Public transport route planning in the stochastic network based on the user individual preferences

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

A.A. Borodinov, A.S. Yumaganov, A.A. Agafonov

Transport Network Model

The part of the graph representing the two routes R_1 and R_2, containing the common stop S_k, is shown in figure 1.

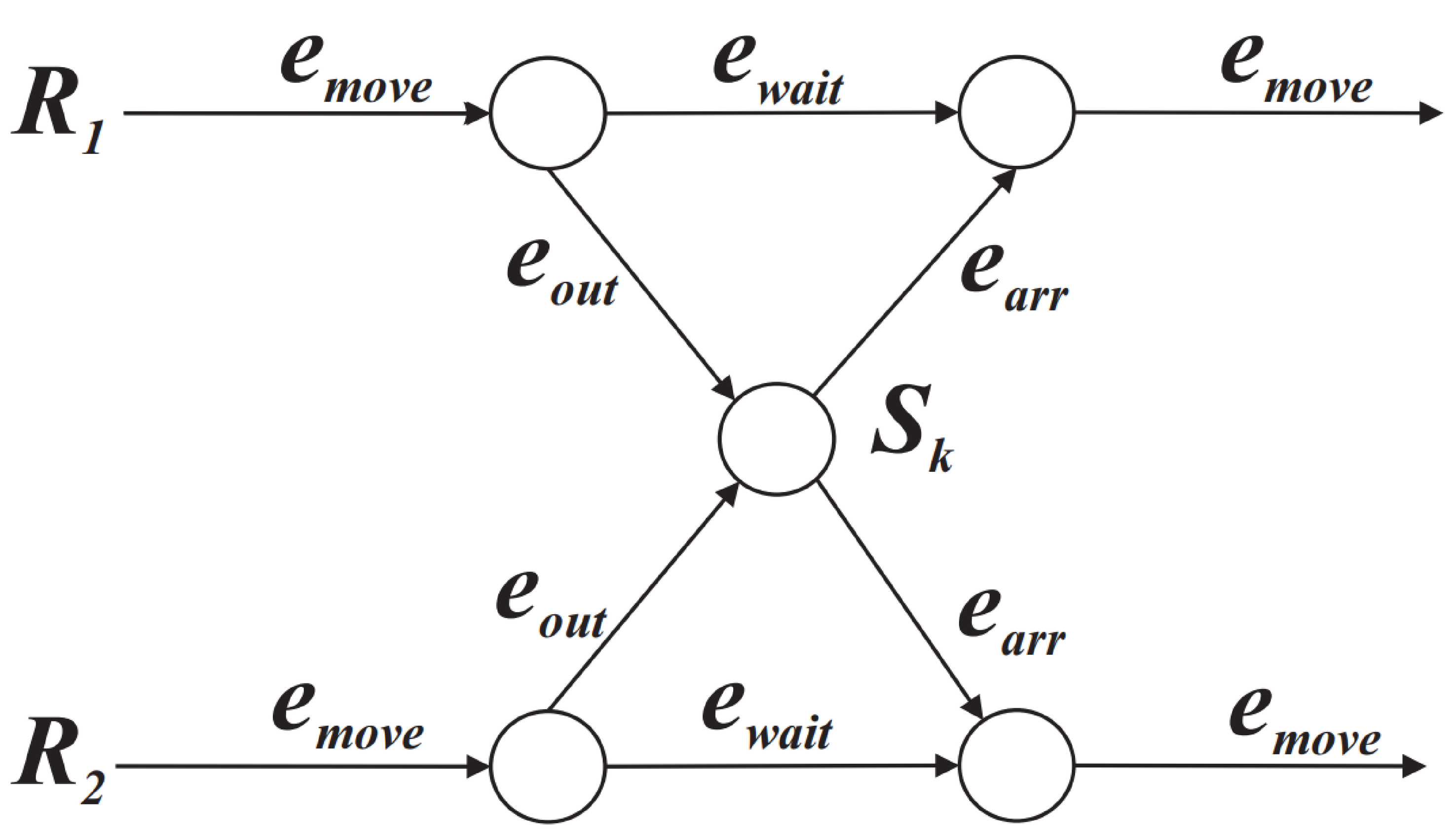

Figure 1. The route network graph structure.

The graph contains edges of several types:

- e_{move} - represents the segment of the route between stops, characterized by the average time μ_{ij} and the dispersion σ_{ij};
- e_{wait} - waiting for passengers boarding/alighting at the bus stop, in the work, constant waiting time is used as the edge weight;
- e_{out} - alighting edge, in the work as the edge weight is used constant alighting time;
- e_{arr} - boarding edge (vehicle waiting) is determined by the vehicle arrival time of the corresponding route to stop;
- e_{walk} (not shown in the figure) - walk from stops to the points of departure/arrival, the edge weight depends on the transition distance;

Transport Network Model

Dijkstra's modified algorithm is used to find the shortest path in a time-dependent stochastic transport network.

Algorithm 1: Algorithm for finding the shortest path

Input data: departure vertex n_s, arrival vertex n_d, departure time t
Output data: shortest path
`// Initialization`
PriorityQueue $pq = \emptyset$
Map $predMap = \emptyset$
Map $costsMap = \emptyset$
Label $l_s = Label(n_s, cost_s)$
pq.insert(l_s)
while $!pq = \emptyset$ **do**
  Label $l_i = pq$.pop()
  $n_i = l_i$.getNode()
  foreach $e_{ij} \in E$ **do**
    $cost_j = calculateCost(e_{ij}, t + l_i$.getCost().getTime())
    if $cost_j$.getCost() $>$ $costsMap$.get(n_i).getCost() **then**
      continue;
    end
    $costsMap$.put($n_j, cost_j$)
    $predMap.put(n_j, n_i)$
    Label $l_j = Label(n_j, cost_j)$
    pq.insert(l_j)
  end
end

The main interest is the estimation obtaining function of the passing the graph edge price *calculateCost*(). The proposed method for determining the cost of an edge depending on its type is described in Algorithm 2. In algorithm 2, the following notation is used:

- t_{wait} = 30 (seconds) - alighting/boarding time;
- t_{out} = 10 (seconds) - alighting time;
- t_{arr} - the vehicle arrival estimated time at the stop;
- s_{walk} = 1 (m/s) - walking speed;
- α_0; α_1; α_2 - individual preferences parameters that affect the reliable path choice, the transfers number and walking distance.

Algorithm 2: $calculateCost$

Input data: graph edge e_ij, arrival time t to the vertex i
Output data: cost $cost_j = (c_j, t_j)$
if e_{ij} *is* e_{move} **then**
  `// route segment between stops`
  return $(\mu_{ij} + \alpha_0\sqrt{\sigma_{ij}},\ \mu_{ij})$;
end
else if e_{ij} *is* e_{wait} **then**
  `// alighting/boarding`
  return $(t_{wait},\ t_{wait})$;
end
else if e_{ij} *is* e_{out} **then**
  `// alighting`
  return $(t_{out},\ t_{out})$;
end
else if e_{ij} *is* e_{arr} **then**
  `// waiting for transport`
  return $(\alpha_1(t_{arr} - t),\ t_{arr} - t)$;
end
else if e_{ij} *is* e_{walk} **then**
  `// foot transition`
  return $(\alpha_2|e_{ij}|/s_{walk},\ |e_{ij}|/s_{walk})$;
end

Experimental studies of the developed method were carried out for the Samara street-road network. The road network consists of 48139 segments. The data on the passenger transport bus routes movement were used to predict the time of arrival.

Six pairs of different departure and arrival vertices on the transport network graph were chosen to study the algorithm quality, after which for each pair they solved the problem of finding a way on public transport, varying the departure time.

The average trip time was t_{avg} = 2645 seconds. The following values were obtained from the experimental analysis results:

***MAPE* = 6.62%, *MAE* = 179.8 seconds**.

The obtained values allow us to conclude about the good quality of the proposed algorithm.

Acknowledgments

The work was funded by the Ministry of Science and Higher Education of the Russian Federation (unique project identifier RFMEFI57518X0177).