Throughput Maximization for Result Multicasting by Admitting Delay-aware Tasks in MEC Networks for High-speed Railways: Some Supplemental Materials

Road map: In section 1, we list the key notations of the paper. In section 2, we give more details of the communication model and give the process of deriving Eq.(1) in the main body of the paper. In section 3, we give the analysis of handovers during the wireless transmission of each train. In section 4, we formulate the defined problem as integer linear programming (ILP). In section 5, we provide the analysis of the procedures and algorithms devised in terms of performance guarantee and time complexity. In section 6, we list the communication parameters used in the simulations.

1 Key Notations

We list the key notations of our paper in Table 1.

2 More Details for the Communication Model

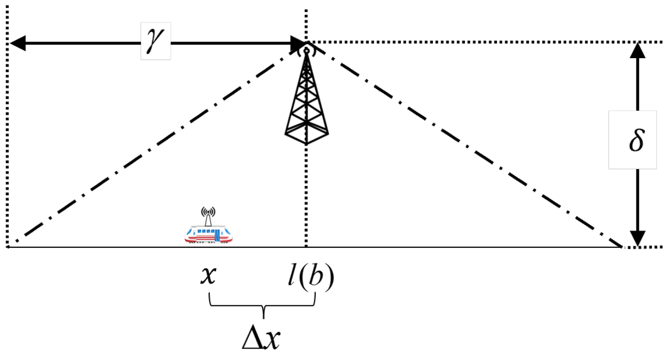

Figure 1: An example of the coverage area of an BS.

Passengers on the train access the MEC network through a roof-mounted access point (AP), which is called the two-hop structure [1]. For simplicity, we regard the point of the vertical projection of the BS on the track in Euclidean space as the location of the BS. The position of a train is considered the position of its AP, and investigating the communication between an AP and the equipment of passengers is beyond the scope of this paper. We assume that all the trains' APs are homogeneous. The APs and BSs are based on orthogonal frequency division multiple access (OFDMA) systems [2]. Specifically, the spectrum within one BS is orthogonally assigned to each passenger, so that the data can be transmitted in parallel. BSs can be coordinated using game- or learning-based algorithms [3, 4] to eliminate interference between them [5].

As shown in Fig. (1), δ is the distance between each BS and the tracks, the wireless coverage radius of each BS along the track is denoted as γ, $l(b)$ is the location of the BS $b \in B$. Denote by $\mathcal{G}_b(x)$ the channel gain between BS b and a train in the communication area of BS b, where "x" represents the location of the train. $\mathcal{G}_b(x)$ can be computed using the channel gain model [3] $\mathcal{G}_b(x)=\left((l(b)-x)^2+\delta^2\right)^{-\frac{\zeta}{2}}$, where ζ is the path loss exponent. We consider the basic fixed channel assignment (FCA) scheme for each train. Let $\mathcal{W}$ and P_b be the wireless bandwidth of each subcarrier and the transmission power of a BS, respectively. Denote by $\mathcal{C}_b^{\downarrow}(x)$ the channel capacity of the downlink of BS b, which can be expressed as $\mathcal{C}_b^{\downarrow}(x) = \mathcal{W}\log_2\left(\frac{P_b\mathcal{G}_b(x)^2}{I\cdot P_b\mathcal{G}_b(x)^2+\sigma^2}+1\right)$, where σ^2 is the noise power, and I is the inter-channel interference (ICI) factor [6] caused by the Doppler effect. I is expressed as $I=1-\int_{-1}^{1}(1-|y|)J_0\left(2\pi f\frac{\nu_h}{c_l}\tau y\right)dy$, where f is the frequency of a subcarrier, ν_h is the velocity of the train, c_l is the speed of light, τ is the duration of the signal symbol, $J_0(\cdot)$ is the Bessel function of the first kind of order zero.

Next, we give details on deducing the formula of $u_b(l(h), l'(h))$.

Table 1: key notations

Notations	Definition	Notations	Definition
$B \subset V$	the set of uniformly deployed homogeneous BSs with routing capabilities	B_H	the BS groups serving train set H
$B_h \in B_H$	the BS group serving train h	$B_h^{\downarrow}$	the group of destination BSs for transmitting result set RE_h to train h
C_H	the admission cost of task set $K_H(t)$	$C_{i,h}$	the admission cost of task $k_{i,h}$
C_v	the computing capacity of cloudlet $v_c \in V_c$	$\mathcal{C}_b^{\downarrow}(x)$	the channel capacity of the downlink of BS b when the train location is x
$c(e)$	the cost of consuming one unit of bandwidth at link $e \in E$	$c(v)$	the cost of consuming one unit of storage capacity at router $v \in V$
$c(v_K)$	the cost of using computing resources in computing cloudlet v_K	$D_h \subseteq B$	the set of destination BSs to which result set R_h is delivered
d_e	the delay on link $e \in E$ for transmitting a unit of data traffic	d_v	the delay on router $v \in V$ for transmitting a unit of data traffic
d_{req}	the identical end-to-end delay requirement of each task	$d_{i,h}^{com}$	the delay that task $k_{i,h}$ computed in computing cloudlet v_K
$d_{i,h}^{rou}$	the routing delay of task $k_{i,h}$ in T_K	$d_{i,h}^{net}$	the network delay of task $k_{i,h}$ and its results
d_H^{net}	the network delay of task set $K_H(t)$	d_H^{total}	the total delay experienced by multicast task set $K_H(t)$ in G
$f_{i,h}$	the demanded CPU cycles for computing task $k_{i,h}$	F_v	the number of CPU cycles per second of each container of cloudlet $v_c \in V_c$
$G = (V, E)$	the MEC network with a set V of routers and a set E of wired links	H	the set of trains in the system, each train is denoted by $h \in H$
H_h^{des}	the set of the destination trains of task set K_h	$H_{i,h}^{des} \subseteq H_h^{des}$	the set of the destination trains to which the computation results of set R_h need to be sent
$K_H(t)$	the task set from train set H at moment t	$K_h \subseteq K_H(t)$	the task set from train h at moment t
$K_H^v(t) \subseteq K_H(t)$	the set of tasks routing on tree T_v	$k_{i,h}$	the ith task from train h, where $k_{i,h} \in K_h$
$l(\cdot)$	the function to obtain the location of a train / BS	$\mathcal{N}_h^{\downarrow}$	the number of handovers for train h in downlinks
p_h	the simple path connects a certain BS $b \in B_h$ with $v_K \in T_K$	$p_{h,h'} = p_h \cup p_{h'}$	the path passing through root vertex v_T and connecting group B_h and group $B_{h'}$ in T_v
R_H	the set of computation results of $K_H(t)$	$R_h \subseteq R_H$	the computation result set associated with task set K_h
$R_h^{rec} \subseteq R_H$	the set of computation results received by train h	$R_H^v \subseteq R_H$	the set of results routing on tree T_v
$r_{i,h,h'} \in R_h$	a copy of result received by each destination train $h' \in H_{i,h}^{des}$	γ	each BS's wireless coverage radius along the track
T_K	the optimal multicast tree which connects computing cloudletv_K with each $B_h \in B_H$	T_v	a general group Steiner tree rooted at any cloudlet $v_T \in V_c$
$u_b(l(\cdot), l(*))$	the data volume transmitted by BS b when train moves from location "$\cdot$" to "$*$"	$\mathcal{U}(b_h^j)$	the data volume that each BS $b_h^j \in B_h^{\downarrow}$ transmits to train h
u_{max}	the maximum volume of data that a train receives from a BS	$V_c \subset V$	the routers with attached cloudlets
v_K	the computing cloudlet for processing task set $K_H(t)$	ν_h	the velocity of train h
$z(\cdot)$	the function to obtain the data volume of a task or result or task set or result set	$\rho_{i,h} \in \mathbb{R}^+$	the ratio between the volumes of task $k_{i,h}$ and result $r_{i,h,h'}$

Proof. Let Δu_b be the data volume that a BS b transmits to a train h, where the initial location of train h is x, and the train travels a distance of Δx in a very short period Δt. Δu_b can be represented as

$$\Delta u_b = \mathcal{C}_b^{\downarrow}(x + \Delta x) \cdot \Delta t \approx \mathcal{C}_b^{\downarrow}(x) \cdot \frac{\Delta x}{\nu_h}, \tag{1}$$

where $l(b) - \gamma \leq \Delta x \leq l(b) + \gamma$.

According to Shannon's theory, channel capacity $\mathcal{C}_b^{\downarrow}(x)$ is a continuous function on parameter x. Thus, the value of $u_b(l(h), l'(h))$ can be regarded as the accumulated sum of Δu_b, i.e.,

$$u_b(l(h), l'(h)) \approx \sum \Delta u_b. \tag{2}$$

According to numerical integration [7], the data volume that a BS transmits to a train from location $l(h)$ to location $l'(h)$ can be considered as the following integration within a given interval $[l(h), l'(h)]$, i.e.,

$$u_b(l(h), l'(h)) = \int_{l(h)}^{l'(h)} \mathcal{C}_b^{\downarrow}(x) \cdot \frac{|l(b) - x|}{\nu_h} dx. \tag{3}$$

□

3 Handover analysis

Similar to our previous study [5], four handover cases are discussed as follows.

Case 1: Train h finishes downloading the data of the computation results before h leaves the coverage area of BS b, and $z(R_h^{rec}) < u_{max}$, i.e., $z(R_h^{rec}) \leq u_b(l(h), l(b) + \gamma)$. In this case, no handover occurs and $\mathcal{N}_h^{\downarrow}$ equals 0.

Case 2: Train h cannot finish downloading the data of the computation results before h leaves the coverage area of BS b, and $z(R_h^{rec}) \leq u_{max}$, i.e., $u_b(l(h), l(b) + \gamma) < z(R_h^{rec}) \leq u_{max}$. The remaining data from the computation results will be transmitted to the adjacent BS of BS b. In this case, the handover occurs only once. Thus, $\mathcal{N}_h^{\downarrow}$ equals 1.

Case 3: Data volume of computation results $z(R_h^{rec}) > u_{max}$, so computation results can be split into $\lfloor z(R_h^{rec})/u_{max} \rfloor$ portions. Different portions need to be transmitted to different BSs, and thus the number of BSs is $\lfloor z(R_h^{rec})/u_{max} \rfloor$. If train h can download volume $z(R_h^{rec}) - \lfloor z(R_h^{rec})/u_{max} \rfloor \cdot u_{max}$ of data of computation results from BS b, before train h leaves the coverage area of BS b, i.e., $z(R_h^{rec}) - \lfloor z(R_h^{rec})/u_{max} \rfloor \cdot u_{max} \leq u_b\left(l(h), l(b) + \gamma\right)$. Then $\mathcal{N}_h^{\downarrow}$ equals $\lfloor z(R_h^{rec})/u_{max} \rfloor$.

Case 4: Data volume of computation results $z(R_h^{rec}) > u_{max}$, and train h cannot download $z(R_h^{rec}) - \lfloor z(R_h^{rec})/u_{max} \rfloor \cdot u_{max}$ of data of computation results from BS b, before train h leaves the coverage area of BS b, i.e., $z(R_h^{rec}) - \lfloor z(R_h^{rec})/u_{max} \rfloor \cdot u_{max} > u_b\left(l(h), l(b) + \gamma\right)$. Then $\mathcal{N}_h^{\downarrow}$ equals $\lfloor z(R_h^{rec})/u_{max} \rfloor$.

In summary, $\mathcal{N}_h^{\downarrow}$ can be expressed as

$$\mathcal{N}_h^{\downarrow} = \begin{cases} \lfloor z(R_h^{rec})/u_{max} \rfloor & \text{Case 1 or Case 3,} \\ \lceil z(R_h^{rec})/u_{max} \rceil & \text{Case 2 or Case 4.} \end{cases} \tag{4}$$

4 An ILP Formulation

For the defined problem, we start with the classic ILP formulation. For ease of description, we transform undirected graph $G = (V, E)$ into directed graph $G_d = (V_d, E_d)$. Specifically, for each node $v \in V$, add node v into V_d, for each link $e \in E$, directed edge $\langle u, v \rangle$ and directed edge $\langle v, u \rangle$ are added into E_d, where $u \in V$ and $v \in V$. The weights of edge $\langle u, v \rangle$ and edge $\langle v, u \rangle$ are equal to the weight of link $e \in E$, i.e., $V_d = \{v | \ v \in V\}$, $E_d = \{\langle u, v \rangle \mid u, v \in V\}$. For brevity, we define a sign function $sgn(x)$ as follows.

$$\mathrm{sgn}(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{5}$$

The ILP includes the following decision variables.

$\mathbf{n}_{i,h,v}$ is a decision variable of value 1 if cloudlet task $k_{i,h}$ is computed in cloudlet $v \in V_c$, and value 0 otherwise.

$\mathbf{n}_v$ is a decision variable of value 1 if cloudlet $v \in V_c$ is the computing cloudlet and value 0 otherwise.

$\mathbf{x}_{i,h}$ is 1 if task $k_{i,h}$ is admitted or 0 otherwise.

$\mathbf{y}_{i,h,h'}$ is 1 if result $r_{i,h,h'}$ is multicasted or 0 otherwise.

$\mathbf{m}_v$ is a decision variable that has value 1 if node v carries the traffic of any task or result, otherwise, the value is 0.

$\mathbf{m}_{\langle u,v \rangle}$ is a decision variable that has value 1 is edge $\langle u, v \rangle$ carries the traffic of any task or result, otherwise, the value is 0.

According to the problem we have defined, the objective function of the ILP is as follows:

$$maximize \quad \sum_{h=1}^{|H|} \sum_{i=1}^{|K_h|} \sum_{h' \in H_{i,h}^{des}} \mathbf{y}_{i,h,h'}. \tag{6}$$

Constraints (7), (8) and (9) ensure that (i) each task is computed in the same cloudlet; (ii) there exists only one computing cloudlet in G for computing task set $K_H(t)$; (iii) node $v \in V$ can compute a task if and only if it is a cloudlet,

$$\sum_{v \in V_c} \mathbf{n}_{i,h,v} = \mathbf{x}_{i,h},\ \ i \in K_h,\ \ h \in H. \tag{7}$$

$$\sum_{v \in V_c} \mathbf{n}_v = \operatorname{sgn}\left(\sum_{v \in V_c} \mathbf{n}_{i,h,v}\right),\ \ i \in K_h,\ \ h \in H. \tag{8}$$

$$\mathbf{n}_{i,h,v} = \mathbf{n}_v = 0,\ \ \forall v \in V_d \setminus V_c. \tag{9}$$

Constraint (10) enforces the capacity constraint for each cloudlet $v_c \in V$.

$$\sum_{h=1}^{|H|} \sum_{i=1}^{|K_h|} \mathbf{x}_{i,h} \le C_v,\ \ \forall v \in V_d,\ \ \mathbf{n}_v = 1. \tag{10}$$

Constraint (11) is standard linear programming of a GST [8], where $\delta(V')$ denotes the set of edges that possess exactly one endpoint in V'. All data traffic of the tasks and results will be routed on the obtained GST.

$$\sum_{\langle u,v\rangle \in \delta(V')} \mathbf{m}_{\langle u,v\rangle} \ge 1,\ \ \forall V' \subseteq V, \text{ such that } v_K \in V' \text{ and } Q \cap B_h = \emptyset \text{ for some } h \in H. \tag{11}$$

Constraint (12) enforces that the total cost to route the tasks and the results on the obtained GST will not exceed the budget.

$$\sum_{h=1}^{|H|} \sum_{i=1}^{|K_h|} \left(\begin{array}{l} z(k_{i,h})\mathbf{x}_{i,h} \cdot \left(\sum\limits_{\langle u,v\rangle \in E_d} c\left(\langle u,v\rangle\right) \cdot \mathbf{m}_{u,v} + \sum\limits_{v \in V_d} c(v) \cdot \mathbf{m}_v \right) + c(v) z(k_{i,h}) \mathbf{x}_{i,h} \\ + \sum\limits_{h' \in H_{i,h}^{des}} \left(z(r_{i,h,h'}) \mathbf{y}_{i,h,h'} \cdot \sum\limits_{\langle u,v\rangle \in E_d} c\left(\langle u,v\rangle\right) \cdot \mathbf{m}_{u,v} \right) \end{array} \right) \le \beta,\ \forall u, v \in V_d\ . \tag{12}$$

Constraint (13) enforces that the total delay experienced by an admitted task and its corresponding results on the obtained GST can meet the delay requirement of the task.

$$\left(\sum_{v \ne u} \mathbf{m}_{\langle u,v\rangle} \cdot d_e + \sum_{v \in V_d} \mathbf{m}_v \cdot d_e \right) \cdot \mathbf{x}_{i,h} + \max_{h' \in H_{i,h}^{des}} \left\{ \left(\sum_{v \ne u} \mathbf{m}_{\langle u,v\rangle} \cdot d_e + \sum_{v \in V_d} \mathbf{m}_v \cdot d_e \right) \cdot \mathbf{y}_{i,h,h'} \right\} + \sum_{v \in V_c} d_{i,h}^{com} \mathbf{n}_v \le d_{req},\ \ i \in K_h,\ \ h \in H. \tag{13}$$

Inspired by the network flow model [9] for multicast routing in the directed Steiner tree, we propose our flow model (Constraints (14)-(22)) to capture traffic changes in the directed GST. We call that a node *consumes* a task/result if the node takes out one task/result from the passing data flow. Clearly, such a node can be computing cloudlet or a BS in the BS group, while the other nodes do not consume any task or result. Specifically, let $R_{u,v}$ and $S_{u,v}$ be the aggregate number of results and tasks that go from vertex u to v, respectively. Constraint (14) restricts the range of $R_{u,v}$ and $S_{u,v}$

$$S_{u,v}, R_{u,v} \ge 0,\ \ \forall u, v \in V_d. \tag{14}$$

Constraint (15) enforces that the number of results routing on the path of the obtained GST is less than the number of results multicast by the root.

$$\sum_{u \ne v} \mathbf{m}_{\langle v,u\rangle} \cdot R_{v,u} \le \sum_{h=1}^{|H|} \sum_{i=1}^{|R_h|} \sum_{h' \in H_{i,h}^{des}} \mathbf{y}_{i,h,h'},\ \ \forall u, v \in V_d,\ \ \mathbf{n}_v = 1. \tag{15}$$

Constraint (16) enforces that no result can return to the root.

$$\sum_{u \in V_d, u \ne v} R_{u,v} = 0,\ \ \mathbf{n}_v = 1. \tag{16}$$

Constraints (17) and (18) handle four cases of the results consumed by the groups, where node v will not consume any result if (i) $v \notin B_H$ or (ii) v is in a certain BS group and v connects other nodes of the same group; otherwise, node v will consume $\sum_{h=1}^{|H|} \sum_{i=1}^{|R_h|} \mathbf{y}_{i,h,h'}$ amount of results if (iii) v is in a certain BS group and v connects a node $u \notin B_H$ or (iv) v is in a certain BS group and v connects a node $u \in B_H$ belonging to another BS group.

$$\sum_{v \ne u} \mathbf{m}_{\langle u,v\rangle} \cdot R_{u,v} - \sum_{v \ne w} \mathbf{m}_{\langle v,w\rangle} \cdot R_{v,w} = 0,\ \ \begin{cases} \text{Case 1:} \forall \mathbf{n}_v, \mathbf{n}_u, \mathbf{n}_w \ne 1,\ \ \forall v \notin B_H, \\ \text{Case 2:} \forall \mathbf{n}_v, \mathbf{n}_u, \mathbf{n}_w \ne 1,\ \ v \in B_{h'},\ \ \forall u, w \in B_{h'},\ \ h' \in H_h^{des}. \end{cases} \tag{17}$$

$$\sum_{v\neq u}\mathbf{m}_{\langle u,v\rangle}\cdot R_{u,v}-\sum_{v\neq w}\mathbf{m}_{\langle v,w\rangle}\cdot R_{v,w}=\sum_{h=1}^{|H|}\sum_{i=1}^{|R_h|}\mathbf{y}_{i,h,h'},\ \begin{cases}\text{Case 3:}\,\forall\mathbf{n}_v,\mathbf{n}_u,\mathbf{n}_w\neq 1,\ v\in B_{h'},\ h'\in H_h^{des},\ \exists u\notin B_H,\\ \text{Case 4:}\,\forall\mathbf{n}_v,\mathbf{n}_u,\mathbf{n}_w\neq 1,\ v\in B_{h'},\ \forall u\in B_{h''},\ h',h''\in H_h^{des}.\end{cases} \tag{18}$$

Constraint (19) ensures that no unprocessed task can leave the root of GST.

$$\sum_{u\neq v}S_{v,u}=0,\ \ \forall u,v\in V_d,\ \mathbf{n}_v=1. \tag{19}$$

Constraint (20) and constraint (21) ensure that any node will not consume the task except that the node is the root.

$$\sum_{v\neq u}\mathbf{m}_{\langle u,v\rangle}\cdot S_{u,v}-\sum_{v\neq w}\mathbf{m}_{\langle v,w\rangle}\cdot S_{v,w}=0,\ \ \forall u,v,w\in V_d,\ \forall\mathbf{n}_v,\ \mathbf{n}_u,\mathbf{n}_w\neq 1. \tag{20}$$

$$\sum_{v\neq u}\mathbf{m}_{\langle u,v\rangle}\cdot S_{u,v}-\sum_{v\neq w}\mathbf{m}_{\langle v,w\rangle}\cdot S_{v,w}=\sum_{h=1}^{|H|}\sum_{i=1}^{|K_h|}\mathbf{x}_{i,h},\ \ \forall u,v,w\in V_d,\ \mathbf{n}_v=1,\ \forall\mathbf{n}_u,\mathbf{n}_w\neq 1. \tag{21}$$

Constraint (22) ensures that $S_{u,v}>0$ only occurs when an edge between vertex u and vertex v is included in the GST.

$$\left(\sum_{h=1}^{|H|}\sum_{i=1}^{|K_h|}\mathbf{x}_{i,h}\right)\cdot\sum_{\langle u,v\rangle\in E}\mathbf{m}_{\langle u,v\rangle}\geq S_{u,v},\ \ \forall u,v\in V_d. \tag{22}$$

Constraints (23), (24), (25), (26), (27) restrict the ranges of decision variables to 0 and 1.

$$\mathbf{n}_{i,h,v},\mathbf{n}_v\in\{0,1\},\ \ \forall v\in V_c,\ \forall i\in K_h,\ \forall h\in H. \tag{23}$$

$$\mathbf{x}_{i,h}\in\{0,1\},\ \ \forall i\in K_h,\ \forall h\in H. \tag{24}$$

$$\mathbf{y}_{i,h,h'}\in\{0,1\},\ \ \forall i\in K_h,\ \forall h\in H,\ h'\in H_{i,h}^{des}. \tag{25}$$

$$\mathbf{m}_{\langle u,v\rangle}\in\{0,1\},\ \ \forall u,v\in V_d. \tag{26}$$

$$\mathbf{m}_v\in\{0,1\},\ \ \forall v\in V_d. \tag{27}$$

Constraints (28) and (29) describe the priority relationships between the tasks and their results, i.e., the number of the results is greater than those of the corresponding tasks, and a result can be multicast if and only if its corresponding task has been admitted.

$$\mathbf{x}_{i,h}\geq\mathbf{y}_{i,h,h'},\ \ \forall i\in K_h,\ \forall h,h'\in H. \tag{28}$$

$$\mathbf{x}_{i,h}\leq\sum_{h'\in H_h^{des}}\mathbf{y}_{i,h,h'}\leq\mathbf{x}_{i,h}\cdot|R_{i,h}|,\ \ \forall i\in K_h,\ \forall h,h'\in H. \tag{29}$$

5 Algorithm Analysis

Corollary 1. *If a GST can be successfully constructed or adjusted, then the delay requirements of the tasks routing on this GST can be met.*

Theorem 1. *Procedure* 2 *has a relative performance guarantee of* 1/2.

Proof. First, let $q_{\max}$ be a feasible solution of the ILP, where we admit the task with the most results and the results are also multicast, i.e.,

$$q_{\max}=\max_{i,h}\sum_{h'\in H_{i,h}^{des}}\mathbf{y}_{i,h,h'}.$$

Then, denote p^{LP} as the optimal solution of linear relaxation of **P1**, where p^{LP} is an upper bound of the optimal solution. Furthermore, denote by p^G the objective function value obtained by Procedure 2. p^G takes the best solution among p^{LP} and $q_{\max}$, and p^G can be expressed as follows

$$p^G=\max\left\{p^{LP},q_{\max}\right\}.$$

Considering the optimal solution of **P1**, which is denoted as p^*, it can be seen that the following inequality for p^* holds:

$$p^*<p^{LP}+q_{\max}.$$

Let $\rho = \frac{p^G}{p^*}$ be the approximation ratio and we can get with

$$\rho = \frac{p^G}{p^*} = \frac{\max\left\{p^{LP}, q_{\max}\right\}}{p^*} > \frac{\max\left\{p^{LP}, q_{\max}\right\}}{p^{LP} + q_{\max}}.$$

Clearly, $p^{LP} + q_{\max} \leq 2 \cdot \max\left\{p^{LP}, q_{\max}\right\}$, thus $\rho > \frac{\max\left\{p^{LP}, q_{\max}\right\}}{2 \cdot \max\left\{p^{LP}, q_{\max}\right\}} = \frac{1}{2}$.

□

Lemma 1. *Given a network $G = (V, E)$, a group Steiner tree T_v rooted at v_T in G, an auxiliary graph $G_v = (V_v, E_v)$ with respect to G and v_T, delay constraint $\widetilde{d_H^{net}}$, BS group $B_H = \{B_1, ..., B_h, ..., B_{|H|}\}$, Task set $K_H(t)$, and Result set R_H, Procedure 1 takes $O\left(|H|^2 \cdot \left(|E|^2 log^4|E| + |K_H(t)||H|(|V| + |E|) + |K_H(t)|log|K_H(t)| + |H|log|H|\right)\right)$ time in the worst case for (i) adjusting tree T_v, such that $d(T_v) \leq \widetilde{d_H^{net}}$, (ii) obtaining task set $K_H^v(t)$ and result set R_H^v that routing on tree T_v.*

Proof. First, it performs the initialization in $O(1)$ time (Lines 1-2). Then it processes every $h \in H$ as follows (Lines 3-9). It conducts a for loop with $O(|H|)$ iterations (Line 3). In each iteration, it finds path p_h in $O(|V| + |E|)$ time (Line 4). After that, it constructs path set P_H using a while loop with $|H|$ iterations (Lines 5-8) as follows. It finds path $p_{h'}$ in $O(|V| + |E|)$ time and constructs path $p_{h,h'}$ and merges $p_{h,h'}$ into set P_H in $O(1)$ time. Thus, it tasks $O\left(|V| + |E|\right)$ time to do the operations of Line 6. Then, in Line 7, it finds task set K_p in $O(|K_H(t)|)$ time. The calculation of the routing delay $d_{i,h}^p$ for each task $k_{i,h} \in K_p$ takes $O((|V| + |E|))$ time, thus the total calculations of the routing delays for task set K_p take $O\left((|V| + |E|) \cdot |K_H(t)|\right)$ time. After that, the sorting operations tasks $O(|K_H(t)| \cdot log|K_H(t)|)$ time. Calculating the routing delay $d_{i,h}^p$ takes $O(1)$ time. It tasks $O\left(log|H|\right)$ time to insert $p_{h,h'}$ into max-heap Q_{delay}. Thus, it tasks $O\left((|V| + |E| + log|K_H(t)|) \cdot |K_H(t)| + \log|H|\right)$ to perform the operations of Line 7.

Next, it processes the paths in P_H with a while loop with $|P_H|$ iterations (Line 10) as follows (Lines 11-31). In each iteration, It pops out path $p_{h,h'}$ from max-heap Q (Line 11) in $O\left(log|H|\right)$ time, where $p_{h,h'}$ has the maximum delay among all paths in P_H. Especially, it tasks $O(1)$ time to do the calculations of $d(p_{h,h'})$ since we have calculated all the delays of tasks that pass through path $p_{h,h'}$ in Line 7. Thus, the operations of Lines 12-13 take $O(1)$ time. In the **BEST** case, Procedure 1 will stop at Line 13. Otherwise, it will do the operations of Lines 14-26. It takes $O(|E|^2 \cdot log^4|E|)$ to find delay-constrained-least-cost (DCLC) path $p_{h,h'}^{dclc}$ by using Juttner's algorithm [10] (Line 15). After that, it takes $O(|V| + |E|)$ time to obtain the adjusted path $p_{h,h'}^{adj}$ (Lines 16-17), and the feasibility examination for the adjusted path takes $O\left(|H|^2 \cdot |K_H(t)| + |H| \cdot |K_H(t)|\right)$ time (Line 18), which has been proved in Lemma 2. After the examination, the path replacement takes $O(|V| + |E|)$ time and the merging operation of J_h and $J_{h'}$ takes $O(1)$ time (Line 21). However, if the above conditions (Line 16 and Line 20) cannot be met, then Procedure 1 is in the **WORST** case (Line 23), and it processes the adjusted paths at Line 24. The operations of Line 24 take $O(|K_H(t)|)$ time to search for a task whose delay is greater than $\widetilde{d_H^{net}}$. Then, the corresponding results that cause the greater delay are rejected. Then, it takes $O\left(|H| \cdot \log|H|\right)$ time to remove elements from heap Q_{delay} and insert new elements into Q_{delay} (Lines 27-30).

Thus, the total time complexity of Procedure 1 in the worst case is
$O\left(|H|^2 \cdot \left(|E|^2 log^4|E| + |K_H(t)||H|(|V| + |E|) + |K_H(t)|log|K_H(t)| + |H|log|H|\right)\right)$.

□

Lemma 2. *Given an adjusted path p_h^{adj}, delay constraint $\widetilde{d_H^{net}}$, historical record set J_h, there exists a procedure of Line 19, Procedure 1 for examining path p_h^{adj} is feasible or not, which takes $O\left(|H|^2 \cdot |K_H(t)| + |H| \cdot |K_H(t)|\right)$ time.*

Proof. First, it conducts a for loop with $O(|J_h|)$ iterations. The value of $|J_h|$ is equal to that of $|H|^2$. In each iteration, it calculates the routing delay of a path with function $d(\cdot)$ at a cost of $O\left((|V| + |E|) \cdot |K_H(t)|\right)$. The reason is that the calculations here are the same as that of Line 7, Procedure 1. Thus, the total time complexity of Line 18, Procedure 1 is $O\left(|H|^2 \cdot (|V| + |E|) \cdot |K_H(t)|\right)$. □

Lemma 3. *Given a bipartite graph $G_b = (V_b, U_b, E_b)$, there exists a procedure Procedure 2 for determining the the sets of admitted tasks and multicast results, which takes $O(|R_H|)$ time.*

Proof. The initialization of variables tasks $O(1)$ time (Lines 1-2). Then, it conducts a for loop with $|U_b|$ iterations. The value of $|U_b|$ is equal to that of $|R_H|$. In each iteration, the operations take $O(1)$ time (Lines 4-25). Thus, the total time complexity of Procedure 2 is $O(|R_H|)$. □

Theorem 2. *Given a train set H with the initial locations and velocities of the trains, a MEC network $G = (V, E)$ with a set V of routers, a subset B of BSs, a subset $V_c \subseteq V$ of cloudlets, and a set E of links, task set $K_H(t)$ and their multicast-oriented computation results with delay requirements and resource demands, there is an algorithm* `HeuAlg` *for the network throughput maximization problem, which takes*
$O\begin{pmatrix} |H|^3|V|\left(|K_H(t)|\left(|V| + |E|\right) + \log|H|\right) + |H|^2|V|\left(|K_H(t)|\log|K_H(t)| + |E|^2\log^4|E|\right) \\ +|V|\left(|H||V|\log|V| + |R_H|\log|R_H|\right) + |R_H|\log(1/\zeta^3) + 1/\zeta^4 + |B| \end{pmatrix}$*, where ζ is a constant with $\zeta > 0$.*

Proof. First, it takes $O(1)$ time to initialize the variables (Lines 1-2). Then, it takes $O(|B|)$ time to create the BS groups, and the rejection of results and tasks takes $O(|R_H| \cdot log(1/\zeta^3) + 1/\zeta^4)$ time using Lawler's fast approximation algorithm [11] (Line 3), where $\zeta > 0$ is the accuracy for the 0-1 knapsack problem.

Then, it conducts a for loop with $O(|V|)$ iterations (Line 4), and processes each cloudlet $v_c \in V_c$ as follows (Lines 5-6). In each iteration, it constructs the auxiliary graph in $O(|V| + |E|)$ time and finds the GST on the auxiliary graph in $O(|H| \cdot (|E| + |V|log|V| + |V| + log|H|))$ time by using Sun's $|H|$-approximation algorithm [12] (Line 5). The operation of inserting the obtained GST into queue Q takes $O(1)$ time (Line 6).

Then, it processes GSTs in queue Q using a while loop with $|V|$ iterations (Lines 8-23). It pops out the first GST in the queue (Line 9). Calculating the delay of a GST takes $O\left((|V| + |E|) \cdot |K_H(t)|\right)$ time (Line 10). It employs Procedure 1 to adjust the GST in $O\left(|H|^2 \cdot \left(|E|^2 log^4|E| + |K_H(t)||H|(|V| + |E|) + |K_H(t)|log|K_H(t)| + |H|log|H|\right)\right)$ time (Line 11). Set $K_H(t)$ and R_H can be obtained in $O(1)$ time (Line 13). The sorting operations of Line 15 and Line 16 take $O(|K_H(t)|log|K_H(t)|) + O(|R_H|log|R_H|)$ time. The construction of the bipartite graph tasks $O(|K_H(t)| + |R_H|)$ time (Line 17). Then, it employs Procedure 2 to admit the tasks and multicast the results in $O(|R_H|)$ time (Line 18). The operations to obtain GST with the maximum throughput and return the feasible task set and the result set take $O(1)$ time (Lines 19-22).

Thus, the total time complexity of HeuAlg is

$$O\begin{pmatrix} |H|^3|V|\left(|K_H(t)|\left(|V| + |E|\right) + \log|H|\right) + |H|^2|V|\left(|K_H(t)|\log|K_H(t)| + |E|^2\log^4|E|\right) \\ +|V|\left(|H||V|\log|V| + |R_H|\log|R_H|\right) + |R_H|\log(1/\zeta^3) + 1/\zeta^4 + |B| \end{pmatrix}.$$

6 Wireless Communication Parameters

The wireless communication parameters are listed in Table 2, which are set according to [3, 5, 13–15].

Table 2: Wireless Communication Parameters

Parameters	Value
Path Loss Exponent ζ [3]	3
Length of Train [5]	200m
Distance between BS and Railways [5]	100m
BS Antenna Height [14]	32m
AP Antenna Height [14]	1.5m
Coverage Radius of a BS	1000m
Handover Time [13]	100ms
Number of Subcarriers [14]	1024
Bandwidth of Subcarrier W_b [14]	15kHz
Carrier Frequency f [13]	2Ghz
Symbol Duration τ [13]	1/14 ms
BS Transmit Power P_b [15]	53dBm
Power Density of Background Noise σ [15]	-145dBm/Hz

□

References

[1] M. Gao, B. Ai, Y. Niu, W. Wu, P. Yang, F. Lyu, and X. Shen, "Efficient hybrid beamforming with anti-blockage design for high-speed railway communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9643–9655, 2020.

[2] S. Yu, B. Dab, Z. Movahedi, R. Langar, and L. Wang, "A socially-aware hybrid computation offloading framework for multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1247–1259, 2019.

[3] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[4] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, and Y. Zhang, "Reinforcement learning-based mobile offloading for edge computing against jamming and interference," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6114–6126, 2020.

[5] J. Xu, Z. Wei, Z. Lyu, L. Shi, and J. Han, "Throughput maximization of offloading tasks in multi-access edge computing networks for high-speed railways," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9525–9539, 2021.

[6] Y. Li and L. J. Cimini, "Bounds on the interchannel interference of ofdm in time-varying impairments," *IEEE Transactions on Communications*, vol. 49, no. 3, pp. 401–404, 2001.

[7] P. J. Davis and P. Rabinowitz, *Methods of numerical integration.* Academic Press, Inc., 1984.

[8] N. Garg, G. Konjevod, and R. Ravi, "A polylogarithmic approximation algorithm for the group steiner tree problem," *Journal of Algorithms*, vol. 37, no. 1, pp. 66–84, 2000.

[9] T. Liu and W. Liao, "Multicast routing in multi-radio multi-channel wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 10, pp. 3031–3039, 2010.

[10] A. Juttner, B. Szviatovski, I. Mécs, and Z. Rajkó, "Lagrange relaxation based method for the qos routing problem," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 859–868, IEEE, 2001.

[11] E. L. Lawler, "Fast approximation algorithms for knapsack problems," *Mathematics of Operations Research*, vol. 4, no. 4, pp. 339–356, 1979.

[12] Y. Sun, X. Xiao, B. Cui, S. Halgamuge, T. Lappas, and J. Luo, "Finding group steiner trees in graphs with both vertex and edge weights," *Proceedings of the VLDB Endowment*, vol. 14, no. 7, pp. 1137–1149, 2021.

[13] L. Tian, J. Li, Y. Huang, J. Shi, and J. Zhou, "Seamless dual-link handover scheme in broadband wireless communication systems for high-speed rail," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 4, pp. 708–718, 2012.

[14] C. Zhang, P. Fan, K. Xiong, and P. Fan, "Optimal power allocation with delay constraint for signal transmission from a moving train to base stations in high-speed railway scenarios," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5775–5788, 2015.

[15] L. Wang, B. Ai, Y. Niu, X. Chen, and P. Hui, "Energy-efficient power control of train–ground mmwave communication for high-speed trains," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7704–7714, 2019.