A Bilevel Edge Computing Architecture and Collaborative Offloading Mechanism for Offshore Buoys Network (Appendix)

Jiaxuan Xie, *Graduate Student Member, IEEE,* Linling Kuang, *Member, IEEE,* Shan Gao, *Member, IEEE*

III. Problem Solving

B. Joint Iterative Mechanism

2) Solving for Decision Variables $\{\mathcal{P}^b\}$: In this subsection, the derivation of the decision variable $\mathcal{P}^b = \{p_{ij}\}_{i\in\mathcal{N},j\in\mathcal{N}}$ is proposed, when other decision variables are fixed, and the optimization problem P_3 is given as follows.

$$\boldsymbol{P_3^{\text{Sub}}}: \min_{\mathcal{P}^b} \sum_{m=1}^{M}\sum_{q=1}^{Q} T_{qm}^{c1}(\mathcal{P}^b) \tag{32a}$$

$$s.t. \quad C_3, C_4', C_5, C_7, C_9 \tag{32b}$$

The objective function in (32a), regarded as a composition function, is illustrated based on the formula (11), where the outer function of (32a) can be regarded as an inverse proportional function, and the inner function of (32a) is a monotonically increasing function. According to the composition rules, it is easy to know that this objective function is also a convex function with respect to $\{\mathcal{P}^b\}$.

In addition, for $\forall i \in \mathcal{N}$, constraints C_5 and C_9 can be combined and expressed as follows.

$$E_i^{c1} \le c_5, \quad \forall i \in \mathcal{N} = \mathcal{N}^M \cup \mathcal{N}^Q \tag{33}$$

where constant c_5 is calculated as follows.

$$c_5 = \begin{cases} E_0 - E_i^{c2}, \text{ for } C_5 \text{ constraint} \\ E_0 - E_i^{s} - E_i^{c2}, \text{ for } C_9 \text{ constraint} \end{cases} \tag{34}$$

it is evident that E_i^{c1} can be regarded as the function form of $f(x) = \sum_i \frac{x_i}{\log_2(x_i+1)} - c$ based on (13), which is a concave function when $x > 0$. The above conclusion implies that the constraint of formula (33) is a non-convex constraint.

To solve such problem, a novel power iterative algorithm, based on Successive Convex Approximation (SCA) method, is proposed to approximate this non-convex constraint. Specifically, the non-convex constraint E_i^{c1} can be regarded as the sum of two convex functions multiplied together. Then, define a new function $h_q(p_{i\widetilde{\xi}^q})$, which implies that $E_i^{c1} = \sum_{q=1}^{Q} h_q(p_{i\widetilde{\xi}^q})$ holds. We can derive the convex upper bound for each non-convex $h_q(p_{i\widetilde{\xi}^q})$ function in (36). Then, (33) can be converted to convex constraint as follows.

$$C_5', C_9'': \sum_{q=1}^{Q} \widetilde{h}_q(p_{i\widetilde{\xi}^q}; \widetilde{p}_{i\widetilde{\xi}^q}) \le c_5, \forall i \in \mathcal{N} \tag{35}$$

which is a convex constraint with respect to $p_{i\widetilde{\xi}^q}$. In addition, it is worth mentioning that the link $i \to \widetilde{\xi}^q$ from i to $\widetilde{\xi}^q$ is chosen according to the decision variable $x_{i\widetilde{\xi}^q} = 1$, and $\widetilde{\xi}^q$ is the next hop for i within the path $\Gamma_{q\widetilde{m}}$.

Then, C_4' constraint, as mentioned in (18), is also a convex function. Similarly, C_3 and C_7 constraints can also be combined as follows.

$$C_3', C_7'': \sum_{n=1}^{N} p_{in} \le c_6, \quad \forall i \in \mathcal{N} = \mathcal{N}^M \cup \mathcal{N}^Q \tag{37}$$

where constant c_6 is calculated as follows.

$$c_6 = \begin{cases} P_0, \text{ for } C_3 \text{ constraint} \\ P_0 - \sum_{\omega=1}^{W} g_{ms}^{\omega}, \text{ for } C_7 \text{ constraint} \end{cases} \tag{38}$$

Next, the Lagrangian function of P_3^{Sub} is expressed as follows.

$$\begin{aligned} &L_3(\{\mathcal{P}^b\}, \{\mu_i, \gamma_i\}_{i\in\mathcal{N}}, \{\delta_q\}_{q\in\mathcal{N}^Q}) \\ &= \sum_{m=1}^{M}\sum_{q=1}^{Q} T_{qm}^{c1}(\mathcal{P}^b) + \sum_{i=1}^{N}\mu_i\left(\sum_{n=1}^{N} p_{in} - c_6\right) \\ &+ \sum_{i=1}^{N}\gamma_i \cdot \left(\sum_{q=1}^{Q}\widetilde{h}_q(p_{i\widetilde{\xi}^q}; \widetilde{p}_{i\widetilde{\xi}^q}) - c_5\right) \\ &+ \sum_{q=1}^{Q}\Bigg[\delta_q^1 \cdot (-R_{q\to\xi_1} + R_{\min}) + \sum_{k\in\{1,\cdots,K_{q\widetilde{m}}-1\}} \delta_q^k \cdot \left(-R_{\xi_k\to\xi_{k+1}} + R_{\min}\right) \\ &+ \delta_q^{K_{q\widetilde{m}}} \cdot \left(-R_{\xi_{K_{q\widetilde{m}}}\to\widetilde{m}} + R_{\min}\right)\Bigg] \\ &\triangleq \sum_{m=1}^{M} L_3^m(\{p_{m\widetilde{\xi}^m}\}_{m\in\mathcal{N}^M}) + \sum_{q=1}^{Q} L_3^q(\{p_{q\widetilde{\xi}^q}\}_{q\in\mathcal{N}^Q}) + c_7 \end{aligned} \tag{39}$$

where $L_3^m(\{p_{m\widetilde{\xi}^m}\}_{m\in\mathcal{N}^M})$ is defined as the function of decision variables $\{p_{m\widetilde{\xi}^m}\}_{m\in\mathcal{N}^M}$ for gateway buoys $\mathcal{N}^M$, $L_3^q(\{p_{q\widetilde{\xi}^q}\}_{q\in\mathcal{N}^Q})$ is defined as the function of decision variables $\{p_{q\widetilde{\xi}^q}\}_{q\in\mathcal{N}^Q}$ for detection buoys $\mathcal{N}^Q$, and c_7 is the constant that is independent of $\mathcal{P}^b$. We use the symbol $\widetilde{\xi}^q$ to refer to the next hop of buoy q, and the symbol $\widetilde{\xi}^m$ to refer to the next hop of buoy m.

Then, take the partial derivative of the Lagrangian function $L_3^m(\{p_{m\widetilde{\xi}^m}\}_{m\in\mathcal{N}^M})$ with respect to $p_{m\widetilde{\xi}^m}$ for any $m \in \mathcal{N}^M$, take the partial derivative of the Lagrangian function $L_3^q(\{p_{q\widetilde{\xi}^q}\}_{q\in\mathcal{N}^Q})$ with respect to $p_{q\widetilde{\xi}^q}$ for any $q \in \mathcal{N}^Q$. We represent all power variables in terms of $\{p_{ij}\}_{i\in\mathcal{N},j\in\mathcal{N}}$, and a unified representation for updates is formulated as follows:

$$p_{ij}^{(\kappa+1)}=\begin{cases}p_{m\widetilde{\xi}^m}^{(\kappa)}-\zeta_{m\widetilde{\xi}^m}^{(\kappa)}\cdot\frac{\partial L_3^m(\{p_{m\widetilde{\xi}^m}^{(\kappa)}\}_{m\in\mathcal{N}^M})}{\partial p_{m\widetilde{\xi}^m}^{(\kappa)}}, & \text{if } i\in\mathcal{N}^M\wedge i\in\widetilde{\Gamma}\\ p_{q\widetilde{\xi}^q}^{(\kappa)}-\zeta_{q\widetilde{\xi}^q}^{(\kappa)}\cdot\frac{\partial L_3^q(\{p_{q\widetilde{\xi}^q}^{(\kappa)}\}_{q\in\mathcal{N}^Q})}{\partial p_{q\widetilde{\xi}^q}^{(\kappa)}} & \text{if } i\in\mathcal{N}^Q\wedge i\in\widetilde{\Gamma}\\ 0,\ \text{if } i\notin\widetilde{\Gamma}\end{cases}\tag{40}$$

where the calculate of variable $\{p_{ij}\}_{i\in\mathcal{N},j\in\mathcal{N}}$ is divided into three cases. If the node i serves as a gateway buoy and has an associated forwarding task, (i.e., $i\in\mathcal{N}^M\wedge i\in\widetilde{\Gamma}$) holds, then we can use $L_3^m(\{p_{m\widetilde{\xi}^m}\}_{m\in\mathcal{N}^M})$ to update $p_{m\widetilde{\xi}^m}$. In addition, if the node i serves as a detection buoy and has an associated forwarding task, (i.e., $i\in\mathcal{N}^Q\wedge i\in\widetilde{\Gamma}$) holds, then we can use $L_3^q(\{p_{q\widetilde{\xi}^q}\}_{q\in\mathcal{N}^Q})$ to update $p_{q\widetilde{\xi}^q}$. Otherwise, $p_{ij}=0$ when the node i does not have an associated forwarding task.

Let κ represent the index of dual ascent algorithm for outer loop, and let $\widetilde{\kappa}$ represent the index of SCA algorithm for inner loop. Then, the update of $\widetilde{p}_{ij}^{(\widetilde{\kappa}+1)}$ for $\widetilde{\kappa}+1$-th iteration is given as follows.

$$\widetilde{p}_{ij}^{(\kappa,\widetilde{\kappa}+1)}=\arg\min_{\mathcal{P}^b}\sum_{m=1}^{M}\sum_{q=1}^{Q}T_{qm}^{c1}(\mathcal{P}^b)\tag{41a}$$

$$s.t.\begin{cases}C_3',C_7'':\ \sum\limits_{n=1}^{N}p_{in}\le c_6,\quad\forall i\in\mathcal{N}\\ C_5',C_9'':\ \sum\limits_{q=1}^{Q}\widetilde{h}_q(p_{i\widetilde{\xi}^q};\widetilde{p}_{i\widetilde{\xi}^q}^{(\kappa,\widetilde{\kappa}+1)})\le c_5,\forall i\\ C_4'\quad\text{(refer to (??))}\end{cases}\tag{41b}$$

where the above formula can be used to update the auxiliary variables in SCA.

Then, the Lagrange multipliers are calculated as follows.

$$\begin{aligned}\mu_i^{(\kappa+1)}&=\mu_i^{(\kappa)}+\zeta_{m\mu}^{(\kappa)}\cdot\left(\sum_{n=1}^{N}p_{in}^{(\kappa)}-c_6\right)\\ \gamma_i^{(\kappa+1)}&=\gamma_i^{(\kappa)}+\zeta_{m\gamma}^{(\kappa)}\cdot\left(\sum_{q=1}^{Q}\widetilde{h}_q(p_{i\widetilde{\xi}^q};\widetilde{p}_{i\widetilde{\xi}^q})-c_5\right)\\ \delta_q^{k(\kappa+1)}&=\delta_q^{k(\kappa)}+\zeta_{qk\delta}^{(\kappa)}\cdot\left(-R_{\xi_k\to\xi_{k+1}}+R_{\min}\right)\end{aligned}\tag{42}$$

where $\zeta_{m\mu}^{(\kappa)},\zeta_{m\gamma}^{(\kappa)},\zeta_{qk\delta}^{(\kappa)}$ are the step sizes for Lagrange multipliers. Finally, the solution algorithm for decision variables $\{\mathcal{P}^b\}$ is given in Alg. 3.

3) Solving for Decision Variables $\{\mathcal{C}\}$: In this subsection, the derivation of the decision variable $\mathcal{C}$ is proposed, when other decision variables are fixed, and the optimization problem P_4^{Sub} is given as follows.

$$\boldsymbol{P_4^{\text{Sub}}}:\min_{\mathcal{C}}\sum_{m=1}^{M}\sum_{q=1}^{Q}T_{qm}^{s1}(\mathcal{C})\tag{43a}$$

$$s.t.\quad C_2,C_9\tag{43b}$$

where the objective function is an inverse proportional function with respect to $\mathcal{C}=\{c_{qm}\}_{q\in\mathcal{N}^Q,m\in\mathcal{N}^M}$, which is also a convex function. Then, C_9 constraint can be rewritten as

$$E_m^s\le c_8,\quad\forall m\in\mathcal{N}^M\tag{44}$$

where constant c_8 is equals to $E_0-E_m^s$. Then, C_2 constraint and C_9 constraint are linear functions and quadratic functions with respect to $\mathcal{C}$ respectively, and obviously both satisfy the

$$\begin{aligned}h_q(p_{i\widetilde{\xi}^q})&=\frac{\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_q\cdot p_{i\widetilde{\xi}^q}}{B_0\log_2\left(1+\frac{p_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)}=\left(\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_q\cdot p_{i\widetilde{\xi}^q}\right)\cdot\left(\frac{1}{B_0\log_2\left(1+\frac{p_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)}\right)\\ &=\frac{1}{2}\left(\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_qp_{i\widetilde{\xi}^q}+\frac{1}{B_0\log_2\left(1+\frac{p_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)}\right)^2-\frac{1}{2}\left(\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_qp_{i\widetilde{\xi}^q}\right)^2-\frac{1}{2}\left[B_0\log_2\left(1+\frac{p_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)\right]^{-2}\\ &\le H_q(p_{i\widetilde{\xi}^q})-\frac{1}{2}\left(\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_q\widetilde{p}_{i\widetilde{\xi}^q}\right)^2-\frac{1}{2}\left[B_0\log_2\left(1+\frac{\widetilde{p}_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)\right]^{-2}\\ &\quad-\left(\mathbb{I}_{\{i\in\Gamma_{q\widetilde{m}}\}}I_q\right)^2\cdot\widetilde{p}_{i\widetilde{\xi}^q}\cdot(p_{i\widetilde{\xi}^q}-\widetilde{p}_{i\widetilde{\xi}^q})-\frac{-\ln 2\cdot B\cdot\frac{\widetilde{p}_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2+\widetilde{p}_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}}{\left(B\log_2\left(1+\frac{\widetilde{p}_{i\widetilde{\xi}^q}\left|h_{i\widetilde{\xi}^q}^{\text{IoT}}\right|^2}{\sigma_{\widetilde{\xi}^q}^2}\right)\right)^3}\cdot(p_{i\widetilde{\xi}^q}-\widetilde{p}_{i\widetilde{\xi}^q})\\ &\triangleq\quad\widetilde{h}_q(p_{i\widetilde{\xi}^q};\widetilde{p}_{i\widetilde{\xi}^q})\end{aligned}\tag{36}$$

Algorithm 3 Solving for Computing Resources $\{\mathcal{P}^b\}$ for Gateway Guoys

Input: Optimization Problem P_3^{Sub}.
Output: $\{\mathcal{P}^b\} = \{p_{ij}\}_{i\in\mathcal{N},j\in\mathcal{N}}$

Initialize $\{\{p_{ij}^{(0)}\}_{j\in\mathcal{N}}, \mu_i^{(0)}, \gamma_i^{(0)}\}_{i\in\mathcal{N}}, \{\delta_q^{k(0)}\}_{q\in\mathcal{N}^Q,\xi_k\in\Gamma_{q\widetilde{m}}}$.
while (1) **do**
  // Part(a): Dual-Ascent-based Variable Update
  for $m = \{1,2,\cdots,M\}$ **do**
    Calculate the Lagrangian function based on (39).
    Calculate the partial derivatives and update decision variable $p_{ij}^{(\kappa+1)}$ for $\forall i,j$, based on (40).
    Update Lagrange multipliers $\mu_i^{(\kappa+1)}, \gamma_i^{(\kappa+1)}$, $\delta_q^{k(\kappa+1)}$ for $\forall i,q$ based on (42).
  end for
  if $\max\limits_i\{\mu_i^{(\kappa+1)} - \mu_i^{(\kappa)}\} < \varepsilon_\mu \wedge \max\limits_i\{\gamma_i^{(\kappa+1)} - \gamma_i^{(\kappa)}\} < \varepsilon_\gamma \wedge \max\limits_{q,k}\{\delta_q^{(\kappa+1)} - \delta_q^{(\kappa)}\} < \varepsilon_\delta$ **then**
    BreakWhile
  end if
  Update $\kappa = \kappa + 1$.
  // Part(b): SCA-based Power Iteration
  Initialize $\{\widetilde{p}_{ij}^{(\kappa,0)}\}_{i\in\mathcal{N},j\in\mathcal{N}}$ variables.
  while (1) **do**
    Update $\widetilde{p}_{ij}^{(\kappa,\widetilde{\kappa}+1)}$ by solving the convex optimization problem (41) with substitute variable $\widetilde{h}_q(p_{ij}^{(\kappa)}; \widetilde{p}_{ij}^{(\kappa,\widetilde{\kappa})})$ mentioned in (36).
    if $\{\widetilde{p}_{i\widetilde{\xi}^q}^{(\kappa,\widetilde{\kappa}+1)}\}_{i\in\mathcal{N},j\in\mathcal{N}}$ converge **then**
      Break While
    end if
    Update $\widetilde{\kappa} = \widetilde{\kappa} + 1$.
  end while
end while
return $\{\mathcal{P}^b\}$

definition of convex functions. Thus, the Lagrangian function of P_4^{Sub} is expressed as follows.

$$
\begin{aligned}
&L_4(\{\mathcal{C}\}, \{\alpha_m, \beta_m, \eta_m\}_{m\in\mathcal{N}^M}) \\
&= \sum_{m=1}^{M}\sum_{q=1}^{Q} \frac{x_{qm}\phi_q(1-y_{qm})}{c_{qm}} + \sum_{m=1}^{M}\lambda_m\left(\sum_{q=1}^{Q} c_{qm} - C_0\right) \\
&+ \sum_{m=1}^{M}\theta_m\cdot\left(\epsilon_m\sum_{q=1}^{Q} c_{qm}^2 x_{qm}\phi_q(1-y_{qm}) - c_8\right) \\
&\triangleq \sum_{m=1}^{M} L_4^m(\{c_{qm}\}_{m\in\mathcal{N}^M}) + c_9
\end{aligned} \tag{45}
$$

where $L_4^m(\{c_{qm}\}_{m\in\mathcal{N}^M})$ is defined as the function of decision variables $\{\mathcal{C}\}$, ans c_9 is the constant that is independent of $\{\mathcal{C}\}$.

Then, take the partial derivative of the Lagrangian function $L_4^m(\{c_{qm}\}_{m\in\mathcal{N}^M})$ with respect to c_{qm} for any $m \in \mathcal{N}^M$ as

$$
\begin{aligned}
\frac{\partial L_4^m(\{c_{qm}\}_{m\in\mathcal{N}^M})}{\partial c_{qm}} &= -\frac{x_{qm}\phi_q(1-y_{qm})}{c_{qm}^2} + \lambda_m \\
&+ \Big(2\theta_m\epsilon_m x_{qm}\phi_q(1-y_{qm})\Big)\cdot c_{qm}
\end{aligned} \tag{46}
$$

Algorithm 4 Solving for Computing Resources $\{\mathcal{C}\}$ for Gateway Guoys

Input: Optimization Problem P_4^{Sub}.
Output: $\{\mathcal{C}\} = \{c_{qm}\}_{q\in\mathcal{N}^Q,m\in\mathcal{N}^M}$

Initialize $\{\{c_{qm}^{(0)}\}_{q\in\mathcal{N}^Q}, \lambda_m^{(0)}, \theta_m^{(0)}\}_{m\in\mathcal{N}^M}$
while (1) **do**
  for $m = \{1,2,\cdots,M\}$ **do**
    Calculate the Lagrangian function based on (45).
    Calculate the partial derivatives based on (46).
    Update decision variable $c_{qm}^{(\kappa+1)}$ for $\forall q,m$, based on (47).
    Update Lagrange multipliers $\lambda_m^{(\kappa+1)}$ and $\theta_m^{(\kappa+1)}$ for $\forall m$ based on (48).
  end for
  if $\max\limits_m\{\lambda_m^{(\kappa+1)} - \lambda_m^{(\kappa)}\} < \varepsilon_\lambda \wedge \max\limits_m\{\theta_m^{(\kappa+1)} - \theta_m^{(\kappa)}\} < \varepsilon_\theta$ **then**
    BreakWhile
  end if
end while
return $\{\mathcal{C}\}$

let $\frac{\partial L_4^m(\{c_{qm}\}_{m\in\mathcal{N}^M})}{\partial c_{qm}} = 0$, then the formula (46) can be regarded as the intersection of the linear function $(2\theta_m\epsilon_m)\cdot c_{qm} + \frac{\lambda_m}{x_{qm}\phi_q(1-y_{qm})}$ and the power function c_{qm}^{-2}. Next, let $x^* = H(a,b)$ denote the intersection (or solution) of function $ax + b = x^{-2}$, which implies that the iteration formula can be written as follows.

$$
c_{qm}^{(\kappa+1)} = H(2\theta_m\epsilon_m, \frac{\lambda_m^{(\kappa)}}{x_{qm}\phi_q(1-y_{qm}^{(\kappa)})}) \tag{47}
$$

then, the Lagrange multipliers are calculated as follows.

$$
\lambda_m^{(\kappa+1)} = \lambda_m^{(\kappa)} + \zeta_{m\lambda}^{(\kappa)}\cdot\left(\sum_{q=1}^{Q} c_{qm} - C_0\right) \tag{48}
$$

$$
\theta_m^{(\kappa+1)} = \theta_m^{(\kappa)} + \zeta_{m\theta}^{(\kappa)}\cdot\left(\epsilon_m\sum_{q=1}^{Q} c_{qm}^2 x_{qm}\phi_q(1-y_{qm}) - c_8\right)
$$

Finally, the solution algorithm for decision variables $\{\mathcal{C}\}$ is given in Alg. 4.

4) Solving for Decision Variables $\{\mathcal{Y}\}$: In this subsection, the solution of the decision variable $\mathcal{Y}$ is proposed, when other decision variables are fixed, and the optimization problem P_5^{Sub} is given as follows.

$$
\boldsymbol{P_5^{\text{Sub}}}: \min_{\mathcal{Y}} \sum_{m=1}^{M} T_m^{s1} + T^{s2} + \sum_{m=1}^{M} T_{ms}^{c2} \tag{49a}
$$

$$
s.t. \quad C_5, C_6, C_9 \tag{49b}
$$

where both the objective function and the constraints are linear function with respect to $\mathcal{Y} = \{y_{qm}\}_{q\in\mathcal{N}^Q,m\in\mathcal{N}^M}$. Thus, we can solve the optimization problem P_5^{Sub} using a similar method mentioned above (or using the barrier function interior point method, CVX, and so on).

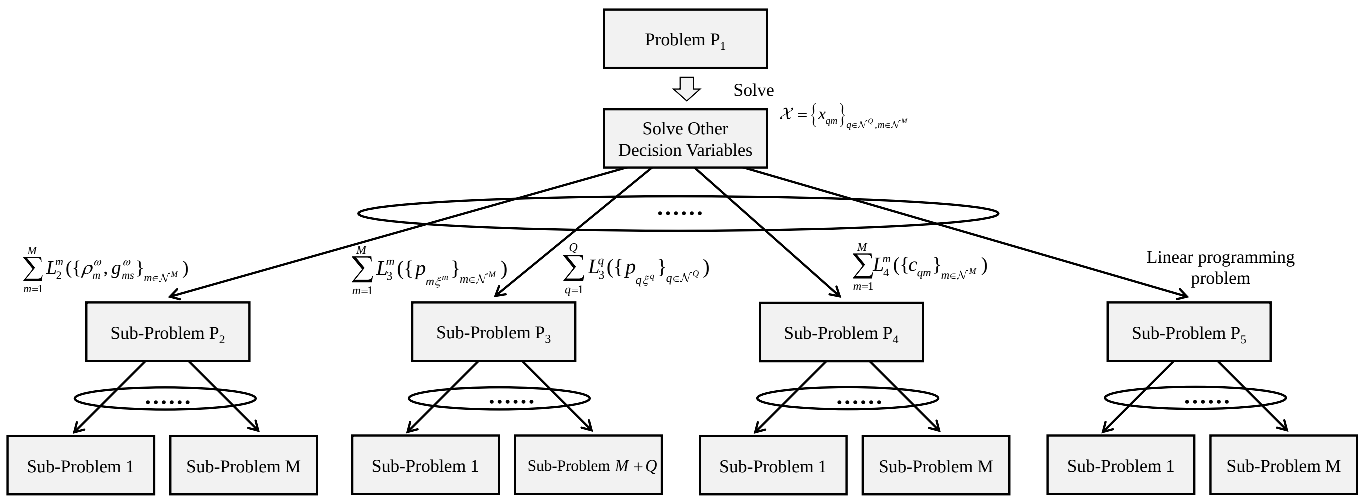

Fig. 7: Detailed decomposition architecture for the optimization problem P_1.

Algorithm 5 Joint Iteration Algorithm (Top Algorithm)

Input: Optimization Problems P_2^{Sub}, P_3^{Sub} and P_4^{Sub}.
Output: $\{\varrho, \mathcal{G}, \mathcal{P}^b, \mathcal{C}, \mathcal{Y}\}$.

1: **Initialize** $\varrho^{(0)}, \mathcal{G}^{(0)}, \mathcal{P}^{b(0)}, \mathcal{C}^{(0)}, \mathcal{Y}^{(0)}, \tau = 0$.
2: **while** (1) **do**
3: Fix the decision variables $\{\mathcal{P}^{b(\tau)}, \mathcal{C}^{(\tau)}, \mathcal{Y}^{(\tau)}\}$, and solve $\{\varrho^{(\tau+1)}, \mathcal{G}^{(\tau+1)}\}$ by Alg. 2.
4: Fix the decision variables $\{\varrho^{(\tau+1)}, \mathcal{G}^{(\tau+1)}, \mathcal{C}^{(\tau)}, \mathcal{Y}^{(\tau)}\}$, and solve $\{\mathcal{P}^{b(\tau+1)}\}$ by Alg. 3.
5: Fix the decision variables $\{\varrho^{(\tau+1)}, \mathcal{G}^{(\tau+1)}, \mathcal{P}^{b(\tau+1)}, \mathcal{Y}^{(\tau)}\}$, and solve $\{\mathcal{C}^{(\tau+1)}\}$ by Alg. 4.
6: Fix the decision variables $\{\varrho^{(\tau+1)}, \mathcal{G}^{(\tau+1)}, \mathcal{P}^{b(\tau+1)}, \mathcal{C}^{(\tau+1)}\}$, and solve $\{\mathcal{Y}^{(\tau+1)}\}$ by barrier function interior point method.
7: **if** $\{\mathcal{P}^{b(\tau+1)}, \mathcal{C}^{(\tau+1)}, \mathcal{Y}^{(\tau+1)}, \varrho^{(\tau+1)}, \mathcal{G}^{(\tau+1)}\}$ converge **then**
8: **BreakWhile**
9: **end if**
10: Update $\tau = \tau + 1$.
11: **end while**
12: **return** $\{\varrho, \mathcal{G}\}$

The computational complexity is analyzed in this paper. In particular, the BCD method, as mentioned in Alg. 5, has a computational complexity of $\mathcal{O}(\tau \cdot C(M,Q,W))$, where τ represents the iteration count for BCD, and $C(M,Q,W)$ denotes the maximum complexity for solving each optimization problem (Alg. 2, Alg. 3, and Alg. 4) when other decision variables are fixed, which implies that $C(M,Q,W) = \max\{C_2(M,W), C_3(M,Q), C_4(M,Q), LP\}$.

5) Outer Iteration Algorithm and Complexity Analysis : It is worth noting that Alg. 2, Alg. 3, and Alg. 4 serve as inner-layer iterative algorithms, each solving one variable while keeping the others fixed. Based on Fig. 3, a joint iterative algorithm based on BCD method for the outer layer is proposed. This algorithm iteratively polls and resolves the different variables in sequence, as detailed in Alg. 5.

Next, we analyze each individual sub-optimization problem. It is worth mentioning that each sub-optimization problem mentioned in this article can be decomposed in the form of $L^m(\cdot)$ or $L^q(\cdot)$, as mentioned in Fig. 7, which implies that distributed parallel computing is feasible, and the algorithm complexity can be effectively reduced. Specifically, for the sub-optimization problem P_2, the computational complexity $C_2(M,W) = k_1' \cdot W$, where M is the number of nodes for parallel computing, and k_1' is the number of iteration index. For the sub-optimization problem P_3, the computational complexity $C_3(M,W) = \max\{C_3(M), C_3(Q)\} = \max\{k_2' \cdot M, k_3' \cdot Q\}$. In addition, $C_4(M,Q) = k_4' \cdot Q$, and LP is the computational complexity of linear programming problem. Thus, based on the decomposition architecture for the optimization problem P_1, the computational complexity can be effectively reduced