Supplementary Material for
A Use-Case-Oriented Benchmark Framework for Classical, Quantum, Hybrid, and Quantum-Inspired Optimization

Dominik Leib Tobias Seidel Sebastian Velten Michael Bortz

This supplementary document collects the use cases and model formulations that are implemented in the framework but are not discussed in detail in the main manuscript. In particular, it covers Maximum Independent Set, Container Scheduling, and Portfolio Optimization. For consistency with the other formulation blocks, it also records the explicit Constrained Quadratic Model (CQM) variants of the vehicle-routing use case that are only summarized in the main appendix.

S1 Additional Use Cases and Model Formulations

S1.1 Maximum Independent Set

Given an undirected graph $G = (V, E)$, $n := |V|$, with nonnegative node weights $w : V \to \mathbb{R}_{\geq 0}$, the Maximum Independent Set problem is to find an independent set $S \subseteq V$, that is, a subset of nodes such that no two nodes in S are adjacent, with maximum total weight:

$$\max_{S \subseteq V} \sum_{i \in S} w_i.$$

Parameter	Definition	Explanation
adjacency_matrix	Matrix $A \in \{0,1\}^{\lvert V\rvert \times \lvert V\rvert}$	Adjacency matrix of G
node_weights	Vector $w \in \mathbb{R}_{\geq 0}^{\lvert V\rvert}$	Weights of the graph nodes

Table S1: MaximumIndependentSetProblem Parameters

S1.1.1 MaximumIndependentSetMIP (MISMIP)

The MISMIP reads

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} w_i \, x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \qquad \forall (i,j) \in E, \end{aligned}$$

where

$$x_i \in \{0,1\} \quad (i = 1, \ldots, n), \qquad x_i = \begin{cases} 1, & \text{if node } i \text{ is contained in the independent set,} \\ 0, & \text{otherwise.} \end{cases}$$

S1.1.2 MaximumIndependentSetQUBO (MISQUBO)

The MISQUBO reads

$$\min_{x\in\{0,1\}^n} \; E(x) = \lambda\, P_1(x) + F(x),$$

where

$$x_i \in \{0,1\} \quad (i = 1,\ldots,n), \qquad x_i = \begin{cases} 1, & \text{if node } i \text{ is contained in the independent set,} \\ 0, & \text{otherwise.} \end{cases}$$

with the penalty term

$$P_1(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\, x_i x_j, \tag{P_1}$$

and objective term

$$F(x) = -\sum_{i=1}^{n} w_i\, x_i. \tag{F}$$

The penalty factor is set by default as

$$\lambda = \frac{\max_{\substack{i,j=1,\ldots,n \\ A_{ij}>0}} (w_i + w_j) + 1}{2}.$$

S1.1.3 MaximumIndependentSetCQM (MISCQM)

The MISCQM is the constrained-quadratic counterpart of MISMIP. It uses the same binary node-selection variables and the same independence constraints, but represents them in the CQM interface. The MISCQM reads

$$\begin{aligned} \min \quad & -\sum_{i=1}^{n} w_i\, x_i \\ \text{s.t.} \quad & x_i + x_j \le 1 \qquad \forall (i,j) \in E, \end{aligned}$$

where

$$x_i \in \{0,1\} \quad (i = 1,\ldots,n), \qquad x_i = \begin{cases} 1, & \text{if node } i \text{ is contained in the independent set,} \\ 0, & \text{otherwise.} \end{cases}$$

S1.2 Container Scheduling

Let $\mathcal{B} = \{0,\ldots,B-1\}$ be the set of unloading bays, $\mathcal{C} = \{0,\ldots,N-1\}$ the set of containers, and $\mathcal{T} = \{0,\ldots,T-1\}$ the discrete planning horizon. For each container $j \in \mathcal{C}$, let a_j and b_j denote the earliest and latest admissible unloading start time, let p_j be the unloading duration, let $D_{ij} \in \{0,1\}$ indicate whether bay $i \in \mathcal{B}$ is admissible for container j, and let $E_j \subseteq \mathcal{C}$ contain the containers that must be scheduled in the same bay group as j. Moreover, let $\mathcal{G}_1,\ldots,\mathcal{G}_q$ be a partition of the bays into bay groups. The container scheduling problem is to assign each container exactly one admissible bay and one admissible start time such that no overlap occurs on any bay, grouped containers are assigned to the same bay group, and the sum of unloading start times is minimized.

For compact notation, let

$$\Omega := \{(i,j,t) \in \mathcal{B}\times\mathcal{C}\times\mathcal{T} \mid a_j \le t \le b_j,\ t + p_j \le T + 1,\ D_{ij} = 1\}$$

be the set of admissible bay-container-time triplets.

Parameter	Definition	Explanation
num_bays	Integer $B \in \mathbb{N}_{>0}$	Number of unloading bays
T	Integer $T \in \mathbb{N}_{>0}$	Number of discrete time steps in the planning horizon
container_data	Sequence $\left([a_j, b_j, p_j, D_j, E_j]\right)_{j\in\mathcal{C}}$	Container data with time window, duration, bay feasibility vector, and grouping information
bay_groups	Partition $(\mathcal{G}_1, \dots, \mathcal{G}_q)$ of $\mathcal{B}$	Groups of bays that must be used consistently for grouped containers

Table S2: ContainerSchedulingProblem Parameters

S1.2.1 ContainerSchedulingMIP (CSMIP)

The CSMIP reads

$$\min_x \quad \sum_{(i,j,t)\in\Omega} t\, x_{i,j,t}$$
$$\text{s.t.} \quad \text{C1, C2, C3}$$

where

$$x_{i,j,t} \in \{0,1\} \quad ((i,j,t)\in\Omega), \qquad x_{i,j,t} = \begin{cases} 1, & \text{if container } j \text{ starts unloading at bay } i \text{ at time } t,\\ 0, & \text{otherwise.} \end{cases}$$

with the constraints

i) Each container is scheduled exactly once:

$$\sum_{i\in\mathcal{B}} \sum_{\substack{t\in\mathcal{T}\\(i,j,t)\in\Omega}} x_{i,j,t} = 1 \quad \forall j \in \mathcal{C}. \tag{C1}$$

ii) No two containers overlap on the same bay:

$$x_{i,j,t} + x_{i,j',t'} \le 1 \tag{C2}$$

for all $i \in \mathcal{B}$, $j, j' \in \mathcal{C}$ with $j \neq j'$, and all $(i,j,t),(i,j',t') \in \Omega$ such that

$$[t, t+p_j-1] \cap [t', t'+p_{j'}-1] \neq \emptyset.$$

iii) Grouped containers must be assigned to the same bay group:

$$x_{i,j,t} + x_{i',j',t'} \le 1 \tag{C3}$$

for all $j \in \mathcal{C}$, all $j' \in E_j$ with $j' \neq j$, all $g \neq h$, all $i \in \mathcal{G}_g$, $i' \in \mathcal{G}_h$, and all admissible $(i,j,t),(i',j',t') \in \Omega$.

S1.2.2 ContainerSchedulingQUBO (CSQUBO)

The CSQUBO reads

$$\min_{x\in\{0,1\}^{|\Omega|}} \; E(x) = \lambda_1\, P_1(x) + \lambda_2\, P_2(x) + \lambda_3\, P_3(x) + F(x),$$

where

$$x_{i,j,t} \in \{0,1\} \quad ((i,j,t)\in\Omega), \qquad x_{i,j,t} = \begin{cases} 1, & \text{if container } j \text{ starts unloading at bay } i \text{ at time } t,\\ 0, & \text{otherwise.}\end{cases}$$

with the penalty terms

i) Each container is scheduled exactly once:

$$P_1(x) = \sum_{j\in\mathcal{C}} \left(\sum_{i\in\mathcal{B}} \sum_{\substack{t\in\mathcal{T}\\(i,j,t)\in\Omega}} x_{i,j,t} - 1 \right)^2. \tag{P_1}$$

ii) No two containers overlap on the same bay:

$$P_2(x) = \sum_{i\in\mathcal{B}} \sum_{j\in\mathcal{C}} \sum_{\substack{t\in\mathcal{T}\\(i,j,t)\in\Omega}} x_{i,j,t} \sum_{\substack{j'\in\mathcal{C}\\j'\neq j}} \sum_{t'=t}^{\min(t+p_j-1,T-1)} x_{i,j',t'}. \tag{P_2}$$

iii) Grouped containers are assigned to the same bay group:

$$P_3(x) = \sum_{j\in\mathcal{C}} \sum_{g=1}^{q} \sum_{\substack{h=1\\h\neq g}}^{q} \sum_{i\in\mathcal{G}_g} \sum_{i'\in\mathcal{G}_h} \sum_{\substack{t\in\mathcal{T}\\(i,j,t)\in\Omega}} \sum_{t'\in\mathcal{T}} \sum_{\substack{j'\in E_j\\j'\neq j}} x_{i,j,t} x_{i',j',t'}. \tag{P_3}$$

The objective term is

$$F(x) = \sum_{(i,j,t)\in\Omega} t\, x_{i,j,t}. \tag{F}$$

The penalty factors are set by default as

$$\lambda_1 = \lambda_2 = \lambda_3 = T + 1.$$

S1.2.3 ContainerSchedulingCQM (CSCQM)

The CSCQM is the constrained-quadratic counterpart of CSMIP. It uses the same binary start variables and the same scheduling objective, but represents feasibility directly in the CQM interface. The CSCQM reads

$$\min_{x} \quad \sum_{(i,j,t)\in\Omega} t\, x_{i,j,t}$$
$$\text{s.t.} \quad \text{C1, C2, C3}$$

where

$$x_{i,j,t} \in \{0,1\} \quad ((i,j,t)\in\Omega), \qquad x_{i,j,t} = \begin{cases} 1, & \text{if container } j \text{ starts unloading at bay } i \text{ at time } t,\\ 0, & \text{otherwise.}\end{cases}$$

with the constraints

i) Each container is scheduled exactly once:

$$\sum_{i \in \mathcal{B}} \sum_{\substack{t \in \mathcal{T} \\ (i,j,t) \in \Omega}} x_{i,j,t} = 1 \quad \forall j \in \mathcal{C}. \tag{C1}$$

ii) At each bay and time step, at most one container may be processed:

$$\sum_{j \in \mathcal{C}} \sum_{\substack{t \in \mathcal{T} \\ (i,j,t) \in \Omega,\, t \le \tau < t + p_j}} x_{i,j,t} \le 1 \quad \forall i \in \mathcal{B},\ \tau \in \mathcal{T}. \tag{C2}$$

iii) Grouped containers must be assigned consistently to each bay group:

$$\sum_{i \in \mathcal{G}_g} \sum_{\substack{t \in \mathcal{T} \\ (i,j,t) \in \Omega}} x_{i,j,t} - \sum_{i \in \mathcal{G}_g} \sum_{\substack{t \in \mathcal{T} \\ (i,j',t) \in \Omega}} x_{i,j',t} = 0 \tag{C3}$$

for all $j \in \mathcal{C}$, all $j' \in E_j$ with $j' \neq j$, and all $g = 1, \ldots, q$.

S1.3 Portfolio Optimization

Let $\mathcal{I} = \{0, \ldots, n-1\}$ be the set of brands and, for each brand $i \in \mathcal{I}$, let $\mathcal{F}_i = \{0, \ldots, m_i - 1\}$ be the set of admissible facings. Selecting facing $j \in \mathcal{F}_i$ of brand i incurs cost $c_{ij} \ge 0$ and yields profit $w_{ij} \ge 0$. Moreover, let $C \in \mathbb{N}_{>0}$ denote the available budget, let $\mathcal{V} \subseteq \mathcal{I} \times \mathcal{I}$ be the set of *Verbund* pairs that must be selected jointly or not at all, and let $\mathcal{L} \subseteq \mathcal{I} \times \mathcal{I}$ be the set of brand pairs with performance-discount factors $\alpha_{ab} \ge 0$. If both brands a and b are selected and $(a,b) \in \mathcal{L}$, the achieved profit is reduced by the fraction α_{ab} of the combined selected profit of that pair. The portfolio optimization problem is therefore to choose at most one facing per brand such that the budget and coupling constraints are satisfied and the discounted total profit is maximized.

Parameter	**Definition**	**Explanation**
weights_values	Nested mapping $\big((c_{ij}, w_{ij})\big)_{i \in \mathcal{I},\, j \in \mathcal{F}_i}$	Cost and profit for each brand-facing combination
budget	Integer $C \in \mathbb{N}_{>0}$	Total investment budget
verbund	Set $\mathcal{V} \subseteq \mathcal{I} \times \mathcal{I}$	Brand pairs that must be selected jointly or omitted jointly
leistungsabschlag	Mapping $(\alpha_{ab})_{(a,b) \in \mathcal{L}}$ with $\alpha_{ab} \ge 0$	Pairwise discount factors applied when both brands of a listed pair are selected

Table S3: PortfolioOptimizationProblem Parameters

S1.3.1 PortfolioOptimizationMIP (POMIP)

The POMIP reads

$$\begin{aligned} \max_{x,y,z} \quad & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{F}_i} w_{ij}\, x_{ij} - \sum_{(a,b) \in \mathcal{L}} \alpha_{ab}\, z_{ab} \\ \text{s.t.} \quad & \text{C1, C2, C3, C4, C5} \end{aligned}$$

where

$$x_{ij} \in \{0,1\} \quad (i \in \mathcal{I},\ j \in \mathcal{F}_i), \qquad x_{ij} = \begin{cases} 1, & \text{if facing } j \text{ of brand } i \text{ is selected,} \\ 0, & \text{otherwise,} \end{cases}$$

$$y_{ab} \in \{0,1\} \quad ((a,b) \in \mathcal{L}), \qquad y_{ab} = \begin{cases} 1, & \text{if both brands } a \text{ and } b \text{ are selected,} \\ 0, & \text{otherwise,} \end{cases}$$

and $z_{ab} \geq 0$ for all $(a,b) \in \mathcal{L}$. Moreover, define

$$G_{ab}(x) := \sum_{j \in \mathcal{F}_a} w_{aj} x_{aj} + \sum_{j \in \mathcal{F}_b} w_{bj} x_{bj} \qquad ((a,b) \in \mathcal{L}).$$

The constraints are

i) At most one facing is selected for each brand:

$$\sum_{j \in \mathcal{F}_i} x_{ij} \leq 1 \quad \forall i \in \mathcal{I}. \tag{C1}$$

ii) The total investment does not exceed the budget:

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{F}_i} c_{ij} x_{ij} \leq C. \tag{C2}$$

iii) Verbund pairs are selected jointly or not at all:

$$\sum_{j \in \mathcal{F}_a} x_{aj} - \sum_{j \in \mathcal{F}_b} x_{bj} = 0 \quad \forall (a,b) \in \mathcal{V}. \tag{C3}$$

iv) The binary variables y_{ab} indicate whether both brands of a discounted pair are selected:

$$\sum_{j \in \mathcal{F}_a} x_{aj} + \sum_{j \in \mathcal{F}_b} x_{bj} - y_{ab} \leq 1, \qquad \sum_{j \in \mathcal{F}_a} x_{aj} + \sum_{j \in \mathcal{F}_b} x_{bj} \geq 2 y_{ab} \tag{C4}$$

for all $(a,b) \in \mathcal{L}$.

v) The auxiliary variables z_{ab} linearize the discounted pair profit:

$$z_{ab} \leq G_{ab}(x), \qquad G_{ab}(x) - z_{ab} \leq M(1 - y_{ab}), \qquad z_{ab} \leq M y_{ab} \tag{C5}$$

for all $(a,b) \in \mathcal{L}$, where M is a sufficiently large constant.

S1.3.2 PortfolioOptimizationQUBO (POQUBO)

The POQUBO reformulates the problem as a binary quadratic minimization problem. Besides the facing variables x_{ij}, it introduces binary brand-indicator variables u_i and binary slack variables s_r to encode the budget inequality. Let

$$L = \lfloor \log_2(C) \rfloor + 1$$

and define the slack expression

$$S(s) = \sum_{r=0}^{L-2} 2^r s_r + \big(C + 1 - 2^{L-1}\big) s_{L-1}.$$

Then the POQUBO reads

$$\min_{x,u,s} \; E(x,u,s) = \lambda_1 \, P_1(x,s) + \lambda_2 \, P_2(x,u) + \lambda_3 \, P_3(u) + F(x),$$

where

$$x_{ij}, u_i, s_r \in \{0,1\} \qquad (i \in \mathcal{I},\; j \in \mathcal{F}_i,\; r = 0, \dots, L-1),$$

with

$$x_{ij} = \begin{cases} 1, & \text{if facing } j \text{ of brand } i \text{ is selected,} \\ 0, & \text{otherwise,} \end{cases} \qquad u_i = \begin{cases} 1, & \text{if some facing of brand } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

The penalty terms are

i) Budget limit:

$$P_1(x,s) = \left(\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{F}_i} c_{ij} x_{ij} + S(s) - C \right)^2. \tag{P_1}$$

ii) The brand-indicator variables are coupled to the facing choices:

$$P_2(x,u) = \sum_{i \in \mathcal{I}} \left(\sum_{j \in \mathcal{F}_i} x_{ij} - u_i \right)^2. \tag{P_2}$$

iii) Verbund pairs are selected jointly:

$$P_3(u) = \sum_{(a,b) \in \mathcal{V}} \frac{1}{2} \big(u_a (1 - u_b) + u_b (1 - u_a) \big). \tag{P_3}$$

The objective term is

$$F(x) = -\left(\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{F}_i} w_{ij} x_{ij} - \sum_{(a,b) \in \mathcal{L}} \alpha_{ab} \sum_{j \in \mathcal{F}_a} \sum_{\ell \in \mathcal{F}_b} (w_{aj} + w_{b\ell}) x_{aj} x_{b\ell} \right). \tag{F}$$

The default penalty factors are set uniformly as

$$\lambda_1 = \lambda_2 = \lambda_3 = \max_{i \in \mathcal{I},\, j \in \mathcal{F}_i} w_{ij} + 1.$$

S1.3.3 PortfolioOptimizationCQM (POCQM)

The POCQM is the constrained-quadratic counterpart of POMIP. It uses the same facing variables and explicit hard constraints, but minimizes the negated discounted profit in the CQM interface. The POCQM reads

$$\begin{aligned}\min_{x,y} \quad & -\left(\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{F}_i} w_{ij}x_{ij} - \sum_{(a,b)\in\mathcal{L}} \alpha_{ab}y_{ab}G_{ab}(x)\right)\\ \text{s.t.} \quad & \text{C1, C2, C3, C4}\end{aligned}$$

where

$$x_{ij} \in \{0,1\} \quad (i\in\mathcal{I},\ j\in\mathcal{F}_i), \qquad y_{ab}\in\{0,1\} \quad ((a,b)\in\mathcal{L}).$$

The constraints are

i) At most one facing is selected for each brand:

$$\sum_{j\in\mathcal{F}_i} x_{ij} \le 1 \quad \forall i\in\mathcal{I}. \tag{C1}$$

ii) The total investment does not exceed the budget:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{F}_i} c_{ij}x_{ij} \le C. \tag{C2}$$

iii) Verbund pairs are selected jointly or not at all:

$$\sum_{j\in\mathcal{F}_a} x_{aj} - \sum_{j\in\mathcal{F}_b} x_{bj} = 0 \quad \forall (a,b)\in\mathcal{V}. \tag{C3}$$

iv) The binary variables y_{ab} indicate whether both brands of a discounted pair are selected:

$$\sum_{j\in\mathcal{F}_a} x_{aj} + \sum_{j\in\mathcal{F}_b} x_{bj} - y_{ab} \le 1, \qquad \sum_{j\in\mathcal{F}_a} x_{aj} + \sum_{j\in\mathcal{F}_b} x_{bj} - 2y_{ab} \ge 0 \tag{C4}$$

for all $(a,b)\in\mathcal{L}$.

S1.4 Vehicle Routing: Detailed CQM Variants

For notation, let $V = \{0,1,\ldots,N-1\}$ be the set of locations with depot 0 and customers $V^\star = V\setminus\{0\}$. Let $D = (d_{uv})_{u,v\in V}$ be a nonnegative distance matrix, let $M\in\mathbb{N}_{>0}$ be the number of vehicles, and let $K\in\mathbb{N}_{>0}$ be the maximum number of customer stops per vehicle.

S1.4.1 VehicleRoutingCQM (VRPCQM)

The VRPCQM is the constrained-quadratic counterpart of the native arc-based formulation VehicleRoutingMIP (VRPMIP). It reads

$$\begin{aligned}\min_{x,u} \quad & \sum_{u\in V}\sum_{\substack{v\in V\\ v\neq u}} d_{uv}\,x_{uv}\\ \text{s.t.} \quad & \text{C1, C2, C3, C4}\end{aligned}$$

where

$$x_{uv} \in \{0,1\} \quad (u,v \in V;\ u \neq v), \quad x_{uv} = \begin{cases} 1, & \text{if edge } (u,v) \text{ is used in some route,} \\ 0, & \text{otherwise,} \end{cases}$$
$$u_i \in \{0,\ldots,K-1\} \quad (i \in V^\star), \quad u_i = k \text{ if customer } i \text{ is assigned route position } k.$$

The constraints are

i) Each customer is left exactly once:

$$\sum_{\substack{v \in V \\ v \neq i}} x_{iv} = 1 \quad \forall i \in V^\star. \tag{C1}$$

ii) Each customer is reached exactly once:

$$\sum_{\substack{u \in V \\ u \neq j}} x_{uj} = 1 \quad \forall j \in V^\star. \tag{C2}$$

iii) Exactly M vehicles leave the depot:

$$\sum_{v \in V^\star} x_{0v} = M. \tag{C3}$$

iv) MTZ coupling constraints prevent subtours and enforce the stop bound:

$$u_i - u_j + K\, x_{ij} \leq K - 1 \quad \forall i,j \in V^\star,\ i \neq j. \tag{C4}$$

This is exactly the native vehicle-routing formulation implemented in the library's CQM interface: the objective remains linear in the arc-selection variables, while the routing semantics are enforced by explicit hard constraints.

S1.4.2 VehicleRoutingBinaryCQM (VRPCQMB)

The VRPCQMB is the constrained-quadratic counterpart of the binary position-indexed formulation VehicleRoutingBinaryMIP (VRPMIPB). It uses the same assignment variables and the same route-length decomposition as VehicleRoutingQUBO (VRPQUBO) and VRPMIPB, but keeps feasibility conditions as explicit constraints. The formulation reads

$$\begin{aligned} \min_{x \in \{0,1\}^{(N-1)MK}} \quad & F(x) \\ \text{s.t.} \quad & \text{C1, C2, C3, C4} \end{aligned}$$

where

$$x_{i,j,k} \in \{0,1\} \quad (i \in V^\star,\ j = 0,\ldots,M-1,\ k = 0,\ldots,K-1),$$

with

$$x_{i,j,k} = \begin{cases} 1, & \text{if customer } i \text{ is the } (k+1)\text{-th stop of vehicle } j, \\ 0, & \text{otherwise.} \end{cases}$$

The objective is again the total route length,

$$F(x) = F_{\text{first}}(x) + F_{\text{inner}}(x) + F_{\text{last}}(x),$$

where

$$F_{\text{first}}(x) = \sum_{j=0}^{M-1} \sum_{i \in V^\star} d_{0i}\, x_{i,j,0},$$

$$F_{\text{inner}}(x) = \sum_{j=0}^{M-1} \sum_{k=0}^{K-2} \sum_{u \in V^\star} \sum_{\substack{v \in V^\star \\ v \neq u}} d_{uv}\, x_{u,j,k}\, x_{v,j,k+1},$$

and

$$F_{\text{last}}(x) = \sum_{j=0}^{M-1} \left[\sum_{u \in V^\star} d_{u0}\, x_{u,j,K-1} + \sum_{k=0}^{K-2} \sum_{u \in V^\star} d_{u0}\, x_{u,j,k} \left(1 - \sum_{v \in V^\star} x_{v,j,k+1} \right) \right].$$

The constraints are

i) Each customer is served exactly once:

$$\sum_{j=0}^{M-1} \sum_{k=0}^{K-1} x_{i,j,k} = 1 \quad \forall i \in V^\star. \tag{C1}$$

ii) Each vehicle occupies at most one customer position at each step:

$$\sum_{i \in V^\star} x_{i,j,k} \leq 1 \quad \forall j = 0, \ldots, M-1,\ k = 0, \ldots, K-1. \tag{C2}$$

iii) Route positions must be consecutive:

$$\sum_{i \in V^\star} x_{i,j,k+1} - \sum_{i \in V^\star} x_{i,j,k} \leq 0 \quad \forall j = 0, \ldots, M-1,\ k = 0, \ldots, K-2. \tag{C3}$$

iv) Every vehicle has exactly one first stop:

$$\sum_{i \in V^\star} x_{i,j,0} = 1 \quad \forall j = 0, \ldots, M-1. \tag{C4}$$

Hence, VRPCQMB keeps the same binary routing logic as VRPQUBO, but without penalty terms: feasibility is enforced directly by the CQM constraints, while the quadratic route-length objective remains unchanged.