Just a cut from paper by Heinz Spiess and Michael Florian, Optimal Strategies: A New Assignment Model for Transit Networks, 1989, pages 93-94

Algorithm 1: Algorithm that solves the transit assignment problem

Result: Optimal strategy

begin
  // Part 1: Find optimal strategy
  // 1.1 Initialization
  $u_r = 0$
  $u_i = \infty$ for all $i \in I \setminus \{r\}$
  $f_i = 0$ for all $i \in I$
  $S = A$; $\overline{A} = \emptyset$
  // 1.2 Get next link
  if $S = \emptyset$ **then**
    STOP;
  end
  else
    find $a = (i, j) \in S$ which satisfies
    $u_j + c_a \leq u_{j'} + c_{a'}$, $a' = (i', j') \in S$
    $S = S \setminus a$
  end
  // 1.3 Update node label
  if $u_i \geq u_j + c_a$ **then**
    $u_i = \frac{f_i u_i + f_a (u_j + c_a)}{f_i + f_a}$
    $f_i = f_i + f_a$
    $\overline{A} = \overline{A} \cup a$
  end
  goto step 1.2
  Part 2: Assign demand according to optimal strategy
  // 2.1 Initialization
  $V_i = g_i$, $i \in I$
  // 2.2 Loading
  Do for every link $a \in A$, in decreasing order of $(u_j + c_a)$
  if $a \in \overline{A}$ **then**
    $v_a = \frac{f_a}{f_i} V_i$
    $V_j = V_j + v_a$
  end
  else
    $v_a = 0$
  end
end

Steps 1.2–1.3 form a loop: after each examined link the algorithm returns to step 1.2, until S is exhausted (the STOP condition). In code this is simply a while-loop over the priority queue, popping the link with the minimal $u_j + c_a$ on every iteration.

Remark (implementation). All equation, proposition and page numbers in this remark refer to the original paper. The implementation uses the strict test $u_i > u_j + c_a$ in step 1.3 instead of the $u_i \geq u_j + c_a$ printed above, so eq. (25) holds with $u_i \leq u_j + c_a$ for links outside $\overline{A}$. At exact equality the label update is a no-op: the combination formula returns u_i unchanged, and for $f_a = \infty$ the basket is replaced at the same value. Hence the labels $\hat{u}$, and with them the optimal expected travel times, are identical under both rules; the rules differ only in the support $\overline{A}$ (a degenerate optimum: for the link rejected at equality, $\mu_a = 0$ satisfies dual feasibility (20) as an equality and complementary slackness (24) holds since $v_a = 0$, so Propositions 3 and 5 are unaffected).

The gap closed by the strict test lies in Proposition 4. Its proof asserts flow conservation "by construction", relying on the claim of p. 94 that processing links in decreasing order of $u_j + c_a$ is a reverse topological order of $\overline{A}$. That claim holds only if $\overline{A}$ is acyclic (and zero-cost key ties between dependent links are broken consistently), which the nonstrict test does not guarantee: in an expanded route graph a boarding link of

cost 0 into a route node whose label was set by its own alighting link of cost 0 has key exactly u_i, so $\geq$ admits the zero-cost cycle stop $\to$ route node $\to$ stop. On such a support the one-pass loading of step 2.2 strands the volume entering the cycle, violating conservation, so $\hat{v}$ is not primal feasible and Proposition 5 no longer applies. The strict test makes the premise of Proposition 4 a theorem: labels never increase during part 1, so accepting the closing link of a zero-cost cycle would require u_i to exceed its own earlier value, a contradiction; $\overline{A}$ stays acyclic and the one-pass loading conserves flow. The prose of p. 94 ("if this time is smaller than u_i, link a is included") describes the strict rule.