CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

8.1 Conclusions

In this thesis, we have proposed a partitioned shortest path algorithm for solving time-dependent label-constrained shortest path problems (referred hereafter as the **exact** algorithm), along with four heuristic techniques to decrease the computational effort while maintaining good quality solutions. The results of our experiments using the heuristic methods exhibits that properly designed parameters can indeed decrease the computational time by (overall) 30%, and provide solutions that are competitive with that obtained for the exact algorithm (with the overall average quality being within 7.8% of optimality). Sometimes (27% in overall average), the heuristic methods yield optimal solutions.

Furthermore, we have provided a comparative analysis of the alternative heuristic methods for different parameter values. Toward this end, we evaluated the **computational time** and **the quality of solutions** for different parameter values, and then identified the best parameter choices for each method, providing a motivation for this choice. Based on these two factors, the ranks of the heuristic methods are as shown below:

Rank based on the computational time (starting from the best alternative)

- Heuristic Method (ii): Network Sectioning Technique,
- Heuristic Method (iii)-2: Level-Based Technique (Linear Relationship Function),
- Heuristic Method (iv): Ellipsoidal Region Technique,
- Heuristic Method (iii)-1: Level-Based Technique (Exponential Relationship Function), and
- Heuristic Method (i): Standard Base-Case.

Rank based on the solution quality (starting from the best alternative)

- Heuristic case (i),
- Heuristic Method (iii)-1: Level-Based Technique (Exponential Relationship Function),
- Heuristic case (iv): Ellipsoidal Region Technique
- Heuristic Method (iii)-2: Level-Based Technique (Linear Relationship Function), and
- Heuristic case (ii): Network Sectioning Technique.

In order to select the best method, one needs to consider a trade-off between these factors. By our judgement, we prescribe either method (i), method (iii)-1 (when $a = 0.25$), or method (iv) (when $g = 1.25$ and $y = 0.75$) as the best compromise between the solution quality and the computational time effort. Tables 19 to 21 present the detailed test results obtained for method (i), method (iii)-1, and method (iv), respectively, versus the exact algorithm.

Table 19: Detailed Results for Heuristic (i): Standard Base-Case ($b_i = 1 \ \forall \ i$).

		Exact algorithm			**Heuristic (i)**				
Trip Type	Problem Class	Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. % of heuristic methods that yielded opt. solns.	Avg. soln. quality
I	1	52.718	315	*wcw*56%, *wbw*39%	51.903	323	*wcw*52%, *wbw*40%	24	1.085
	2	39.267	229	*wcw*44%, *wrw*37%, *wbw*15%	37.157	220	*wcw*40%, *wrw*39%, *wbw*16%	28	1.090
	3	27.190	149	*wcw*49%, *wrw*35%, *wbw*11%	23.841	145	*wcw*48%, *wrw*32%, *wbw*14%	29	1.012
	4	42.081	251	*wcw*41%, *wrw*39%, *wbw*12%	39.883	268	*wcw*42%, *wrw*37%, *wbw*13%	27	1.034
	5	40.497	219	*wcw*48%, *wbw*36%, *wbrw*14%	39.557	220	*Wcw*50%, *wbw*32%, *wbrw*13%	27	1.091
	6	44.891	279	*wcw*55%, *wbw*44%	44.160	288	*wcw*53%, *wbw*44%	26	1.029
	7	37.564	211	*wcw*38%, *wrw*38%, *wbw*14%, *wbrw*10%	37.549	214	*wcw*40%, *wrw*36%, *wbw*13%, *wbrw*10%	25	1.065
	8	39.691	226	*wcw*42%, *wrw*28%, *wbw*19%, *wbrw*10%	31.178	219	*wcw*43%, *wrw*25%, *wbw*20%, *wbrw*11%	26	1.098
	9	38.002	219	*wcw*43%, *wrw*30%, *wbw*15%, *wbrw*10%	32.167	236	*wcw*42%, *wrw*31%, *wbw*14%, *wbrw*11%	26	1.093
	10	40.257	278	*wcw*38%, *wbw*29%, *wrw*18%, *wbrw*14%	36.037	288	*wcw*39%, *wbw*27%, *wrw*19%, *wbrw*14%	25	1.068
II	11	26.583	158	*wcw*45%, *wbw*41%	22.180	160	*wcw*43%, *wbw*43%	27	1.034
	12	29.106	262	*wcw*42%, *wbw*40%, *wrw*17%	21.718	259	*wcw*45%, *wbw*41%, *wrw*14%	27	1.048
	13	43.097	294	*wcw*46%, *wbw*45%	41.289	288	*wcw*49%, *wbw*44%	26	1.055
	14	49.027	287	*wcw*44%, *wbw*46%	36.371	290	*wcw*43%, *wbw*46%	26	1.092
	15	40.097	271	*wcw*37%, *wbw*30%, *wrw*18%, *wbrw*13%	36.587	289	*wcw*37%, *wbw*29%, *wrw*18%, *wbrw*13%	25	1.094
III	16	53.245	327	*wcw*54%, *wbw*40%	50.647	335	*wcw*56%, *wbw*41%	26	1.012
	17	39.891	211	*wcw*43%, *wrw*36%, *wbw*17%	30.023	208	*wcw*40%, *wrw*37%, *wbw*17%	28	1.071
	18	31.081	206	*wcw*47%, *wrw*40%, *wbw*10%	23.207	214	*wcw*46%, *wrw*41%, *wbw*11%	26	1.034
	19	40.992	224	*wcw*49%, *wbw*34%, *wbrw*15%	31.754	231	*wcw*50%, *wbw*34%, *wbrw*15%	28	1.070
	20	41.037	188	*wcw*57%, *wbw*42%	29.274	195	*wcw*58%, *wbw*41%	26	1.013
	21	38.510	222	*wcw*37%, *wrw*38%, *wbw*13%, *wbrw*12%	29.451	235	*wcw*36%, *wrw*36%, *wbw*13%, *wbrw*15%	25	1.033
	22	38.159	225	*wcw*42%, *wrw*29%, *wbw*17%, *wbrw*11%	30.321	223	*wcw*44%, *wrw*29%, *wbw*11%, *wbrw*10%	27	1.090
	23	40.197	282	*wcw*38%, *wbw*28%, *wrw*19%, *wbrw*14%	33.341	294	*wcw*40%, *wbw*27%, *wrw*18%, *wbrw*15%	25	1.081
		Avg.= **39.703** s.	**241**		Avg.= **34.330** s.	**245**		Avg.= **26%**	**1.061**
		STD=6.794	46		8.134	49		1.22	0.030

Table 20: Detailed Results for Heuristic (iii)-1: Level-Based Technique Using an Exponential Decay Function.

Trip Type	Problem Class	**Exact algorithm**			**Heuristic (iii)-1**				
		Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. % of heuristic methods that yielded opt. solns.	Avg. soln. quality
I	1	52.718	315	*wcw*56%, *wbw*39%	44.903	328	*wcw*55%, *wbw*41%	21	1.087
	2	39.267	229	*wcw*44%, *wrw*37%, *wbw*15%	32.208	239	*wcw*43%, *wrw*37%, *wbw*16%	22	1.091
	3	27.190	149	*wcw*49%, *wrw*35%, *wbw*11%	20.781	157	*wcw*46%, *wrw*33%, *wbw*15%	25	1.055
	4	42.081	251	*wcw*41%, *wrw*39%, *wbw*12%	33.837	249	*wcw*42%, *wrw*38%, *wbw*12%	24	1.049
	5	40.497	219	*wcw*48%, *wbw*36%, *wbrw*14%	29.469	231	*wcw*48%, *wbw*34%, *wbrw*16%	24	1.094
	6	44.891	279	*wcw*55%, *wbw*44%	34.495	265	*wcw*55%, *wbw*45%	25	1.038
	7	37.564	211	*wcw*38%, *wrw*38%, *wbw*14%, *wbrw*10%	29.340	224	*wcw*38%, *wrw*38%, *wbw*14%, *wbrw*10%	23	1.071
	8	39.691	226	*wcw*42%, *wrw*28%, *wbw*19%, *wbrw*10%	33.686	231	*wcw*43%, *wrw*25%, *wbw*19%, *wbrw*12%	22	1.098
	9	38.002	219	*wcw*43%, *wrw*30%, *wbw*15%, *wbrw*10%	31.216	217	*wcw*44%, *wrw*31%, *wbw*12%, *wbrw*11%	24	1.096
	10	40.257	278	*wcw*38%, *wbw*29%, *wrw*18%, *wbrw*14%	33.316	284	*wcw*39%, *wbw*27%, *wrw*18%, *wbrw*14%	24	1.058
II	11	26.583	158	*wcw*45%, *wbw*41%	15.707	167	*wcw*42%, *wbw*43%	27	1.052
	12	29.106	262	*wcw*42%, *wbw*40%, *wrw*17%	18.214	254	*wcw*45%, *wbw*41%, *wrw*14%	23	1.050
	13	43.097	294	*wcw*46%, *wbw*45%	35.207	289	*wcw*50%, *wbw*45%	24	1.084
	14	49.027	287	*wcw*44%, *wbw*46%	41.617	294	*wcw*44%, *wbw*48%	25	1.056
	15	40.097	271	*wcw*37%, *wbw*30%, *wrw*18%, *wbrw*13%	30.547	269	*wcw*38%, *wbw*28%, *wrw*18%, *wbrw*13%	26	1.099
III	16	53.245	327	*wcw*54%, *wbw*40%	43.129	321	*wcw*55%, *wbw*41%	22	1.069
	17	39.891	211	*wcw*43%, *wrw*36%, *wbw*17%	29.143	217	*wcw*40%, *wrw*36%, *wbw*18%	21	1.081
	18	31.081	206	*wcw*47%, *wrw*40%, *wbw*10%	21.634	212	*wcw*46%, *wrw*41%, *wbw*11%	24	1.045
	19	40.992	224	*wcw*49%, *wbw*34%, *wbrw*15%	30.818	235	*wcw*51%, *wbw*35%, *wbrw*14%	23	1.019
	20	41.037	188	*wcw*57%, *wbw*42%	29.951	199	*wcw*58%, *wbw*40%	26	1.018
	21	38.510	222	*wcw*37%, *wrw*38%, *wbw*13%, *wbrw*12%	26.474	215	*wcw*37%, *wrw*36%, *wbw*13%, *wbrw*14%	24	1.015
	22	38.159	225	*wcw*42%, *wrw*29%, *wbw*17%, *wbrw*11%	25.129	236	*wcw*45%, *wrw*28%, *wbw*10%, *wbrw*11%	22	1.092
	23	40.197	282	*wcw*38%, *wbw*28%, *wrw*19%, *wbrw*14%	25.155	286	*wcw*40%, *wbw*26%, *wrw*18%, *wbrw*16%	27	1.086
		Avg.= **39.703** s.	**241**		Avg.= **30.260** s.	**244**		Avg.= **24%**	**1.065**
		STD=6.794	46		7.327	44		1.72	0.027

Table 21: Detailed Results for Heuristic (iv): Ellipsoidal Region Technique.

		Exact algorithm			**Heuristic (iv)**				
Trip Type	Problem Class	Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. CPU time (s/trip)	Avg. no. iterations, *l*(*t*)	Avg. % of mode strings used	Avg. % of heuristic methods that yielded opt. solns.	Avg. soln. quality
I	1	52.718	315	*wcw*56%, *wbw*39%	27.541	321	*wcw*54%, *wbw*40%	26	1.092
	2	39.267	229	*wcw*44%, *wrw*37%, *wbw*15%	25.941	223	*wcw*43%, *wrw*37%, *wbw*16%	22	1.041
	3	27.190	149	*wcw*49%, *wrw*35%, *wbw*11%	22.947	140	*wcw*49%, *wrw*31%, *wbw*14%	19	1.045
	4	42.081	251	*wcw*41%, *wrw*39%, *wbw*12%	36.028	258	*wcw*44%, *wrw*36%, *wbw*12%	21	1.053
	5	40.497	219	*wcw*48%, *wbw*36%, *wbrw*14%	30.948	216	*wcw*51%, *wbw*32%, *wbrw*12%	22	1.047
	6	44.891	279	*wcw*55%, *wbw*44%	28.371	288	*wcw*53%, *wbw*45%	24	1.084
	7	37.564	211	*wcw*38%, *wrw*38%, *wbw*14%, *wbrw*10%	23.907	215	*wcw*41%, *wrw*35%, *wbw*13%, *wbrw*10%	25	1.050
	8	39.691	226	*wcw*42%, *wrw*28%, *wbw*19%, *wbrw*10%	26.483	220	*wcw*45%, *wrw*24%, *wbw*20%, *wbrw*10%	26	1.083
	9	38.002	219	*wcw*43%, *wrw*30%, *wbw*15%, *wbrw*10%	22.496	230	*wcw*43%, *wrw*31%, *wbw*14%, *wbrw*10%	24	1.064
	10	40.257	278	*wcw*38%, *wbw*29%, *wrw*18%, *wbrw*14%	26.496	281	*wcw*40%, *wbw*27%, *wrw*18%, *wbrw*14%	25	1.065
II	11	26.583	158	*wcw*45%, *wbw*41%	20.049	159	*wcw*45%, *wbw*43%	21	1.176
	12	29.106	262	*wcw*42%, *wbw*40%, *wrw*17%	20.567	259	*wcw*46%, *wbw*41%, *wrw*13%	23	1.134
	13	43.097	294	*wcw*46%, *wbw*45%	31.284	293	*wcw*50%, *wbw*45%	23	1.050
	14	49.027	287	*wcw*44%, *wbw*46%	38.948	288	*wcw*45%, *wbw*46%	22	1.055
	15	40.097	271	*wcw*37%, *wbw*30%, *wrw*18%, *wbrw*13%	29.134	283	*wcw*38%, *wbw*29%, *wrw*17%, *wbrw*13%	23	1.057
III	16	53.245	327	*wcw*54%, *wbw*40%	25.989	332	*wcw*57%, *wbw*42%	24	1.102
	17	39.891	211	*wcw*43%, *wrw*36%, *wbw*17%	23.567	210	*wcw*42%, *wrw*37%, *wbw*17%	22	1.031
	18	31.081	206	*wcw*47%, *wrw*40%, *wbw*10%	26.456	212	*wcw*47%, *wrw*41%, *wbw*11%	24	1.070
	19	40.992	224	*wcw*49%, *wbw*34%, *wbrw*15%	23.456	230	*wcw*51%, *wbw*34%, *wbrw*15%	23	1.066
	20	41.037	188	*wcw*57%, *wbw*42%	27.784	195	*wcw*59%, *wbw*40%	24	1.100
	21	38.510	222	*wcw*37%, *wrw*38%, *wbw*13%, *wbrw*12%	23.147	226	*wcw*38%, *wrw*35%, *wbw*13%, *wbrw*14%	25	1.025
	22	38.159	225	*wcw*42%, *wrw*29%, *wbw*17%, *wbrw*11%	22.567	224	*wcw*45%, *wrw*28%, *wbw*10%, *wbrw*10%	23	1.066
	23	40.197	282	*wcw*38%, *wbw*28%, *wrw*19%, *wbrw*14%	22.469	291	*wcw*42%, *wbw*26%, *wrw*18%, *wbrw*14%	24	1.060
		Avg.= **39.703** s.	**241**		Avg.= **26.373** s.	**243**		Avg.= **23%**	**1.070**
		STD=6.794	46		4.646	49		1.68	0.034

In order to verify that the **anomalies** for both the travel times exceeding the threshold *T* and the infeasible *O-D* paths are comparable for the exact and the prescribed heuristic methods, we record the % of trips that yield such anomalies in Table 22. The results reveal that the anomalies for the prescribed heuristic methods (i), (iii)-1, and (iv), do not significantly differ from the anomalies for the exact algorithm. Hence, for our test network, the results from the heuristics are comparable to that from the exact algorithm.

Table 22: % of trips that yield anomalies (either the travel time >T or no feasible O-D path found).

Trip Type	Problem Class	Total no. of trips	**Travel Time > *T***				**No Feasible *O-D* Paths**			
			Exact Algorithm	Method (i)	Method (iii)-1	Method (iv)	Exact Algorithm	Method (i)	Method (iii)-1	Method (iv)
I	1	379	5	6	6	6	0	0.61	1.35	1.39
	2	316	3	3	4	5	0	0	0	0
	3	208	3	4	4	4	0	0	0	0
	4	284	4	3	4	4	0	0	1.29	1.23
	5	311	4	5	5	6	0	0	0	0.54
	6	298	4	5	5	5	0	0	1.12	1.10
	7	293	3	4	4	5	0	0	0	0
	8	176	4	5	5	5	0	0	0	0
	9	185	4	4	5	5	0	0	0	0
	10	108	4	4	4	5	0	0	0	0.43
II	11	217	3	4	4	4	0	0	0	0
	12	226	3	3	3	4	0	0	0	0
	13	220	4	3	4	4	0	0	1.30	1.35
	14	153	5	4	5	5	0	0	1.33	1.39
	15	135	4	5	5	5	0	0	0	0.37
III	16	149	5	5	5	6	0	0.63	1.36	1.41
	17	217	4	4	4	5	0	0	0	0
	18	117	3	4	4	4	0	0	0	0
	19	113	4	3	4	4	0	0	0	0.48
	20	104	4	4	4	5	0	0	0	0.52
	21	131	4	3	4	4	0	0	0	0
	22	95	4	4	5	5	0	0	0	0
	23	81	4	5	4	5	0	0	0	0
		Average	**3.87**	**4.09**	**4.39**	**4.78**	**0**	**0.05**	**0.34**	**0.44**
		STD	0.63	0.85	0.66	0.67	0	0.18	0.58	0.56

This thesis has demonstrated that **time-dependent travel times** as well as **label constraints** can be well incorporated within shortest path problems on networks. These considerations address shortest problems that arise in most realistic situations where travel times/link costs are functions of time, and where a certain admissible string of travel modes needs to be considered while selecting the shortest route.

8.2 Recommendations for Future Research

In order to improve the applicability of the proposed time-dependent label-constrained shortest path procedure, especially in the context of time-dependent travel time, a **real time** technology and implementation needs to be composed using either exact or heuristic methods. This can facilitate the feedback of dynamic traffic information for all travelers more effectively as required within TRANSIMS. A more detailed study of the computational performance of the proposed algorithm and the heuristic methods can be conducted by varying the **network size** (in terms of the total number of nodes and links/arcs in the network), as well as the network density and structure. A relationship between the efficiency of the different methods and the size and structure of the network can be derived in order to prescribe a particular scheme for a given type of network. Another scope for future research is the development of more effective heuristic techniques that could adaptively identify routes that **avoid traffic congestion/bottleneck**. The presently proposed heuristic techniques are all based on some Euclidean geometrical or physical locations of the starting node and the terminal node within the planar network region. These methods do not exploit any available information on potential bottlenecks or high speed corridors, or effective transport modes (e.g. a high quality metro system) that might exist within the network region. We have provided some insights into this phenomenon and its incorporation within the algorithmic procedure for the case of method (iv) (Ellipsoidal Region Technique modified by including an inherent freeway system). Some further analysis along these lines would be greatly beneficial for use within TRANSIMS.