Online Supplement for
A study of the Lock-Free Tour Problem and Path-based Reformulations

Mehmet Başdere, Karen Smilowitz, Sanjay Mehrotra

OS.1 Formulation Comparisons on Related Tour Problems

This section repeats the numerical experiments in Section 6.1 for two additional problems to demonstrate the value of reformulations under less structured/more general settings. The first problem is a tour length minimization problem where the length budget constraints are removed. The aim is to find a minimum length tour which satisfies visit and locking requirements. In this setting, we compare formulations when the lower and upper bounds of length budget are not explicitly stated and come implicitly from the objective function. The second problem is a cost minimization problem where the additive coefficients of the objective function are assigned randomly and the aim is to find a minimum cost tour which satisfies visit and length requirements while ignoring locking. In this setting, we compare formulations when solving a generic tour finding problem with respect to various length budgets. These two problems are less structured compared to the LFTP and capture a range of tour finding problems with budget and visit requirements.

OS.1.1 Tour Length Minimization Problem.

In this problem, we remove constraints (1c) and (2g) from *LFTP-S* and reformulations, respectively. Each arc has a unit length and the aim is to find the minimum length tour which visits all edges in M. Table OS.1 and OS.2 provide the results for tour length minimization problem using the instances from Section 6.1. In addition, we repeat the experiments by removing the critical vertices (*Q0* versions) to analyze the effect of locking restrictions on model performance. Before further comparison, we must note that assessing the difficulty of instances for this problem is not trivial as some of the smaller instances turn out to be more difficult to solve than larger ones. In some cases, having more edges to visit can restrict the feasible region more making it more of a feasibility problem rather than an optimization problem. On the other hand, instances without locking restrictions are solved faster by all formulations compared to their counterparts with critical vertices.

Table OS.1: Tour length minimization results on instances with 12, 18 and 24 must-visit edges

Setting	**Formulation**	**Count**		**Duration (s)**		**Performance**		
		Sbtr	*VDLEI*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*
M12Q4	*LFTP-R2*	35	1	0	2	0.0	10	0
	LFTP-R3	29	0	0	2	0.0	10	0
	LFTP-S	720	14	14	89	0.0	10	0
M18Q6	*LFTP-R2*	213	7	2	28	0.0	10	0
	LFTP-R3	210	6	2	22	0.0	10	0
	LFTP-S	3350	54	129	1044	0.5	9	1
M24Q6	*LFTP-R2*	465	8	6	74	0.0	10	0
	LFTP-R3	410	8	5	54	0.0	10	0
	LFTP-S	1809	20	142	768	1.0	9	1
M12Q0	*LFTP-R2*	67	-	0	3	0.0	10	0
	LFTP-R3	20	-	0	2	0.0	10	0
	LFTP-S	635	-	4	27	0.0	10	0
M18Q0	*LFTP-R2*	138	-	1	14	0.0	10	0
	LFTP-R3	197	-	1	19	0.0	10	0
	LFTP-S	1586	-	28	727	0.3	9	1
M24Q0	*LFTP-R2*	254	-	2	33	0.0	10	0
	LFTP-R3	465	-	5	60	0.0	10	0
	LFTP-S	1833	-	47	758	0.2	9	1

Under both critical and noncritical settings, reformulations solve all the instances to optimality within the time limit of two hours whereas *LFTP-S* fails to prove optimality in 5 of 100 instances. Experiments in Table OS.1 show that there is not a significant difference between reformulations, favoring *LFTP-R2* due to its smaller formulation size. Comparing reformulations to *LFTP-S*, reformulations perform better than *LFTP-S*. Significant gaps in total solution times on instances with 18 and 24 must-visit edges are caused by the instances that are not solved to optimality by *LFTP-S*. In cases where *LFTP-S* solve all instances to optimality the solution time gap to reformulations is around two- to three-folds. In summary, these results suggest that reformulations are effective in reducing the solution efforts even in the absence of explicit length budget constraints.

OS.1.2 Cost Minimization Problem.

With the cost minimization objective, each arc has an objective contribution which is assigned randomly between 0 and 1. The aim is to find minimum cost tour while visiting all edges in M and satisfying the tour length restrictions. In this problem, we ignore locking restrictions to test the

Table OS.2: Tour length minimization results on instances with 30 and 36 must-visit edges

Setting	Formulation	Count		Duration (s)		Performance		
		Sbtr	*VDLEI*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*
M30Q8	*LFTP-R2*	423	22	10	200	0.0	10	0
	LFTP-S	2348	64	79	834	0.1	9	1
M36Q8	*LFTP-R2*	394	22	6	76	0.0	10	0
	LFTP-S	1127	32	45	197	0.0	10	0
M30Q0	*LFTP-R2*	600	-	10	126	0.0	10	0
	LFTP-S	1485	-	46	323	0.0	10	0
M36Q0	*LFTP-R2*	250	-	3	41	0.0	10	0
	LFTP-S	1300	-	23	142	0.0	10	0

effect of budget constraints on a more generic routing setting; therefore, the experiments are carried out on networks without critical vertices. The same tour length restrictions from Section 6.1 are used. Table OS.3 and OS.4 summarize the results. Since locking restrictions are not considered, VDLEIs are not used and the tables do not have the corresponding *VDLEI* column.

Table OS.3: Cost minimization results on instances with 12, 18 and 24 must-visit edges

Setting	Formulation	Count	Duration (s)		Performance		
		Sbtr	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*
M12Q0 [95% - 110%]	*LFTP-R2*	20	0	3	0.0	10	0
	LFTP-R3	20	0	3	0.0	10	0
	LFTP-S	266	2	11	0.0	10	0
M12Q0 [110% - 125%]	*LFTP-R2*	47	0	3	0.0	10	0
	LFTP-R3	53	0	4	0.0	10	0
	LFTP-S	223	2	14	0.0	10	0
M12Q0 [125% - 140%]	*LFTP-R2*	107	1	12	0.0	10	0
	LFTP-R3	102	1	9	0.0	10	0
	LFTP-S	357	3	20	0.0	10	0
M18Q0 [95% - 110%]	*LFTP-R2*	44	1	7	0.0	10	0
	LFTP-R3	36	1	8	0.0	10	0
	LFTP-S	247	2	11	0.0	10	0
M18Q0 [110% - 125%]	*LFTP-R2*	71	1	9	0.0	10	0
	LFTP-R3	65	1	9	0.0	10	0
	LFTP-S	245	2	12	0.0	10	0
M18Q0 [125% - 140%]	*LFTP-R2*	122	1	13	0.0	10	0
	LFTP-R3	170	1	19	0.0	10	0
	LFTP-S	234	2	9	0.0	10	0
M24Q0 [95% - 110%]	*LFTP-R2*	90	2	20	0.0	10	0
	LFTP-R3	116	2	39	0.0	10	0
	LFTP-S	502	9	56	0.0	10	0
M24Q0 [110% - 125%]	*LFTP-R2*	132	2	21	0.0	10	0
	LFTP-R3	207	2	46	0.0	10	0
	LFTP-S	996	18	206	0.0	10	0
M24Q0 [125% - 140%]	*LFTP-R2*	192	3	33	0.0	10	0
	LFTP-R3	340	5	92	0.0	10	0
	LFTP-S	1700	40	811	0.2	9	1

Compared to LFTP, the cost minimization problem is significantly easier to solve for two reasons: (i) we do not consider locking restrictions and more importantly, (ii) the cost structure is not clustered. Similar to the results in LFTP and tour length minimization problem, reformulations

Table OS.4: Cost minimization results on instances with 30 and 36 must-visit edges

Setting	Formulation	Count	Duration (s)		Performance		
		Sbtr	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*
M30Q0	*LFTP-R2*	135	7	88	0	10	0
[95% - 110%]	*LFTP-S*	598	19	110	0	10	0
M30Q0	*LFTP-R2*	184	4	46	0	10	0
[110% - 125%]	*LFTP-S*	435	6	27	0	10	0
M30Q0	*LFTP-R2*	170	3	48	0	10	0
[125% - 140%]	*LFTP-S*	1344	28	186	0	10	0
M36Q0	*LFTP-R2*	136	4	61	0	10	0
[95% - 110%]	*LFTP-S*	441	13	62	0	10	0
M36Q0	*LFTP-R2*	190	4	73	0	10	0
[110% - 125%]	*LFTP-S*	847	35	267	0	10	0
M36Q0	*LFTP-R2*	423	22	383	0	10	0
[125% - 140%]	*LFTP-S*	1120	68	647	0	10	0

outperform *LFTP-S* in the number of subtour elimination inequalities needed and in solution times; however, the performance gap is not as significant. All formulations find the optimal solutions within the time limit for all instances except one instance in high budget *M24Q0* runs where *LFTP-S* fails to prove optimality. Between reformulations, *LFTP-R2* performs slightly better than *LFTP-R3*. As the length budget increases, the difference between solution times of *LFTP-S* and reformulations decreases; however, reformulations are still better compared to *LFTP-S*.

The results of these additional experiments indicate that reformulations effectively reduce solution times and subtour formations when compared to *LFTP-S* under different problem settings which is promising as these settings cover a broad range of tour finding variants with length budget and locking restrictions.

OS.2 Case Study: Compactness Objective

This section provides a brief discussion on the compactness objective where the aim is to maximize compactness (or equivalently minimize the area covered by the resulting tour) and uses a proxy objective function to design compact routes for BACM. Maximizing compactness of a marathon course is a crucial objective from two perspectives: (i) the resulting tour spans a smaller area, making it easier to manage for the organizers and (ii) locks a smaller portion of the network increasing overall accessibility. Considering the formulations proposed throughout the paper, a straightforward way to solve this problem is to represent each city block with a critical vertex and assign a penalty coefficient for each vertex that is blocked by the resulting tour. Penalty coefficient for each critical vertex can be set to the size or the population of the block that it represents. However, adding vertices for each block increases the size of the underlying network significantly, making the resulting problem difficult to solve. For this reason, we introduce a proxy compactness objective which aims to minimize the weighted distance from the arcs of the tour to the hypothetical line that passes through the start-finish line of the race. Similar to medical distance minimization objective, the weights come from the length of the arc.

We use *LFTP-R2* to solve the course design problem with a compactness objective for similarity levels from 100% to 0% as in Section 6.2. The detailed results of the experiments are not reported here as all the instances are solved within ~ 30 seconds. Instead, we provide the courses obtained at different similarity levels in Figure OS.4 along with their improvement measured by the proxy compactness objective. The resulting courses indicate that proxy compactness objective works well in limiting the area covered by the resulting tour. Similar to the medical distance objective, notable improvements can be made while keeping a large portion of the course the same.

(a) Current course and hypothetical line (b) 70% similarity 31.9% improvement. (c) 50% similarity 37.6% improvement. (d) $0-40\%$ similarity 38.0% improvement.

Figure OS.4: Tours with different similarity requirements.