$ is the set of nodes. The arc sets are $A = \{(i,j,k) \mid i,j \in V \; k \in K\}$, $A^D = \{(i,j) \mid i,j \in \mathcal{D}\}$, and $A^L = \{(i,j) \mid i,j \in \mathcal{L}\}$. A corresponds to allowed movements of drivers from node i to node j. For each driver k, the allowed movements are the following:

- From the starting depot s_k to a loading dock $l \in \mathcal{L}$ or to the ending depot e_k.

- From a loading dock $l \in \mathcal{L}$ to a delivery node $d \in \mathcal{D}$.
- From a delivery node $d \in \mathcal{D}$ to a loading dock $l \in \mathcal{L}$ or to the ending depot e_k.

For a customer c, arcs in A^D link consecutive delivery nodes of the same order $\{(i,j) \in \mathcal{D}_o, o \in \mathcal{O}_c, i < j\}$, and pair of delivery nodes of two different orders $\{(i, d^0_{o_2}), i \in \mathcal{D}_{o_1}, i \geq n^{min}_{o_1}, o_1, o_2 \in \mathcal{O}_c, o_1 \neq o_2\}$. Arcs in A^L link all pairs of loading docks of the same batching plant.

$\delta^+(i) = \{(i,j,k) \in A\}$, and $\delta^-(i) = \{(j,i,k) \in A\}$ are the outcoming and incoming arc sets of any node $i \in V$. $\delta^+_D(i) = \{(i,j) \in A^D\}$, and $\delta^-_D(i) = \{(j,i) \in A^D\}$ are the outcoming and incoming arc sets of delivery node $i \in \mathcal{D}$. Similarly, $\delta^+_L(i) = \{(i,j) \in A^L\}$, and $\delta^-_L(i) = \{(j,i) \in A^L\}$ are the outcoming and incoming arc sets of loading node $i \in \mathcal{D}$.

Let the binary variable x^k_{ij} be 1 if the driver k travels from node i to j. The binary variable y_o is 1 when the order o is completely served. v_i and w_i are the start and end of the loading (unloading) operation at node $i \in \mathcal{L} \cup \mathcal{D}$. The binary variable $u_{i,j}$ is 1 if node j is served just after i, the service being either an unloading or a loading operation. The binary variable σ_{ib} is equal to 1 if the orders of customer i are loaded by plant b. Variable q^k_j is the quantity to be loaded towards j with vehicle k. Let w^1_k be a continuous variable indicating the difference between the driver's work time and the minimum number of hours to be worked in a day. w^2_k is a continuous variable indicating the difference between the driver's work time and the normal work time. Let g_i be the time between due date and first service start for customer i.

The objective function minimizes the travel cost, the penalty costs incurred when customer demands are not fully met, the lateness of each customer's first delivery, the cost of drivers working less than the minimum hours, and the total overtime cost of drivers working beyond their scheduled hours. The mathematical model of this variant of the CDP is as follows:

$$\min \sum_{(i,j,k)\in A} t_{ij} x^k_{ij} + \beta_1 \sum_{o\in\mathcal{O}} (1 - y_o) + \beta_2 \sum_{i\in\mathcal{C}} g_i + \sum_{k\in K} \beta_3 * w^1_k + \beta_4 * w^2_k \tag{2}$$

$$\sum_{j\in\delta^+(s_k)} x^k_{s_k j} = 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{j\in\delta^-(e_k)} x^k_{j e_k} = 1 \qquad \forall k \in K \tag{4}$$

$$v_j \geq w_i + \alpha_i + t_{ij} - M\left(1 - x^k_{ij}\right) \qquad \forall i \in \mathcal{L}, j \in \delta^+(i), k \in K \tag{5}$$

$$v_j \geq w_i + \beta_k + t_{ij} - M\left(1 - x^k_{ij}\right) \qquad \forall i \in \mathcal{D}, j \in \delta^+(i), k \in K \tag{6}$$

$$w_i \geq v_i + \frac{q^k_j}{\tau^l_b} - M\left(1 - x^k_{ij}\right) \qquad \forall b \in \mathcal{B}, i \in \mathcal{L}_b, j \in \delta^+(i), k \in K \tag{7}$$

$$w_j \geq v_j + \frac{q^k_j}{\tau^u_c} - M\left(1 - x^k_{ij}\right) \qquad \forall c \in \mathcal{C}, j \in \mathcal{D}_c, i \in \delta^-(j), k \in K \tag{8}$$

$$w_j \leq v_i + \Delta + M\left(1 - x^k_{ij}\right) \qquad j \in \mathcal{D}, i \in \delta^-(j), k \in K \tag{9}$$

$$v_{d^0_o} \geq a_i \qquad \forall i \in \mathcal{C}, \forall o \in \mathcal{O}_i \tag{10}$$

$$g_i \geq v_{d^0_{o_1}} - a_i - M\left(\sum_{j\in\delta^-_D(d^0_{o_1})} u_{j,d^0_{o_1}}\right) \qquad \forall i \in \mathcal{C}, \forall o_1 \in \mathcal{O}_i, d = d^0_{o_1} \tag{11}$$

$$v_{d^0_{o_1}} \geq w_j - M\left(1 - u_{j,d^0_{o_1}}\right) \qquad \forall o_1 \in \mathcal{O}_i, j \in \delta^-_D(d^0_{o_1}) \quad (12)$$

$$v_{d^0_{o_1}} \leq w_j + \gamma_i + M\left(1 - u_{j,d^0_{o_1}}\right) \qquad \forall o_1 \in \mathcal{O}, j \in \delta^-_D(d^0_{o_1}) \quad (13)$$

$$\sum_{o_1 \in \mathcal{O}_i} \sum_{j \in \delta^-_D(d^0_{o_1})} u_{j,d^0_{o_1}} = |\mathcal{O}_i| - 1 \qquad \forall i \in \mathcal{C}, |\mathcal{O}_i| > 1 \quad (14)$$

$$\sum_{o_1 \in \mathcal{O}_i} \sum_{j \in \delta^+_D(d^0_{o_1})} u_{d^0_{o_1},j} = |\mathcal{O}_i| - 1 \qquad \forall i \in \mathcal{C}, |\mathcal{O}_i| > 1 \quad (15)$$

$$\sum_{j \in \delta^+_D(d^0_{o_1})} u_{d^0_{o_1},j} \leq 1 \qquad \forall o_1 \in \mathcal{O} \quad (16)$$

$$\sum_{j \in \delta^-_D(d^0_{o_1})} u_{j,d^0_{o_1}} \leq 1 \qquad \forall o_1 \in \mathcal{O} \quad (17)$$

$$\sum_{j \in \delta^+_D(d^0_{o_1})} u_{d^0_{o_1},j} + \sum_{j \in \delta^-_D(d^0_{o_1})} u_{j,d^0_{o_1}} \geq 1 \qquad \forall o_1 \in \mathcal{O} \quad (18)$$

$$v_j \geq w_{j-1} - M\left(1 - u_{j-1,j}\right) \qquad \forall o \in \mathcal{O}, j \in \mathcal{D}_o, j \geq 1 \quad (19)$$

$$v_j \leq w_{j-1} + \gamma_i + M\left(1 - u_{j-1,j}\right) \qquad \forall i \in \mathcal{C}, \forall o \in O_i, j \in \mathcal{D}_o, j \geq 1 \quad (20)$$

$$u_{j-1,j} \geq u_{j,j+1} \qquad \forall o_1 \in \mathcal{O}, j \in \mathcal{D}_{o_1}, 1 \leq j \leq n_{o_1} - 1 \quad (21)$$

$$u_{j-1,j} \geq \sum_{l \in \mathcal{L}} x_{lj} \qquad \forall o_1 \in \mathcal{O}, j \in \mathcal{D}_{o_1}, j \geq 1 \quad (22)$$

$$v_j \geq w_i - M\left(1 - u_{i,j}\right) \qquad \forall i \in \mathcal{L}, j \in \delta^+_L(i) \quad (23)$$

$$\sum_{j \in \delta^+_L(i)} u_{i,j} \leq 1 \qquad \forall i \in \mathcal{L} \quad (24)$$

$$\sum_{j \in \delta^-_L(i)} u_{j,i} \leq 1 \qquad \forall i \in \mathcal{L} \quad (25)$$

$$\sum_{j \in \delta^-_L(i)} u_{j,i} + \sum_{j \in \delta^+_L(i)} u_{i,j} \geq \sum_{k \in K} \sum_{j \in \delta^-(i)} x^k_{ji} \qquad \forall i \in \mathcal{L} \quad (26)$$

$$\sum_{k \in K} \sum_{j \in \mathcal{D}_o} q^k_j = q_o \qquad \forall o \in \mathcal{O} \quad (27)$$

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} x^k_{ij} \leq 1 \qquad i \in \mathcal{L} \cup \mathcal{D} \quad (28)$$

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} x^k_{ij} = \sum_{k \in K} \sum_{j \in \delta^-(i)} x^k_{ji} \qquad i \in \mathcal{L} \cup \mathcal{D} \quad (29)$$

$$\sum_{k \in K} \sum_{i \in \mathcal{D}_c} \sum_{j \in \delta^-(i)} x^k_{ji} \leq M * \sigma_{c,b} \qquad \forall c \in \mathcal{C}, b \in \mathcal{B} \quad (30)$$

$$\sum_{b \in \mathcal{B}} \sigma_{c,b} \leq 1 \qquad \forall c \in \mathcal{C}, b \in \mathcal{B} \quad (31)$$

$$w^1_k \geq M_T + H_k - v_{e_k} \qquad \forall k \in K \quad (32)$$

$$w^2_k \geq (v_{e_k} - H_k) - N_T \qquad \forall k \in K \quad (33)$$

$$w_{e_k} \leq H_k + O_T \qquad k \in K \quad (34)$$

$$0 \leq q^k_j \leq Q^k \qquad j \in \mathcal{D}, k \in K \quad (35)$$

$$x_{ij}^k \in \{0,1\} \quad (i,j) \in A, k \in K \quad (36)$$

$$u_{ij} \in \{0,1\} \quad (i,j) \in A^D, k \in K \quad (37)$$

$$\sigma_{ib} \in \{0,1\} \quad i \in \mathcal{C}, b \in \mathcal{B} \quad (38)$$

$$y_o \in \{0,1\} \quad o \in \mathcal{O} \quad (39)$$

$$v_i \geq 0, w_i \geq 0 \quad i \in V \quad (40)$$

$$w_k^1 \geq 0, w_k^2 \geq 0 \quad i \in V \quad (41)$$

$$g_i \geq 0 \quad i \in \mathcal{C}. \quad (42)$$

β_1 to β_4 are the penalty coefficients of each component of the objective function (2). Constraints (3–4) state that a driver k leaves his start node exactly once a day and returns to his end node after completing his last unloading operation of the day. Constraints (5–6) set a driver to take some time after loading or unloading to adjust the concrete or clean the truck before moving on to the next node. The duration of the loading operation depends on the plant's loading rate and the amount q_j^k of RMC loaded (Eq. 7). Similarly, the unloading service depends on the site's rate and q_j^k (Eq. 8). Unloading operations must end Δ minutes after loading begins (Eq. 9).

Constraints (10–13) ensure that the first service of any customer i must start at the due time a_i. This first service may be performed at the first delivery node of any order $\in O_i$, and one order must be completed before another is started. It also enforces the precedence constraints between the last delivery of an order and the first delivery of the following order. Constraints (14–18) find the delivery sequence of all orders $\in \mathcal{O}_i$. Constraints (19–22) prohibit two trucks serving consecutive deliveries of the same order from unloading at the same time and enforce a maximum time delay between them. Similarly, constraints (23–26) ensure that two trucks cannot be loaded at the same time at a plant. Constraints (27) require that the cumulative load of all concrete mixers serving an order must equal the required quantities. Constraints (28) require that a driver can only visit a loading/delivery node once. Constraints (29) are degree constraints. Constraints (30) and (31) require that all orders from customer i must be loaded from the same plant. Constraints (32–33) calculate the difference between a driver's hours of service and the minimum and normal hours of service. Finally, constraints (34)–(42) define the nature and bounds of the variables.

Let us consider the solution of a small instance of our problem shown in Figures 1 and 2. Two batching plants B_1 and B_2 with three drivers serve two construction sites. Customer C_1 requests one order of $q_1^1 = 12$ m^3. Customer C_2 requests three orders of $q_2^1 = 3$, $q_2^2 = 11$, and $q_2^3 = 1$ m^3 of three different types of concrete. B_1 has two drivers (D_1, D_2) using concrete mixers of capacity 8 m^3, and B_2 has D_3 with a capacity 12 m^3. Figure 1 shows the flow of concrete mixers flow in the network. We schedule D_1 and D_2 to deliver 8 and 4 m^3 to C_1, respectively. We select o_2^2 as the first order of C_2 to deliver. D_3 travels to B_1 to load q_2^2, then visits C_2 before returning to his home plant. Meanwhile, after their first trip, D_1 and D_2 deliver the remaining orders of C_2.

Figure 2 shows the sequence of loading and unloading operations. The first deliveries of C_1 and C_2 are timely and there is no gap between the two deliveries received by C_1. There is a delay between the end of the first delivery and the start of the second delivery of C_2, however, it is less than the maximal time delay of 20 minutes defined for this example.

4 Constructive heuristics and GRASP

We now present the heuristic approach we implement to solve this variant of the Concrete Delivery Problem. Our method consists of constructing feasible solutions to the CDP with

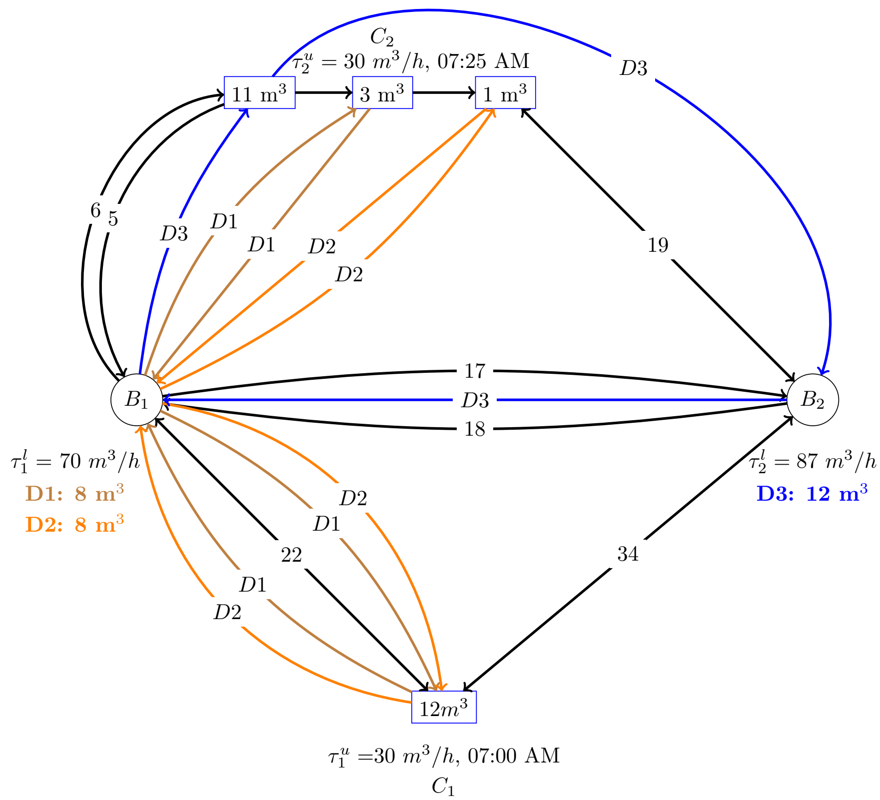

Figure 1: Solution of an instance with two plants, two construction sites, four orders, and three drivers.

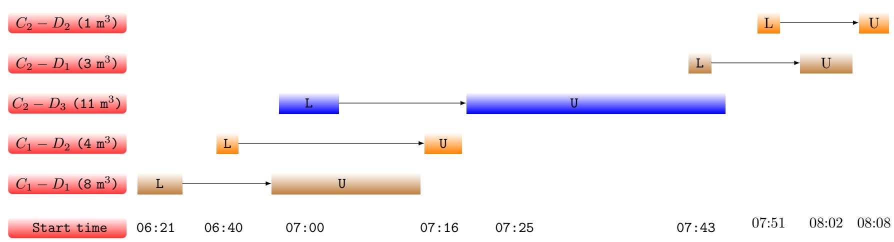

Figure 2: Schedule of the instance of Figure 1.

randomized heuristics and iteratively invoking these heuristics in the GRASP metaheuristic.

In this section, we represent the scheduling of a delivery node d as a pair of loading (L_d) and unloading (U_d) tasks. $L_d = (b, k, q_d, v_b)$ indicates that driver k starts loading q_d m^3 of RMC at plant b at time v_b. $U_d = (k, v_d, w_d)$ indicates that driver k is serving d between v_d and w_d. We then call a solution $S = \left\{ \cup_{1 \leq j \leq n_o} (L_{d_o^j}, U_{d_o^j}), \forall o \in \mathcal{O} \right\}$ the set of all loading and unloading tasks performed on a given day.