5.8 Example of the Time-Independent Label-Constrained Shortest Path Problem (TILSP)

Suppose that we are given a single-trip request shown in Table 4 for a traveler designated ID 13300, involving a trip starting from "home" and going to "work". The admissible mode strings are "*w…wc…cw…w*" or "*w…wb…bw…w*". Note that in this context, as in TRANSIMS, the string *w…wc…cw…w*, for example, represents a *sequence* of one or more walk links, followed by one or more car links, and ending with one or more walk links.

Table 4: Single-Trip Requests for a Traveler.

Person ID	Trip Number	Starting Location	Destination Location	Starting Time (seconds since midnight)	Maximum Travel Time (second)	Mode String
13300	1	845654 (Home)	833503 (Work)	28800 (8.00 a.m.).	*3000 **	wcw or wbw

**Note: the maximum travel time is 3000 seconds hence the maximum finish time equals the starting time plus the maximum travel time, which is 31800 seconds since midnight (8.50 a.m.).*

Furthermore, suppose that we have constructed the Internal Network as given in Figure 17, having unidirectional links, along with associated constant travel times, and **travel mode** labels. This example is the same instance described earlier for TISP, except that the links have mode-labels associated with them such as *w*, *c*, and *b*, and the problem has an associated label string restriction.

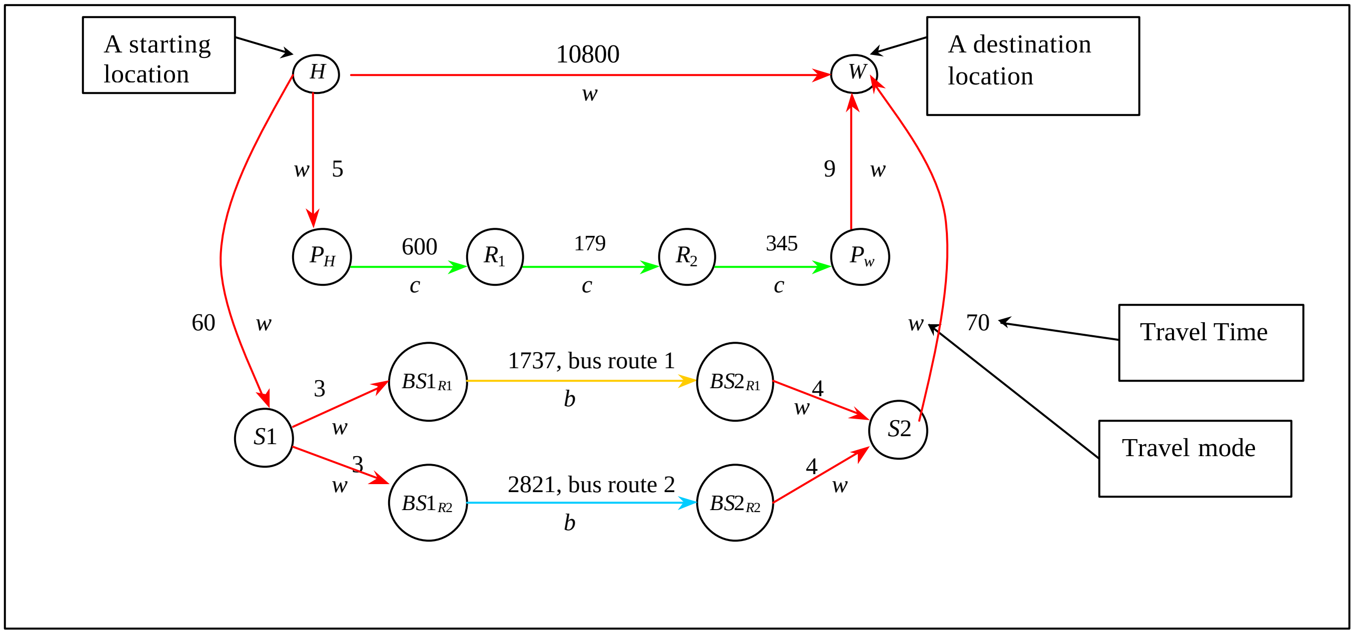


Figure 17: The Internal Network representation for the example for TILSP.

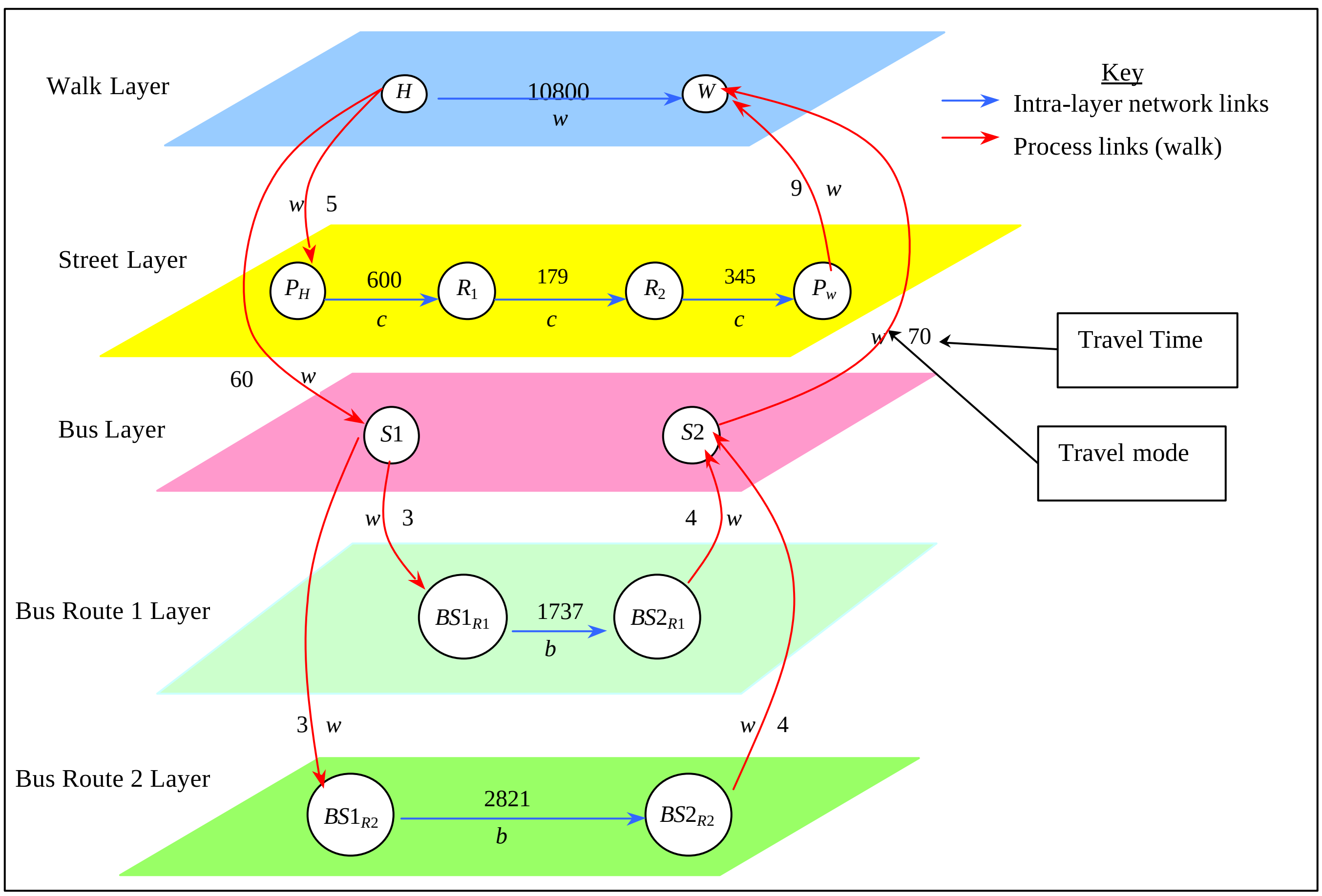
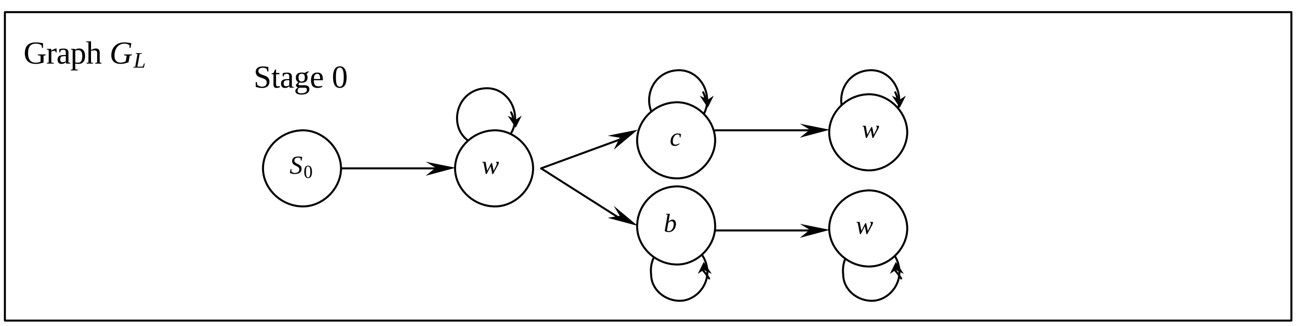


Figure 18: Layers of the Internal Network for the example for TILSP.

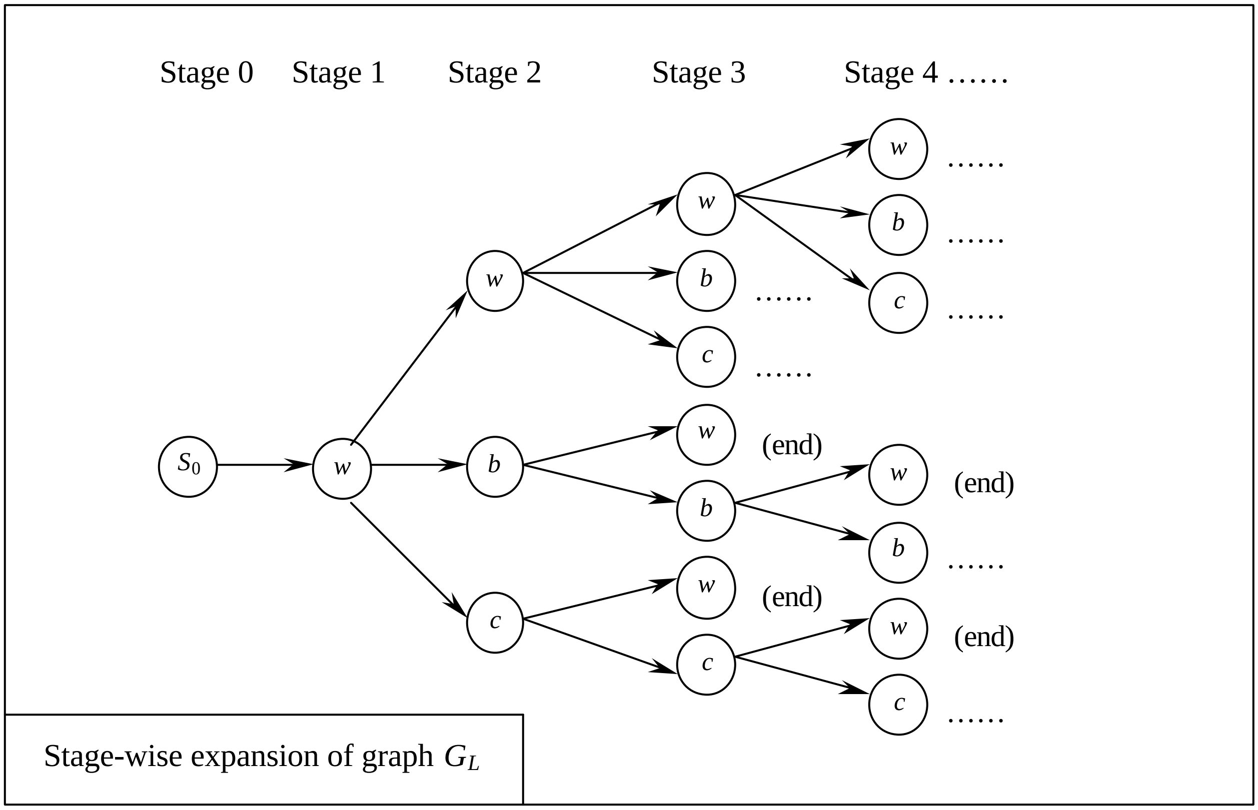
Figure 18 depicts the layers of the Internal Network for this example. The starting location (H) and the ending location (W) are placed on the walk layer. The street layer provides the street network. Here, we have four street nodes (P_H: car parking at home, R_1: start of road 1, R_2: start of road 2, and P_W: car parking at work). This Internal Network has two bus routes. The bus layer contains the two bus shelters for the passengers ($S1$ and $S2$). The bus route 1 layer contains a bus route 1 network, which has only one link in our example, from $BS1_{R1}$ (a bus-parking place 1 for bus route 1) to $BS2_{R1}$ (a bus-parking place 2 for bus route 1). The bus route 2 layer contains a bus route 2 network, which also has only one link in our example, from $BS1_{R2}$ (a bus-parking place 1 for bus route 2) to $BS2_{R2}$ (a bus-parking place 2 for bus route 2).

Step 1

Examining the admissible mode strings "*w…wc…cw…w*" and "*w…wb…bw…w*", we can construct a corresponding *transition graph* G_L as follows. We begin with a single node corresponding to a *dummy label* s_0 representing Stage 0. Then, the next transition (Stage 1) is necessarily conducted via a walk link. We might continue to walk over several subsequent stages (depicted by the self-loop in G_L from w to itself), or transition via a link that represents a car travel or via a link that represents a bus travel. The remainder of G_L shown below has a similar interpretation.



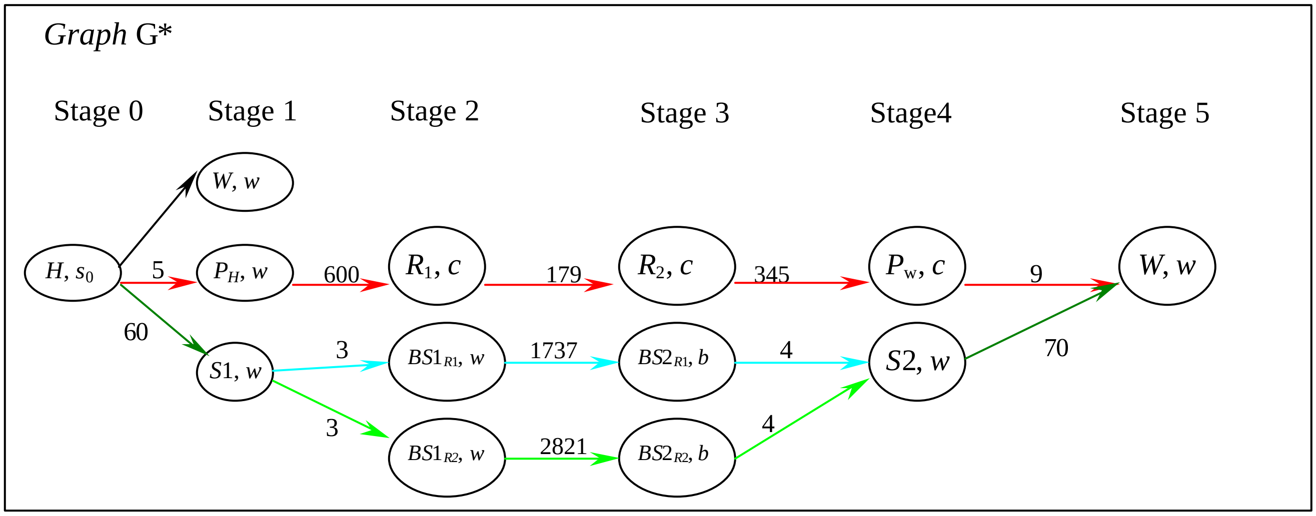
A stage-wise partial blow-up of the graph G_L is shown below.



Stage-wise expansion of graph G_L

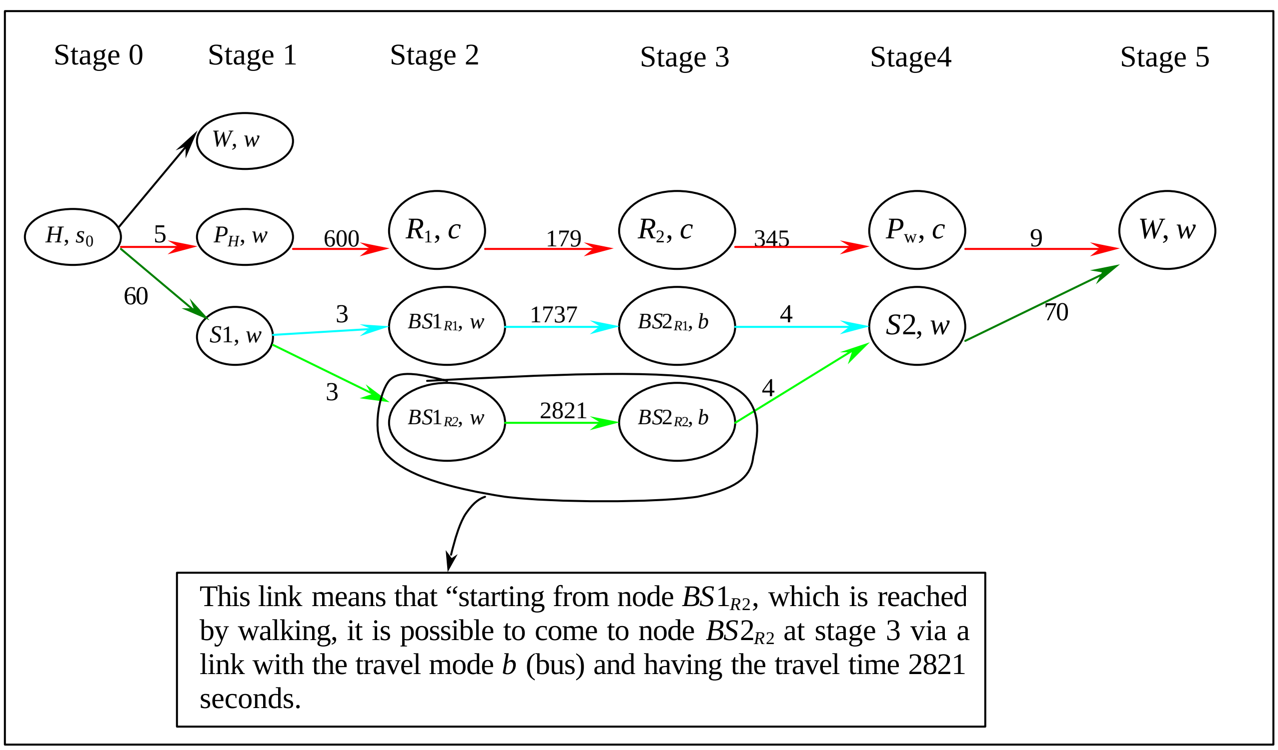
Step 2

Using the graph G_L and the Internal Network shown in Figure 17, we can construct a combined graph G^* as described earlier. The actual shortest path problem will be solved on this graph G^*. Beginning with Stage 0, the graph G^* has a node (H, s_0). Recursively, we determine nodes for each subsequent Stage s as shown below in order to construct G^*.



The graph G^* contains all possible admissible paths starting from the node (H, s_0) and ending at node (W, w) at Stage 5. We can see that there are three possible paths. The first path uses a *wcccw*-mode string. The second path uses a *wwbww*-mode string. The third path also uses a *wwbww*-mode string (on a different bus route). Note that the link from (H, s_0) to (W, w) at Stage 1 has no feasible continuation, and this node (W, w) at Stage 1 is *not* a legitimate terminal node (unless if the string *www…w* is admissible).

The diagram below offers a specific sample explanation of a feature of the graph G^*.



Step 3

In the graph G^*, find the shortest path from the starting node (H, s_0) to the destination node (W, w) at Stage 5 using any standard shortest path algorithm. Here we use Dijkstra's algorithm as in the previous example. This yields a shortest path for the Time-Independent Label-Constrained Shortest Path Problem (TILSP), which turns out to be the same as the solution for the previous example (as shown in Figure 19).

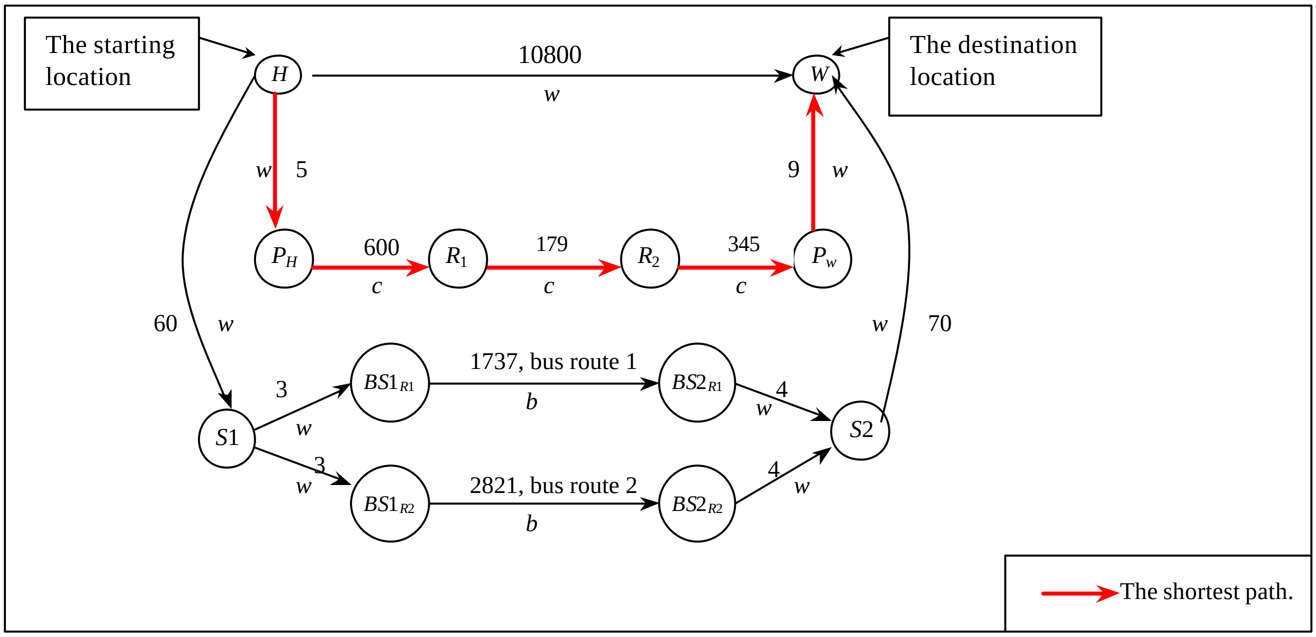


Figure 19: The shortest path solution for the example for TILSP.

Next, we consider the case where the travel time on a link is dependent on the arrival time on that link.

5.9. Example of the Route Planner Module (Time-Dependent Label-Constrained Shortest Path Problem)

In this example we are using the same single-trip request provided in the earlier example (as shown in Table 5) for the traveler designated ID 13300, with admissible mode string "*w…wc…cw…w*" or "*w…wb…bw…w*".

Table 5: Single-Trip Requests for a Traveler.

Person ID	**Trip Number**	**Starting Location**	**Destination Location**	**Starting Time (seconds since midnight)**	**Maximum Travel Time (second)**	**Mode String**
13300	1	845654 (Home)	833503 (Work)	28800 (8.00 a.m.).	*3000* *	wcw or wbw

*N*ote: the maximum travel time is 3000 seconds, hence, the maximum **finish time** equals the starting time plus the maximum travel time, which is 31800 seconds since midnight (8.50 a.m.).*

Also, we are using the same Internal Network as shown in Figure 19, except that the links have time-dependent travel time functions $d_{ij}(t)$, excluding the walk links, which are time-independent.

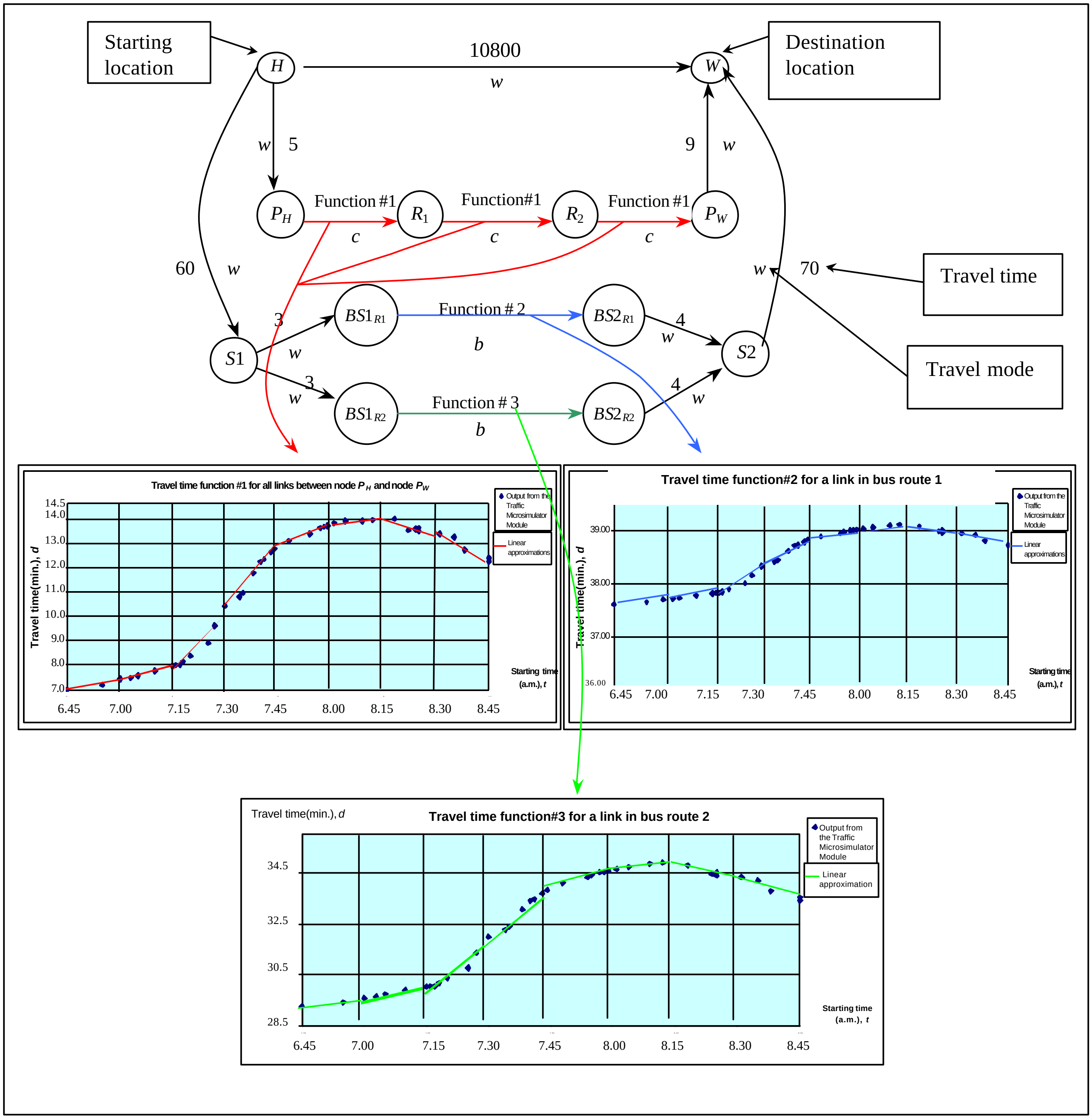


Figure 20: The Internal Network representation for the example for TDLSP.

In Figure 20, the starting location of the trip is node H (home), and the destination location is node W (work location). Figure 21 depicts the layers of the Internal Network, where the links have time-dependent travel time functions.

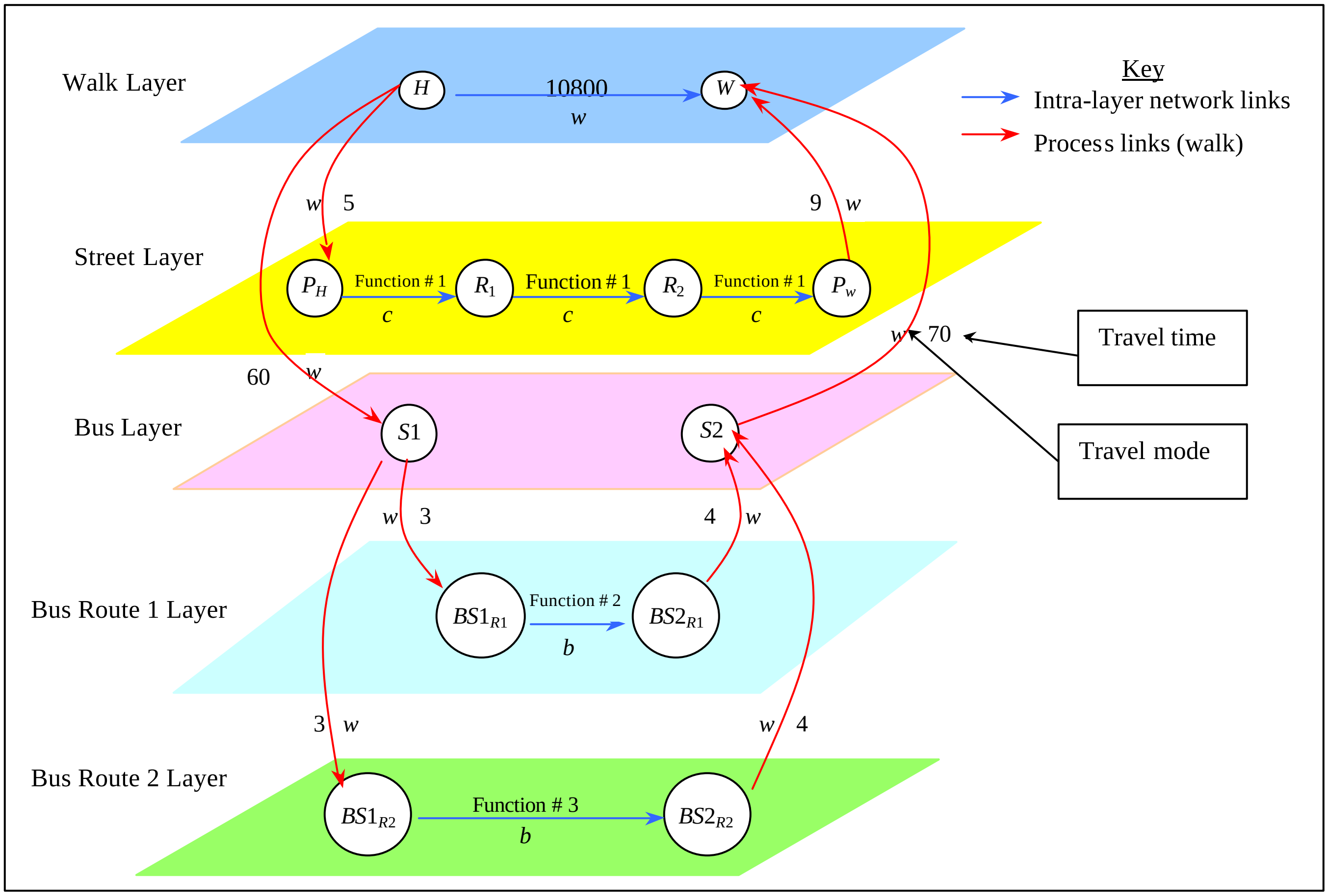
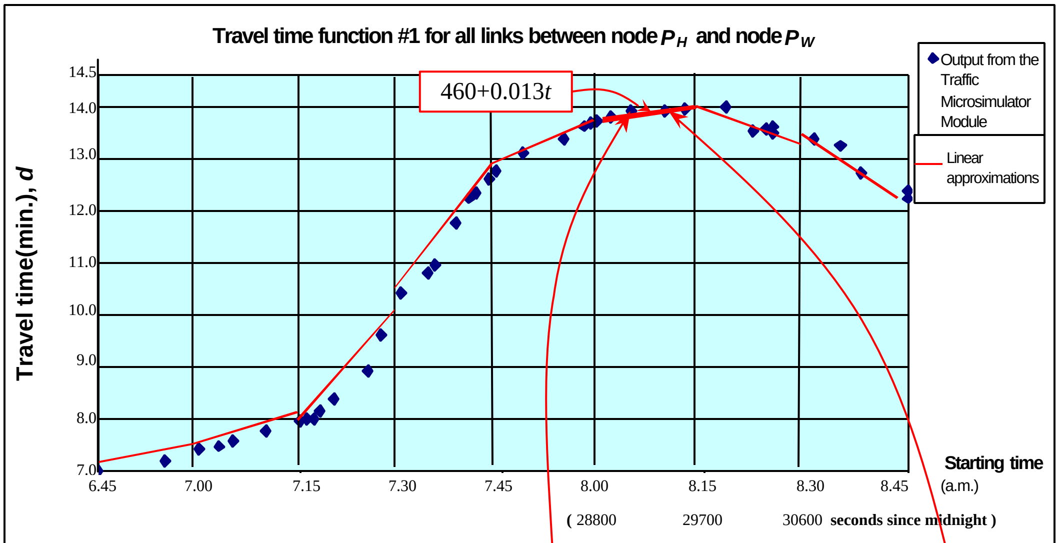


Figure 21: Layers of the Internal Network for the example for TDLSP.

The same procedure is used in developing the graph G_L in TDLSP as described earlier in TILSP except that the links have travel time values dependent on the arrival time at the starting node of that link. To avoid repetition of the procedure for determining graph G_L, we skip directly to the solution of the actual example.

Initialization:

$s = 0$ $$N_0 = \{(H, 288000, s_0)\} \begin{matrix} \xrightarrow{5} \\ \xrightarrow[60]{} \end{matrix} \begin{Bmatrix} (P_H, 28805, w) \\ (S1, 28860, w) \end{Bmatrix} = N_1$$



Obtained from a travel time function for the 8.00-8.15 a.m. interval because the starting time is 28805 (8.08 a.m.).

Also obtained from a travel time function for the 8.00-8.15 a.m. interval because the starting time is 29639 (8.14 a.m.).

<u>s = 1</u>

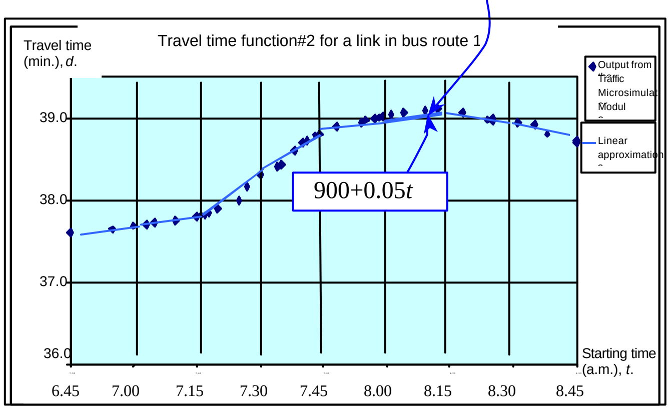
$$N_1 = \begin{Bmatrix}(P_H, 28805, w)\\ (S1, 28860, w)\end{Bmatrix} \begin{matrix}\xrightarrow{460+0.013t}\\ \xrightarrow{3}\\ \xrightarrow{3}\end{matrix} \begin{Bmatrix}(R_1, 29639, c)\\ (BS1_{R1}, 28863, w)\\ (BS1_{R2}, 28863, w)\end{Bmatrix} = N_2$$

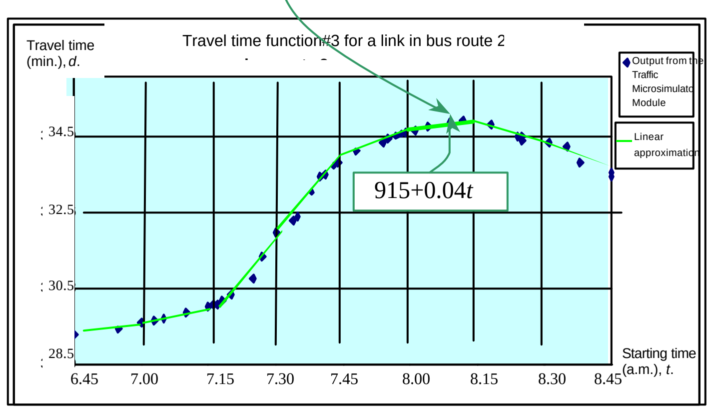
<u>s = 2</u>

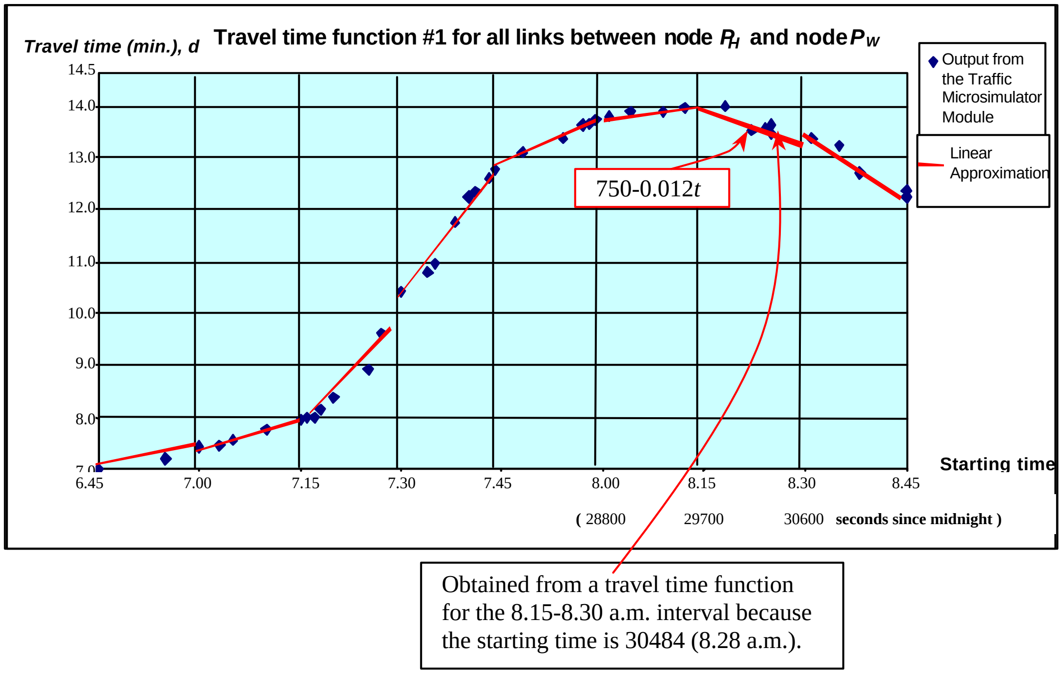
$$N_2 = \begin{Bmatrix}(R_1, 29639, c)\\ (BS1_{R1}, 28863, w)\\ (BS1_{R2}, 28863, w)\end{Bmatrix} \begin{matrix}\xrightarrow{460+0.013t}\\ \xrightarrow{900+0.05t}\\ \xrightarrow{915+0.04t}\end{matrix} \begin{Bmatrix}(R_2, 30484, c)\\ (BS2_{R1}, 31206, b)\\ (Bs2_{R2}, 30933, b)\end{Bmatrix}$$

Obtained from a travel time function for the 8.00-8.15 a.m. interval because the starting time is 28863 (8.01a.m.).

Obtained from a travel time function for the 8.00-8.15 a.m. interval because the starting time is 28863 (8.01 a.m.).







<u>s = 3</u>

$$N_3 = \begin{Bmatrix} (R_2, 30484, c) \\ (BS\,2_{R1}, 31206, b) \\ (Bs\,2_{R2}, 30933, b) \end{Bmatrix} \begin{matrix} \xrightarrow{\overbrace{750\text{-}0.012t}} \\ \xrightarrow{4} \\ \xrightarrow{4} \end{matrix} \begin{Bmatrix} (P_w, 30868, c) \\ (S2, 31210, w) \\ (S2, 30937, w) \end{Bmatrix} = N_4$$

This is eliminated later because we select min {31210, 30937}.

<u>s = 4</u>

$$N_4 = \begin{Bmatrix} (P_w, 30868, c) \\ (S2, 30937, w) \end{Bmatrix} \begin{matrix} \xrightarrow{9} \\ \xrightarrow{70} \end{matrix} \begin{Bmatrix} (W, 30877, w)^* \\ (W, 31007, w) \end{Bmatrix} \rightarrow \{(W, 30877, w)\}$$

<u>s = 5</u> Terminate with *Destination node* = W, and with the node (W, 30877, w) as the terminal node of the shortest path. Note that there are no paths that exceed the maximum finish time (31800 seconds since midnight). Tracing backwards yields the path $H \rightarrow P_H \rightarrow R_1 \rightarrow R_2 \rightarrow P_w \rightarrow W$ (as shown in Figure 22), having a travel mode *wcccw* and an ending time of t^* = 30877 seconds since midnight, or 8.35 a.m. The total travel time is (30877 – 28800) = 2077 seconds, which is less than the maximum allowable travel time of 3000 seconds.

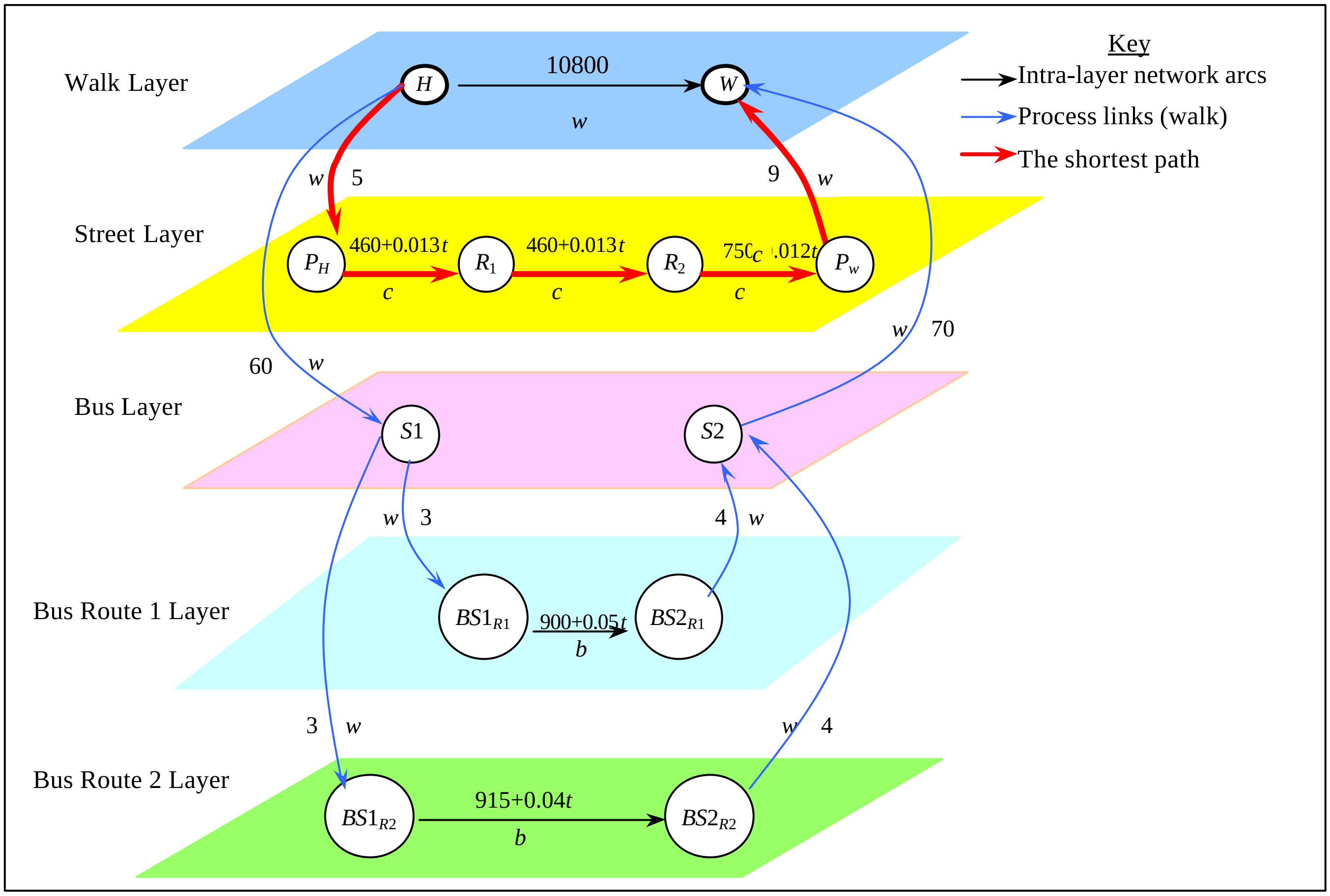


Figure 22: The shortest path solution for the example for TDLSP.

The shortest path has three legs:

- The first leg is a link between node H (home) to node P_H (car parking at home).
- The second leg is comprised of the links between node P_H and node R_1 (road 1), between node R_1 and node R_2 (road 2), and between node R_2 and node P_w (car parking at work).
- The last leg is a link between node P_w and node W (work).