Submitted to *INFORMS Journal on Computing*
manuscript JOC-2013-10-OA-191

Dynamic Programming Driven Memetic Search for the Steiner Tree Problem with Revenues, Budget and Hop Constraints - Online Supplement

Zhang-Hua Fu, Jin-Kao Hao* (Corresponding author)
LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers Cedex 01, France
{fu,hao}@info.univ-angers.fr

---

1. Worst Case Complexity Analysis

We provide here a worst case complexity analysis of the proposed memetic algorithm. The algorithm consists of several main ingredients, including a pre-processing procedure, a probabilistic constructive procedure, a dynamic programming based neighborhood search (NS) procedure, a crossover operator associated with a pool updating strategy, whose computational complexities are respectively bounded as follows.

1. *Pre-processing procedure* (15): The pre-processing procedure is used to calculate and store all the possible values of $L(i,j,l), i,j \in V, r_j > 0, 1 \leq l \leq H$ before the search (which would be fetched directly during the search, instead of calculating them repeatedly). Using a dynamic programming procedure (20), these computations can be achieved within a time of $O(|V| \times |PV| \times H) \leq O(|V|^2 \times H)$, where PV denotes the set containing all the profitable vertices (with $r_j > 0$) of graph G.

2. *Probabilistic constructive procedure* (15): Using the needed $L(i,j,l)$ values provided by the pre-processing step, the computations at each construction step can be finished within a time of $O(|V| \times |PV| + |V| \times H)$. Since at most $|PV|$ steps are needed to complete the

construction procedure, the total time for constructing a saturated BHS-tree is bounded within $O(|V| \times |PV|^2 + |PV| \times |V| \times H) \le O(|V|^3 + |V|^2 \times H)$.

3. *Neighborhood search (NS) procedure*: Due to the possibility of special cases and the unknown number of NS iterations before termination, it is difficult to give an exact computational complexity of the NS procedure, thus we just provide a worst case analysis as follows. At first, at the beginning of each iteration of NS, we should re-execute the dynamic programming procedure of Eq. (2), whose complexity is bounded within $O(|V_{urp}| \times W_{max}) \le O(|PV| \times B) \le O(|V| \times B)$. Then, up to $O(|lv(T)|^2) \le O(|PV|^2) \le O(|V|^2)$ possible neighboring solutions should be examined by the estimation criterion (each one needs $O(H)$ time to get the available budget after deletion and $O(1)$ time to check if the examined neighboring solution could be directly discarded), among which only the hopeful ones identified by the estimation criterion should be actually generated. As explained in subsection 3.2.4, with the aid of the dynamic driven estimation criterion, in most cases only one or very few neighboring solutions should be actually generated. However, in the worst case (i.e., special case 1 occurs for every neighboring solution), it is possible that all the $O(|lv(T)|^2) \le O(|V|^2)$ neighboring solutions should be actually generated. Now we discuss the time needed for actually generating a neighboring solution. At first, for preparation, we should store (and restore afterwards) the incumbent solution, both within $O(|V|)$ time. Then, the path deletion can be done within $O(H)$ time. After that, up to $|V_{urp}|$ new paths may be inserted, each one with $O(|V|)$ time to re-calculate a new hop-constrained shortest path between the chosen vertex for connection and the solution after path deletion, and up to $O(|V| \times H)$ time to insert the renewed hop-constrained shortest path, using the pre-calculated values of $L(i,j,l)$. Thus the time complexity of actually generating a neighboring solution is bounded within $O(|V|) + O(H) + O(|V_{urp}|) \times O(|V|) + O(|V_{urp}|) \times O(|V| \times H) \le O(|V|^2 \times H)$. Based on the above information, the complexity of each iteration of NS is bounded within $O(|V| \times B) + O(|lv(T)|^2) \times O(H) + O(|lv(T)|^2) \times O(|V|^2 \times H) \le O(|V| \times B + |V|^4 \times H)$. Given that it is difficult to exactly estimate the number of iterations before termination. Let I denote the maximum possible number of NS iterations (I is typically but unnecessarily in proposition to $|V|$), then the overall complexity of the NS procedure is bounded by $I \times O(|V| \times B + |V|^4 \times H)$.

4. *Crossover operator*: At first, the shared backbone of any two solutions could be identified (first step of the crossover operator) within a complexity of $O(|PV| \times H) \le O(|V| \times H)$.

Subsequently, to complete the backbone and obtain a saturated BHS-tree, a complexity bounded by the probabilistic constructive procedure is needed, i.e., $O(|V|^3 + |V|^2 \times H)$, as analyzed above. Thus, the crossover operator can be achieved within $O(|V| \times H) + O(|V|^3 + |V|^2 \times H) = O(|V|^3 + |V|^2 \times H)$ time.

5. *Pool updating strategy*: Basically, the distance between any two solutions can be calculated within $O(|V|)$ time. Based on this, for the sake of a quick updating, we maintain in a $Q \times Q$ matrix the distances between any pair of solutions belonging to Pop (requiring a complexity of $O(|V| \times Q^2)$ for initialization). With this distance matrix, given a new offspring solution T^0 and the population Pop containing Q solutions, the goodness score of each solution of Pop can be calculated within $O(|V| + Q)$ time, and the goodness score of T^0 can be calculated within $O(|V| \times Q)$ time, meaning that the worst (with the lowest goodness score) solution of $Pop \cup \{T^0\}$ can be identified within a complexity of $O(Q) \times O(|V| + Q) + O(|V| \times Q) = O(|V| \times Q + Q^2)$. Finally, after replacing T^0 with some other existing solution, the distances matrix is updated within a complexity of $O(|V| \times Q)$. Overall, updating the solutions pool with a new offspring is done within a time of $O(|V| \times Q + Q^2) + O(|V| \times Q) = O(|V| \times Q + Q^2)$ (excluding the complexity needed for distances matrix initialization).

To summarize, the complexity of each round of the NS procedure followed by crossover and pool updating is bounded within $I \times O(|V| \times B + |V|^4 \times H) + O(|V|^3 + |V|^2 \times H) + O(|V| \times Q + Q^2) = O(I \times |V| \times B + I \times |V|^4 \times H + |V| \times Q + Q^2)$. Give that it is difficult to estimate the number of such rounds needed for a run of the memetic algorithm. If denote the maximum number of such rounds by G (G is typically but unnecessarily in proposition to M), the overall complexity (including the pre-processing procedure, the solutions pool initialization, as well as the distance matrix initialization) of each independent run of memetic is bounded within $O(|V|^2 \times H) + Q \times O(|V|^3 + |V|^2 \times H) + O(|V| \times Q^2) + G \times O(I \times |V| \times B + I \times |V|^4 \times H + |V| \times Q + Q^2)$. Again, we would like to point out this is only a worst case analysis which serves as a theoretical upper bound of computational complexity, being much higher than the actually needed computational efforts in most cases.

2. Supplementary results and additional comparisons

Table 1 Results of the memetic algorithm with a reduced termination criterion using $M = 50$ (Memetic(50)) on the 56 most challenging instances of group G4, in comparison with previous fast heuristics D&R and TS(2000)

Instance			D&R		TS(2000)		Memetic(50)		Instance			D&R		TS(2000)		Memetic(50)	
Graph	b	H	R^{best}	$t(s)$	R^{best}	$t(s)$	R^{best}	$t(s)$	*Graph*	b	H	R^{best}	$t(s)$	R^{best}	$t(s)$	R^{best}	$t(s)$
C8_10	20	5	208	1.32	≤228	14.2	*228*	3.02	C8_10	50	5	104	0.12	≤**116**	13.6	*116*	2.14
C8_10	20	15	303	9.51	≤308	42.0	**324**	6.86	C8_10	50	15	152	1.91	≤161	41.6	**171**	5.16
C8_10	20	25	311	16.23	≤310	65.6	**329**	8.87	C8_10	50	25	151	3.19	≤160	63.7	**172**	7.96
C8_100	20	5	2261	1.28	≤2365	13.5	*2365*	4.03	C8_100	50	5	1151	0.23	≤1204	13.5	**1216**	2.61
C8_100	20	15	3269	9.24	≤3237	43.1	**3370**	7.13	C8_100	50	15	1696	1.88	≤1675	41.8	**1750**	5.71
C8_100	20	25	3310	16.87	≤3337	66.8	**3416**	9.74	C8_100	50	25	1735	1.79	≤1742	64.3	**1792**	8.02
C9_10	20	5	274	2.26	≤279	16.0	**302**	5.60	C9_10	50	5	143	0.26	≤145	16.1	**149**	3.67
C9_10	20	15	354	11.35	≤349	42.4	**372**	9.01	C9_10	50	15	172	2.36	≤165	39.3	**184**	8.11
C9_10	20	25	353	9.45	353	63.0	**375**	12.81	C9_10	50	25	168	3.70	≤167	59.1	**185**	11.66
C9_100	20	5	2864	2.24	≤2933	16.4	**3112**	6.31	C9_100	50	5	1514	0.50	≤1509	15.8	**1563**	3.88
C9_100	20	15	3642	11.15	≤3588	41.5	**3853**	10.59	C9_100	50	15	1675	2.13	≤1742	38.6	**1878**	8.13
C9_100	20	25	3602	17.26	≤3644	62.1	**3878**	14.08	C9_100	50	25	1674	3.35	≤1745	59.4	**1905**	12.52
C10_10	20	5	341	2.07	≤371	14.4	**385**	7.73	C10_10	50	5	156	0.39	≤173	13.9	**184**	5.71
C10_10	20	15	505	13.07	≤505	41.4	**540**	19.66	C10_10	50	15	211	2.54	≤221	40.4	**247**	15.95
C10_10	20	25	482	19.86	≤501	67.2	**550**	27.51	C10_10	50	25	219	3.83	≤218	62.9	**250**	23.49
C10_100	20	5	3530	1.88	≤3811	15.5	**4055**	8.89	C10_100	50	5	1513	0.32	≤1836	14.0	**1937**	6.27
C10_100	20	15	5163	13.30	≤5227	45.5	**5621**	21.35	C10_100	50	15	2415	2.88	≤2365	39.7	**2550**	15.69
C10_100	20	25	5287	22.97	≤5286	64.9	**5703**	29.11	C10_100	50	25	2472	4.66	≤2432	63.8	**2586**	24.18
C13_10	100	5	242	2.67	≤243	26.6	**251**	3.40	C13_10	100	15	303	12.57	≤296	72.8	**311**	6.22
C13_10	100	25	302	22.19	≤298	107.3	**311**	8.62	C13_100	100	5	2507	2.48	≤2526	25.9	**2599**	3.90
C13_100	100	15	3064	11.64	≤3029	70.4	**3244**	6.95	C13_100	100	25	3064	22.72	≤3057	109.8	**3251**	9.41
C14_10	100	5	344	5.33	339	26.8	**368**	5.20	C14_10	100	15	377	17.72	≤370	71.4	**392**	9.32
C14_10	100	25	377	28.02	369	101.8	**391**	12.73	C14_100	100	5	3485	5.02	≤3416	26.3	**3843**	6.61
C14_100	100	15	3846	17.32	≤3872	71.5	**4110**	11.50	C14_100	100	25	3846	27.43	≤3856	102.2	**4107**	14.13
C15_10	20	5	1174	79.48	≤1192	28.9	**1218**	30.08	C15_10	100	5	401	4.94	≤427	26.2	**450**	10.55
C15_10	100	15	515	23.48	≤501	71.0	**558**	18.11	C15_10	100	25	520	38.60	510	107.7	**559**	25.74
C15_100	20	5	12078	83.31	≤12198	28.5	**12353**	40.13	C15_100	100	5	4180	5.17	≤4358	27.1	**4741**	11.11
C15_100	100	15	5355	24.04	≤5177	70.8	**5776**	21.98	C15_100	100	25	5393	41.07	5243	106.7	**5792**	27.41

Table 2 Comparison between Memetic/DP and the memetic algorithm for solving the 56 most challenging cases of group G4

Instance			Memetic/DP			Memetic			Instance			Memetic/DP			Memetic		
Graph	*b*	*H*	R^{best}	R^{avg}	$t(s)$	R^{best}	R^{avg}	$t(s)$	*Graph*	*b*	*H*	R^{best}	R^{avg}	$t(s)$	R^{best}	R^{avg}	$t(s)$
C8_10	20	5	*230*	229.4	446.53	*230*	230.0	19.06	C8_10	50	5	*116*	116.0	69.99	*116*	116.0	10.80
C8_10	20	15	328	326.4	539.48	**330**	327.8	42.37	C8_10	50	15	*171*	168.1	70.07	*171*	168.5	23.03
C8_10	20	25	**331**	329.3	548.89	330	329.0	35.60	C8_10	50	25	*172*	171.6	83.17	*172*	171.6	31.50
C8_100	20	5	*2380*	2367.7	525.52	*2380*	2374.0	24.37	C8_100	50	5	*1216*	1212.0	92.18	*1216*	1216.0	12.30
C8_100	20	15	3412	3384.6	636.87	**3418**	3407.6	47.39	C8_100	50	15	*1774*	1757.7	109.59	*1774*	1759.6	26.46
C8_100	20	25	3429	3413.9	620.59	**3443**	3426.5	48.16	C8_100	50	25	*1792*	1784.8	103.36	*1792*	1784.2	33.34
C9_10	20	5	301	298.4	599.20	**302**	299.3	29.31	C9_10	50	5	*149*	148.9	73.95	*149*	149.0	16.04
C9_10	20	15	374	372.5	406.81	**376**	372.4	42.45	C9_10	50	15	**184**	182.1	107.67	183	180.8	34.77
C9_10	20	25	**380**	374.9	444.21	377	374.7	54.65	C9_10	50	25	*186*	185.8	108.47	*186*	184.8	43.77
C9_100	20	5	*3112*	3088.2	845.56	*3112*	3092.8	38.16	C9_100	50	5	*1563*	1563.0	86.07	*1563*	1563.0	17.57
C9_100	20	15	**3875**	3860.6	820.02	3873	3854.5	56.49	C9_100	50	15	**1899**	1871.5	121.28	1879	1863.5	36.32
C9_100	20	25	**3912**	3887.6	621.32	3906	3881.4	71.83	C9_100	50	25	**1918**	1906.9	158.94	1909	1898.6	50.11
C10_10	20	5	*388*	387.3	1009.09	*388*	387.4	57.24	C10_10	50	5	*185*	184.3	142.94	*185*	184.1	26.66
C10_10	20	15	551	544.9	1557.19	**553**	545.2	113.51	C10_10	50	15	*247*	247.0	175.81	*247*	247.0	61.70
C10_10	20	25	558	553.7	1587.71	**561**	555.4	133.69	C10_10	50	25	**256**	250.6	210.59	254	249.6	78.51
C10_100	20	5	**4088**	4067.3	1402.74	4069	4047.9	67.42	C10_100	50	5	*1940*	1939.4	149.69	*1940*	1939.3	34.83
C10_100	20	15	5682	5620.9	2133.47	**5686**	5620.0	130.25	C10_100	50	15	2566	2551.6	233.34	**2601**	2552.1	71.46
C10_100	20	25	**5838**	5738.1	1914.11	5773	5694.6	139.72	C10_100	50	25	2586	2579.5	260.60	**2632**	2576.1	92.41
C13_10	100	5	**254**	251.8	380.54	253	251.1	22.97	C13_10	100	15	312	310.5	266.70	**316**	312.0	33.71
C13_10	100	25	*314*	311.9	342.20	*314*	311.6	37.38	C13_100	100	5	2609	2590.4	450.91	**2622**	2593.0	29.87
C13_100	100	15	3253	3242.3	395.94	**3255**	3242.7	41.05	C13_100	100	25	3257	3245.6	460.48	**3260**	3241.2	42.62
C14_10	100	5	368	365.5	810.57	**370**	366.0	29.95	C14_10	100	15	396	393.4	889.66	**398**	394.7	55.08
C14_10	100	25	396	393.2	928.70	**397**	393.8	53.99	C14_100	100	5	3843	3827.6	1032.97	**3847**	3823.7	36.42
C14_100	100	15	4156	4127.2	1155.39	**4172**	4150.2	72.46	C14_100	100	25	**4161**	4126.2	1199.43	4150	4121.6	76.56
C15_10	20	5	1218	1214.7	7214.87	**1221**	1218.2	188.94	C15_10	100	5	458	453.6	1834.20	**462**	456.0	71.42
C15_10	100	15	**559**	557.4	941.18	558	557.5	92.59	C15_10	100	25	*558*	558.0	1131.41	*558*	556.4	99.20
C15_100	20	5	12370	12341.1	7218.16	**12392**	12366.5	359.13	C15_100	100	5	4779	4668.5	2051.51	**4807**	4758.0	97.01
C15_100	100	15	*5807*	5786.7	1384.34	*5807*	5793.8	109.24	C15_100	100	25	5807	5788.1	1481.98	**5822**	5795.7	127.45

Table 3 Comparison between ILS and the memetic algorithm for solving the 56 most challenging cases of group G4

Instance			ILS			Memetic			Instance			ILS			Memetic		
Graph	*b*	*H*	R^{best}	R^{avg}	$t(s)$	R^{best}	R^{avg}	$t(s)$	*Graph*	*b*	*H*	R^{best}	R^{avg}	$t(s)$	R^{best}	R^{avg}	$t(s)$
C8_10	20	5	*230*	228.4	15.64	*230*	230.0	19.06	C8_10	50	5	*116*	115.8	8.89	*116*	116.0	10.80
C8_10	20	15	327	325.1	25.67	**330**	327.8	42.37	C8_10	50	15	168	167.5	12.77	**171**	168.5	23.03
C8_10	20	25	*330*	327.8	28.60	*330*	329.0	35.60	C8_10	50	25	*172*	171.2	15.36	*172*	171.6	31.50
C8_100	20	5	2373	2349.1	20.29	**2380**	2374.0	24.37	C8_100	50	5	*1216*	1206.0	10.67	*1216*	1216.0	12.30
C8_100	20	15	3414	3375.4	34.18	**3418**	3407.6	47.39	C8_100	50	15	1764	1750.9	16.28	**1774**	1759.6	26.46
C8_100	20	25	3414	3403.4	33.54	**3443**	3426.5	48.16	C8_100	50	25	*1792*	1783.4	18.10	*1792*	1784.2	33.34
C9_10	20	5	301	297.1	27.39	**302**	299.3	29.31	C9_10	50	5	*149*	149.0	11.88	*149*	149.0	16.04
C9_10	20	15	**377**	373.4	30.88	376	372.4	42.45	C9_10	50	15	**184**	182.7	21.34	183	180.8	34.77
C9_10	20	25	**379**	376.5	34.32	377	374.7	54.65	C9_10	50	25	*186*	184.1	24.66	*186*	184.8	43.77
C9_100	20	5	*3112*	3089.3	36.65	*3112*	3092.8	38.16	C9_100	50	5	*1563*	1563.0	11.52	*1563*	1563.0	17.57
C9_100	20	15	**3918**	3882.3	34.91	3873	3854.5	56.49	C9_100	50	15	**1888**	1875.0	21.44	1879	1863.5	36.32
C9_100	20	25	**3932**	3889.2	41.76	3906	3881.4	71.83	C9_100	50	25	1895	1885.8	27.73	**1909**	1898.6	50.11
C10_10	20	5	386	384.9	41.42	**388**	387.4	57.24	C10_10	50	5	184	184.0	17.65	**185**	184.1	26.66
C10_10	20	15	545	540.7	58.81	**553**	545.2	113.51	C10_10	50	15	*247*	247.0	30.27	*247*	247.0	61.70
C10_10	20	25	553	547.4	75.24	**561**	555.4	133.69	C10_10	50	25	251	249.3	43.51	**254**	249.6	78.51
C10_100	20	5	**4079**	4043.7	62.36	4069	4047.9	67.42	C10_100	50	5	1939	1938.8	22.59	**1940**	1939.3	34.83
C10_100	20	15	5662	5584.1	83.03	**5686**	5620.0	130.25	C10_100	50	15	2550	2542.1	37.42	**2601**	2552.1	71.46
C10_100	20	25	5743	5658.9	96.00	**5773**	5694.6	139.72	C10_100	50	25	2587	2569.9	50.27	**2632**	2576.1	92.41
C13_10	100	5	252	249.7	19.11	**253**	251.1	22.97	C13_10	100	15	313	311.1	18.62	**316**	312.0	33.71
C13_10	100	25	313	311.3	20.85	**314**	311.6	37.38	C13_100	100	5	2609	2584.6	26.62	**2622**	2593.0	29.87
C13_100	100	15	3243	3238.2	28.45	**3255**	3242.7	41.05	C13_100	100	25	*3260*	3241.7	31.29	*3260*	3241.2	42.62
C14_10	100	5	368	365.6	29.62	**370**	366.0	29.95	C14_10	100	15	396	391.2	36.06	**398**	394.7	55.08
C14_10	100	25	393	391.4	37.37	**397**	393.8	53.99	C14_100	100	5	3811	3803.3	44.91	**3847**	3823.7	36.42
C14_100	100	15	4151	4108.1	54.37	**4172**	4150.2	72.46	C14_100	100	25	4136	4103.8	49.72	**4150**	4121.6	76.56
C15_10	20	5	1210	1206.5	292.71	**1221**	1218.2	188.94	C15_10	100	5	457	452.4	45.15	**462**	456.0	71.42
C15_10	100	15	**559**	556.9	59.97	558	557.5	92.59	C15_10	100	25	*558*	557.3	64.92	*558*	556.4	99.20
C15_100	20	5	12266	12253.1	378.12	**12392**	12366.5	359.13	C15_100	100	5	4741	4723.8	68.79	**4807**	4758.0	97.01
C15_100	100	15	5792	5773.3	67.00	**5807**	5793.8	109.24	C15_100	100	25	5792	5779.9	70.66	**5822**	5795.7	127.45