Contents

1. Appendix A: Proof of Proposition 2

Proof. When $x_{ij} = 0$, Equation (25) can be expressed as follows:

$$\left.\begin{array}{l} \mu_{ij} \leq 0 \\ -1 \leq 0 \\ \mu_{ij} \geq 0 \end{array}\right\} \Rightarrow \mu_{ij}\left(x_{ij} - 1\right) = 0, \forall (i,j) \in A$$

When $x_{ij} = 1$, Equation (25) can be expressed as follows:

$$\left.\begin{array}{l} \mu_{ij} \leq M \\ x_{ij} - 1 = 0 \\ \mu_{ij} \geq 0 \end{array}\right\} \Rightarrow \mu_{ij}\left(x_{ij} - 1\right) = 0, \forall (i,j) \in A$$

When $x_{ij} = 0$, Equation (26) can be expressed as follows:

$$\left.\begin{array}{l} \xi_{ij} \leq M \\ \xi_{ij} \geq 0 \end{array}\right\} \Rightarrow \xi_{ij} x_{ij} = 0, \forall (i,j) \in A$$

When $x_{ij} = 1$, Equation (26) can be expressed as follows:

$$\left.\begin{array}{l} \xi_{ij} \leq 0 \\ \xi_{ij} \geq 0 \end{array}\right\} \Rightarrow \xi_{ij} x_{ij} = 0, \forall (i,j) \in A$$

This completes the proof. $\square$

2. Appendix B: Proof of Proposition 3

Proof. By temporarily ignoring constraint 38e, the lower level can be viewed as a minimum-cost maximum-flow problem pertaining to the determination of distinct paths without acceptable travel time. Let ρ be the distinct path with the longest travel time in the distinct path set. Define $\hat{\Upsilon}\left(n_{od}(s,t)-1\right)=\Upsilon\left(n_{od}(s,t)\right)-\rho$. Note that $\hat{\Upsilon}\left(n_{od}(s,t)-1\right)$ comprises $n_{od}(s,t)-1$ distinct paths, making it a feasible solution for the model in equations 38f-38h when $n_{od}(s,t) := n_{od}(s,t)-1$. Since this model is a minimization problem, we have:

$$\hat{\Upsilon}\left(n_{od}(s,t)-1\right) \geq \Upsilon\left(n_{od}(s,t)-1\right) \tag{1}$$

Due to the boundedness of $\sum_{(i,j)\in A}(r_{ij}(t)+(1-\beta_{ij}y_{ij})r_{ij}(s))x^*_{ij}$ by definition, the travel time of each path in this path set is also bounded. Moreover, as ρ represents the path with the longest travel time among the path set, it must be greater than or equal to the average travel time of the other $n_{od}(s,t)-1$ paths. Therefore, we can express this relationship as follows:

$$\rho \geq \frac{1}{n_{od}(s,t)-1}\hat{\Upsilon}\left(n_{od}(s,t)-1\right) \tag{2}$$

By combining equations (1) and (2), we have:

$$\begin{aligned}
\frac{1}{n_{od}(s,t)}\Upsilon\left(n_{od}(s,t)\right) &= \frac{1}{n_{od}(s,t)}\left(\hat{\Upsilon}\left(n_{od}(s,t)-1\right)+\rho\right) \\
&\geq \frac{1}{n_{od}(s,t)}\left(\hat{\Upsilon}\left(n_{od}(s,t)-1\right)+\frac{1}{n_{od}(s,t)-1}\hat{\Upsilon}\left(n_{od}(s,t)-1\right)\right) \\
&= \frac{1}{n_{od}(s,t)-1}\hat{\Upsilon}\left(n_{od}(s,t)-1\right) \geq \frac{1}{n_{od}(s,t)-1}\Upsilon\left(n_{od}(s,t)-1\right)
\end{aligned} \tag{3}$$

This completes the proof. □

3. Appendix C: The fitted regression equations

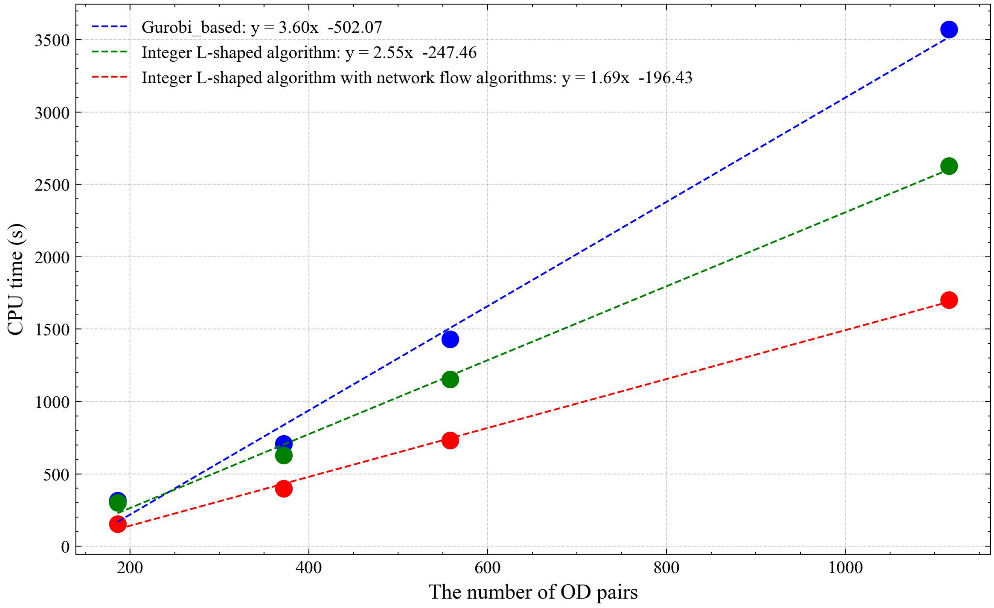

Figure 1: The fitted regression equations for the three methods with increasing OD pair numbers.

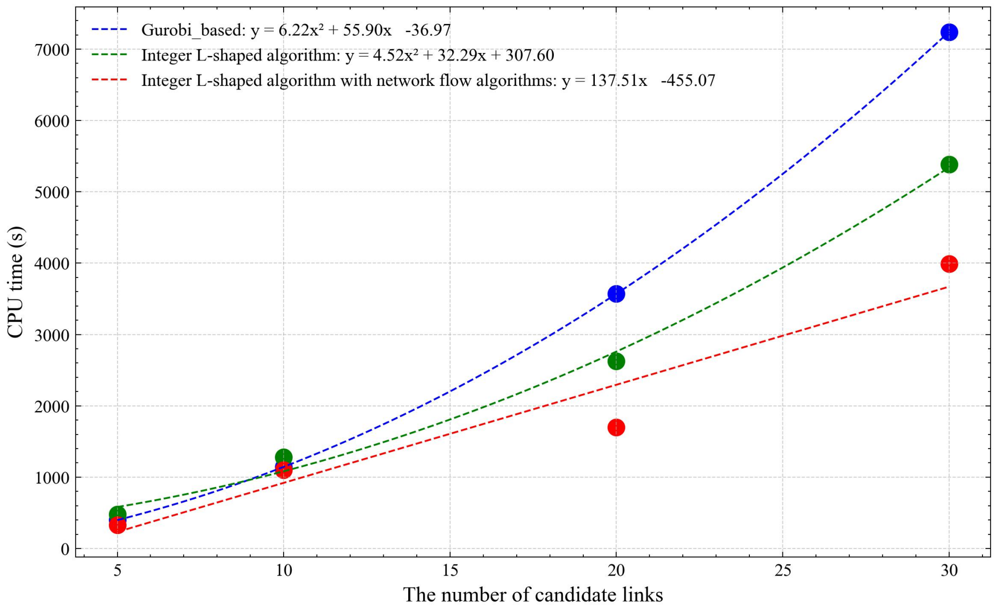

Figure 2: The fitted regression equations for the three methods with increasing candidate link numbers.

4. Appendix D: The logical framework of the methods

Section 4 presents methods for solving the model. Section 4.1 restructures the network to demonstrate optimal integer solutions for decision variables **x**, using KKT conditions to convert the bi-level model into a single-level formulation. Section 4.2 introduces a deterministic equivalent program solvable by commercial solvers like Gurobi and CPLEX. In Section 4.3, we propose an integer L-shaped algorithm with both continuous and integer optimality cuts, along with a polynomial-time algorithm using network flow techniques. In summary, Figure 3 illustrates the logical framework of the methods.

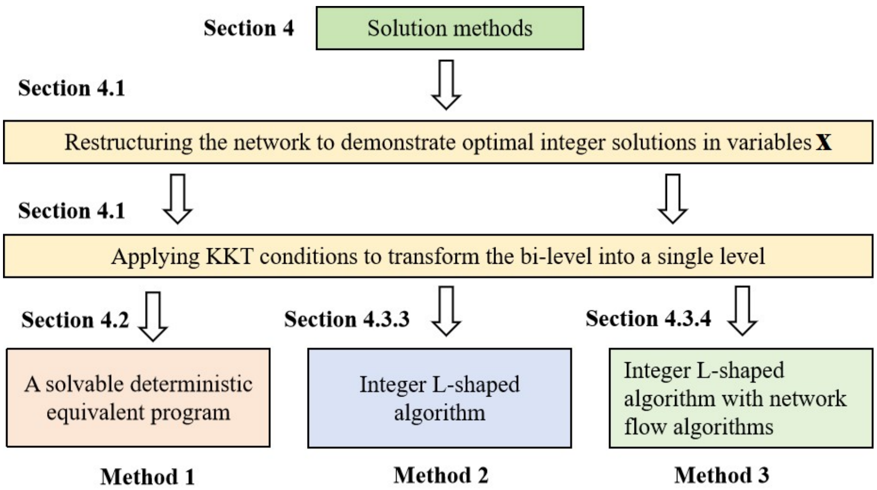

Figure 3: Logical framework of the proposed methods.

5. Appendix E: The flowchart for the algorithmic procedure

Section 4 proposes the integer L-shaped algorithm with continuous and integer optimality cuts, which can effectively solve each *od* pair in each scenario (s, t) separately. To obtain the continuous optimality cut, we relax the lower level to a linear programming problem and derive the cut using duality theory. To obtain the integer optimality cut, instead of directly solving the mixed integer linear programming problem, we propose **Algorithm 2**, which repeatedly employs polynomial-time network flow algorithms to solve each *od* pair in each scenario. In summary, the algorithmic procedure for solving the model is illustrated in the flowchart depicted in Figure 4.

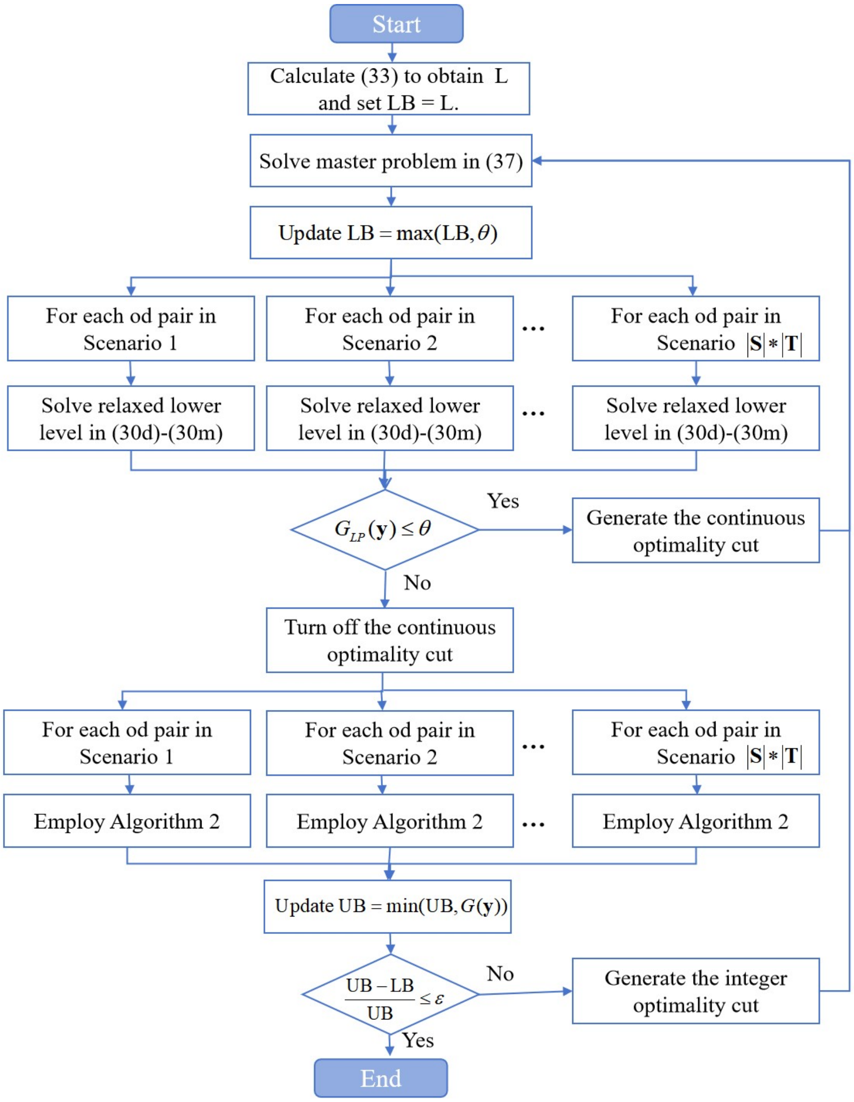

Figure 4: Flowchart of the integer L-shaped algorithm with continuous and integer optimality cuts.