THE UNIVERSITY of ADELAIDE

Optimisation and Logistics
School of Computer Science
The University of Adelaide, Australia

Exact Approaches for the Travelling Thief Problem

Junhua Wu, Markus Wagner, Sergey Polyakovskiy, and Frank Neumann

Motivation

Many evolutionary and constructive heuristic approaches have been introduced in order to solve the Traveling Thief Problem (TTP). However, the accuracy of such approaches is unknown due to their inability to find global optima. We propose three exact algorithms to the TTP. We compare these with the state-of-the-art heuristic approaches to gather a comprehensive overview on the accuracy of heuristic methods for solving small TTP instances.

Travelling Thief Problem

The TTP is a combination of travelling salesman problem (TSP) and 0-1 knapsack problem (KP).

$$Z([\Pi,P]) = \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{m_i} p_{ik}y_{ik}}_{\text{Profits}} - \underbrace{R\left(\frac{d_{x_n x_1}}{\upsilon_{max}-\nu W_{x_n}} + \sum_{i=1}^{n-1}\frac{d_{x_i x_{i+1}}}{\upsilon_{max}-\nu W_{x_i}}\right)}_{\text{Travelling Cost}}$$

Π is a tour for the TSP.

P is a packing plan for the KP.

Dynamic Programming

The DP to the TTP is a combination of Held-Karp algorithm for the TSP and the dynamic programming to the PWT problem[1].

Algorithm 1 Dynamic programming to the TTP

set $A(\{1\},1,0)=0$
for $w=1$ to C **do**
  set $A(\{1\},1,w)=-\infty$
for $s=2$ to n **do**
  for any $S\subseteq N: |S|=s,\ 1\in S$ **do**
    for $w=0$ to C **do**
      set $A(S,1,w)=-\infty$
      for any $j\in S,\ j\neq 1$ **do**
        compute $A(S,j,w)=$
        $\max_{i\in S:i\neq j}\left\{A(S\setminus\{j\},i,w-\overline{W_j}(S\setminus\{j\},i))+\overline{P_j}(S\setminus\{j\},i)-\frac{d_{ij}}{\upsilon_{max}-\nu w}\right\}$
return $\max_{i\in S:i\neq 1}\left\{A(N,i,w)-\frac{d_{i1}}{\upsilon_{max}-\nu w}\right\}$

Branch and Bound Search

We propose the upper bound that calculates the maximal possible profit that the thief may obtain by passing the remaining part of the tour with the minimal possible cost.

$$E_U(A(S,j,\cdot)) = \max_{0\le w\le W} A(S,j,w) + \sum_{k\in N\setminus S}\sum_{l=1}^{m_k} p_{kl} - R\frac{d_{j1}}{\upsilon_{max}}$$

Constraint Programming

Our constraint programming model employs a simple permutation based representation of the tour that allows the use of the AllDifferent[2] filtering algorithm.

$$max \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{ij}y_{ij}$$

$$-R\left(\sum_{i=1}^{n-1}\frac{\texttt{Element}(d, n(x_i-1)+x_{i+1})}{\upsilon_{max}-\nu\texttt{Element}(W,x_i)} + \frac{\texttt{Element}(d,n(x_n-1)+1)}{\upsilon_{max}-\nu\texttt{Element}(W,x_n)}\right)$$

$$\texttt{AllDifferent}[x_1,\ldots,x_n]$$

$$W_i = W_{i-1} + \sum_{j\in M_i} w_{ij}y_{ij},\ i\in\{2,\ldots,n\}$$

$$W_n \le C$$

Experiments

Comparison of the exact approaches.

Instance	n	m	Running time (in seconds) DP	BnB	CP
eil51_n05_m4_uncorr_01	5	4	0.018	0.023	0.222
eil51_n06_m5_uncorr_01	6	5	0.07	0.079	0.24
eil51_n07_m6_uncorr_01	7	6	0.143	0.195	0.497
eil51_n08_m7_uncorr_01	8	7	0.343	0.505	4.594
eil51_n09_m8_uncorr_01	9	8	0.633	1.492	63.838
eil51_n10_m9_uncorr_01	10	9	0.933	5.188	776.55
eil51_n11_m10_uncorr_01	11	10	2.414	23.106	12861.181
eil51_n12_m11_uncorr_01	12	11	3.938	204.786	-
eil51_n13_m12_uncorr_01	13	12	14.217	2007.074	-
eil51_n14_m13_uncorr_01	14	13	13.408	36944.146	-
eil51_n15_m14_uncorr_01	15	14	89.461	-	-
eil51_n16_m15_uncorr_01	16	15	59.526	-	-
eil51_n17_m16_uncorr_01	17	16	134.905	-	-
eil51_n18_m17_uncorr_01	18	17	366.082	-	-
eil51_n19_m18_uncorr_01	19	18	830.18	-	-
eil51_n20_m19_uncorr_01	20	19	2456.873	-	-

Comparison between DP and the heuristics.

gap	MA2B	CS2B	CS2SA	S1	S5	C5	DP-S1	DP-S5
avg	0.3%	15.3%	11.5%	38.9%	15.7%	09.9%	30.1%	3.3%
stdev	2.2%	17.8%	16.7%	29.4%	24.6%	18.8%	20.1%	8.5%

Instance	TTP-DP OPT	RT	MA2B Gap	Std	RT	C5 Gap	Std	DP-S5 Gap	Std
eil51_n05_m4_multiple-strongly-corr_01	619.227	0.02	29.1	12.1	2.71	35.5	1.20e-6	41.3	0.0
eil51_n05_m4_uncorr_01	466.929	0.02	0.0	0.0	3.22	0.0	2.20e-6	0.0	2.20e-6
eil51_n05_m4_uncorr-similar-weights_01	299.281	0.02	0.0	0.0	3.21	7.8	2.40e-6	7.8	1.20e-6
eil51_n05_m20_multiple-strongly-corr_01	773.573	0.08	13.4	0.0	1.44	14.3	0.0	12.8	0.0
eil51_n05_m20_uncorr_01	2144.796	0.07	0.0	0.0	3.35	7.4	0.0	6.6	2.30e-6
eil51_n05_m20_uncorr-similar-weights_01	269.015	0.04	0.0	0.0	3.51	0.0	2.30e-6	0.0	0.0
eil51_n10_m9_multiple-strongly-corr_01	573.897	1.21	0.0	0.0	6.07	0.0	0.0	0.0	0.0
eil51_n10_m9_uncorr_01	1125.715	0.93	0.0	0.0	6.06	0.0	1.30e-6	0.0	1.30e-6
eil51_n10_m9_uncorr-similar-weights_01	753.230	0.86	0.0	0.0	5.87	0.0	0.0	0.0	0.0
eil51_n10_m45_multiple-strongly-corr_01	1091.127	14.89	0.0	0.0	7.99	0.0	0.0	0.0	0.0
eil51_n10_m45_uncorr_01	6009.431	6.39	0.0	0.0	8.6	6.6	2.30e-6	0.0	0.0
eil51_n10_m45_uncorr-similar-weights_01	3009.553	8.87	0.0	0.0	6.78	0.0	2.30e-6	0.0	2.30e-6
eil51_n12_m11_multiple-strongly-corr_01	648.546	4.58	0.0	0.0	6.08	4.6	2.20e-6	4.6	2.20e-6
eil51_n12_m11_uncorr_01	1717.699	3.94	0.0	0.0	7.21	0.0	1.20e-6	0.0	1.20e-6
eil51_n12_m11_uncorr-similar-weights_01	774.107	3.36	0.0	0.0	7.03	0.0	2.30e-6	0.0	2.30e-6
eil51_n12_m55_multiple-strongly-corr_01	1251.780	117.99	0.0	0.0	9.19	0.0	0.0	0.0	0.0
eil51_n12_m55_uncorr_01	8838.012	35.79	0.0	0.0	9.76	0.0	0.0	0.0	0.0
eil51_n12_m55_uncorr-similar-weights_01	3734.895	38.36	12.3	0.0	8.34	12.3	0.0	0.2	0.0
eil51_n15_m14_multiple-strongly-corr_01	547.419	39.82	0.0	0.0	7.87	14.1	1.30e-6	13.3	1.30e-6
eil51_n15_m14_uncorr_01	2392.996	89.46	0.0	0.0	7.28	3.8	0.0	3.8	0.0
eil51_n15_m14_uncorr-similar-weights_01	637.419	16.35	0.0	0.0	6.86	0.0	1.60e-6	0.0	1.60e-6
eil51_n15_m70_multiple-strongly-corr_01	920.372	3984.29	2.1	1.1	12.11	0.0	2.70e-6	0.0	2.70e-6
eil51_n15_m70_uncorr_01	9922.137	740.22	0.0	0.0	9.67	7	1.20e-6	1.9	0.0
eil51_n15_m70_uncorr-similar-weights_01	4659.623	867.78	0.0	0.0	7.98	0.0	0.0	0.0	0.0
eil51_n16_m15_multiple-strongly-corr_01	794.745	105.5	0.0	0.0	7.7	18.9	1.6e-6	18.9	1.6e-6
eil51_n16_m15_multiple-strongly-corr_10	4498.848	623.4	0.0	0.0	9.1	12.9	0.0	16.6	1.3e-6
eil51_n16_m15_uncorr_01	2490.889	59.5	1.0	0.7	8.4	1.6	2.3e-6	1.6	2.3e-6
eil51_n16_m15_uncorr_10	3601.077	211.5	0.0	0.0	9.0	7.1	1.6e-6	7.1	1.6e-6
eil51_n16_m15_uncorr-similar-weights_01	540.897	36.4	0.0	0.0	8.5	0.0	3.0e-6	0.0	3.0e-6
eil51_n16_m15_uncorr-similar-weights_10	3948.211	245.4	0.0	0.0	8.7	5.8	1.5e-6	13.6	0.0
eil51_n17_m16_multiple-strongly-corr_01	685.565	248.6	0.0	0.0	8.4	0.2	1.5e-6	0.0	1.5e-6
eil51_n17_m16_multiple-strongly-corr_10	3826.098	2190.4	0.0	0.0	9.8	0.0	1.5e-6	0.0	1.5e-6
eil51_n17_m16_uncorr_01	2342.664	134.9	0.0	0.0	8.3	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr_10	2275.279	554.5	0.0	0.0	9.6	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr-similar-weights_01	556.851	70.8	0.0	0.0	8.1	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr-similar-weights_10	2935.961	787.7	0.0	0.0	9.7	0.0	0.0	0.0	0.0
eil51_n18_m17_multiple-strongly-corr_01	834.031	715.7	7.9	0.8	10.2	9.2	0.0	12.9	1.7e-6
eil51_n18_m17_multiple-strongly-corr_10	5531.373	6252.4	0.0	0.0	10.5	0.4	1.5e-6	0.4	1.5e-6
eil51_n18_m17_uncorr_01	2644.491	366.1	0.0	0.0	9.7	0.2	0.0	1.8	0.0
eil51_n18_m17_uncorr_10	3222.603	1462.7	0.0	0.0	10.3	0.0	1.3e-6	0.2	0.0
eil51_n18_m17_uncorr-similar-weights_01	532.906	148.3	0.0	0.0	8.5	0.0	1.3e-6	0.0	1.3e-6
eil51_n18_m17_uncorr-similar-weights_10	4420.438	1929.3	0.0	0.0	9.9	0.0	2.9e-6	0.3	1.8e-6
eil51_n19_m18_multiple-strongly-corr_01	910.229	1771.6	0.0	0.0	9.3	20.1	1.6e-6	20.1	1.6e-6
eil51_n19_m18_multiple-strongly-corr_10	-	-	-	-	10.4	-	-	-	-
eil51_n19_m18_uncorr_01	2604.844	830.2	0.0	0.0	9.7	0.0	0.0	0.0	0.0
eil51_n19_m18_uncorr_10	4048.408	3884.3	0.0	0.0	10.9	0.0	1.4e-6	0.0	1.4e-6
eil51_n19_m18_uncorr-similar-weights_01	472.186	412.3	0.0	0.0	9.2	0.0	1.5e-6	0.0	1.5e-6
eil51_n19_m18_uncorr-similar-weights_10	5573.695	5878.8	0.0	0.0	10.5	0.0	0.0	0.0	0.0
eil51_n20_m19_multiple-strongly-corr_01	518.189	4533.7	0.6	0.6	11.1	14.1	1.4e-6	12.3	0.0
eil51_n20_m19_multiple-strongly-corr_10	-	-	-	-	12.1	-	-	-	-
eil51_n20_m19_uncorr_01	2092.673	2456.9	0.0	0.0	8.7	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr_10	3044.391	12776.0	0.0	0.0	9.8	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr-similar-weights_01	451.052	1007.7	0.0	0.0	7.9	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr-similar-weights_10	4169.799	15075.7	0.0	0.0	9.4	0.0	0.0	0.0	0.0

$$Gap = \frac{OPT - Obj}{OPT}\%$$

References:

[1] F. Neumann, S. Polyakovskiy, M. Skutella, L. Stougie, and J. Wu. A Fully Polynomial Time Approximation Scheme for Packing While Traveling. ArXiv e-prints, 2017.
[2] P. Benchimol, W.-J. v. Hoeve, J.-C. Regin, L.-M. Rousseau, and M. Rueher. Improved filtering for weighted circuit constraints. Constraints, 17(3):205– 233, Jul 2012.

THE UNIVERSITY of ADELAIDE

Optimisation and Logistics
School of Computer Science
The University of Adelaide, Australia

Exact Approaches for the Travelling Thief Problem

Junhua Wu, Markus Wagner, Sergey Polyakovskiy, and Frank Neumann

Motivation

Many evolutionary and constructive heuristic approaches have been introduced in order to solve the Traveling Thief Problem (TTP). However, the accuracy of such approaches is unknown due to their inability to find global optima. We propose three exact algorithms to the TTP. We compare these with the state-of-the-art heuristic approaches to gather a comprehensive overview on the accuracy of heuristic methods for solving small TTP instances.

Travelling Thief Problem

The TTP is a combination of travelling salesman problem (TSP) and 0-1 knapsack problem (KP).

$$Z([\Pi, P]) = \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{m_i} p_{ik} y_{ik}}_{\text{Profits}} - \underbrace{R\left(\frac{d_{x_n x_1}}{\upsilon_{max} - \nu W_{x_n}} + \sum_{i=1}^{n-1} \frac{d_{x_i x_{i+1}}}{\upsilon_{max} - \nu W_{x_i}}\right)}_{\text{Travelling Cost}}$$

Π is a tour for the TSP.

P is a packing plan for the KP.

Dynamic Programming

The DP to the TTP is a combination of Held-Karp algorithm for the TSP and the dynamic programming to the PWT problem[1].

Algorithm 1 Dynamic programming to the TTP

set $A(\{1\}, 1, 0) = 0$
for $w = 1$ to C **do**
  set $A(\{1\}, 1, w) = -\infty$
for $s = 2$ to n **do**
  for any $S \subseteq N : |S| = s,\ 1 \in S$ **do**
    for $w = 0$ to C **do**
      set $A(S, 1, w) = -\infty$
      for any $j \in S,\ j \neq 1$ **do**
        compute $A(S, j, w) =$
        $$\max_{i \in S: i \neq j} \left\{ A(S \setminus \{j\}, i, w - \overline{W}_j(S \setminus \{j\}, i)) + \overline{P}_j(S \setminus \{j\}, i) - \frac{d_{ij}}{\upsilon_{max} - \nu w} \right\}$$
return $\max_{i \in S: i \neq 1} \left\{ A(N, i, w) - \frac{d_{i1}}{\upsilon_{max} - \nu w} \right\}$

Branch and Bound Search

We propose the upper bound that calculates the maximal possible profit that the thief may obtain by passing the remaining part of the tour with the minimal possible cost.

$$E_U(A(S, j, \cdot)) = \max_{0 \le w \le W} A(S, j, w) + \sum_{k \in N \setminus S} \sum_{l=1}^{m_k} p_{kl} - R\frac{d_{j1}}{\upsilon_{max}}$$

Constraint Programming

Our constraint programming model employs a simple permutation based representation of the tour that allows the use of the AllDifferent[2] filtering algorithm.

$$max \sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{ij} y_{ij}$$

$$- R\left(\sum_{i=1}^{n-1} \frac{\texttt{Element}(d, n(x_i - 1) + x_{i+1})}{\upsilon_{max} - \nu \texttt{Element}(W, x_i)} + \frac{\texttt{Element}(d, n(x_n - 1) + 1)}{\upsilon_{max} - \nu \texttt{Element}(W, x_n)}\right)$$

$$\texttt{AllDifferent}[x_1, \ldots, x_n]$$

$$W_i = W_{i-1} + \sum_{j \in M_i} w_{ij} y_{ij},\ i \in \{2, \ldots, n\}$$

$$W_n \le C$$

Experiments

Comparison of the exact approaches.

Instance	n	m	Running time (in seconds) DP	BnB	CP
eil51_n05_m4_uncorr_01	5	4	0.018	0.023	0.222
eil51_n06_m5_uncorr_01	6	5	0.07	0.079	0.24
eil51_n07_m6_uncorr_01	7	6	0.143	0.195	0.497
eil51_n08_m7_uncorr_01	8	7	0.343	0.505	4.594
eil51_n09_m8_uncorr_01	9	8	0.633	1.492	63.838
eil51_n10_m9_uncorr_01	10	9	0.933	5.188	776.55
eil51_n11_m10_uncorr_01	11	10	2.414	23.106	12861.181
eil51_n12_m11_uncorr_01	12	11	3.938	204.786	-
eil51_n13_m12_uncorr_01	13	12	14.217	2007.074	-
eil51_n14_m13_uncorr_01	14	13	13.408	36944.146	-
eil51_n15_m14_uncorr_01	15	14	89.461	-	-
eil51_n16_m15_uncorr_01	16	15	59.526	-	-
eil51_n17_m16_uncorr_01	17	16	134.905	-	-
eil51_n18_m17_uncorr_01	18	17	366.082	-	-
eil51_n19_m18_uncorr_01	19	18	830.18	-	-
eil51_n20_m19_uncorr_01	20	19	2456.873	-	-

Comparison between DP and the heuristics.

gap	MA2B	CS2B	CS2SA	S1	S5	C5	DP-S1	DP-S5
avg	0.3%	15.3%	11.5%	38.9%	15.7%	09.9%	30.1%	3.3%
stdev	2.2%	17.8%	16.7%	29.4%	24.6%	18.8%	20.1%	8.5%

Instance	TTP-DP OPT	RT	MA2B Gap	Std	RT	C5 Gap	Std	DP-S5 Gap	Std
eil51_n05_m4_multiple-strongly-corr_01	619.227	0.02	29.1	12.1	2.71	35.5	1.20e-6	41.3	0.0
eil51_n05_m4_uncorr_01	466.929	0.02	0.0	0.0	3.22	0.0	2.20e-6	0.0	2.20e-6
eil51_n05_m4_uncorr-similar-weights_01	299.281	0.02	0.0	0.0	3.21	7.8	2.40e-6	7.8	1.20e-6
eil51_n05_m20_multiple-strongly-corr_01	773.573	0.08	13.4	0.0	1.44	14.3	0.0	12.8	0.0
eil51_n05_m20_uncorr_01	2144.796	0.07	0.0	0.0	3.35	7.4	0.0	6.6	2.30e-6
eil51_n05_m20_uncorr-similar-weights_01	269.015	0.04	0.0	0.0	3.51	0.0	2.30e-6	0.0	0.0
eil51_n10_m9_multiple-strongly-corr_01	573.897	1.21	0.0	0.0	6.07	0.0	0.0	0.0	0.0
eil51_n10_m9_uncorr_01	1125.715	0.93	0.0	0.0	6.06	0.0	1.30e-6	0.0	1.30e-6
eil51_n10_m9_uncorr-similar-weights_01	753.230	0.86	0.0	0.0	5.87	0.0	0.0	0.0	0.0
eil51_n10_m45_multiple-strongly-corr_01	1091.127	14.89	0.0	0.0	7.99	0.0	0.0	0.0	0.0
eil51_n10_m45_uncorr_01	6009.431	6.39	0.0	0.0	8.6	6.6	2.30e-6	0.0	0.0
eil51_n10_m45_uncorr-similar-weights_01	3009.553	8.87	0.0	0.0	6.78	0.0	2.30e-6	0.0	2.30e-6
eil51_n12_m11_multiple-strongly-corr_01	648.546	4.58	0.0	0.0	6.08	4.6	2.20e-6	4.6	2.20e-6
eil51_n12_m11_uncorr_01	1717.699	3.94	0.0	0.0	7.21	0.0	1.20e-6	0.0	1.20e-6
eil51_n12_m11_uncorr-similar-weights_01	774.107	3.36	0.0	0.0	7.03	0.0	2.30e-6	0.0	2.30e-6
eil51_n12_m55_multiple-strongly-corr_01	1251.780	117.99	0.0	0.0	9.19	0.0	0.0	0.0	0.0
eil51_n12_m55_uncorr_01	8838.012	35.79	0.0	0.0	9.76	0.0	0.0	0.0	0.0
eil51_n12_m55_uncorr-similar-weights_01	3734.895	38.36	12.3	0.0	8.34	12.3	0.0	0.2	0.0
eil51_n15_m14_multiple-strongly-corr_01	547.419	39.82	0.0	0.0	7.87	14.1	1.30e-6	13.3	1.30e-6
eil51_n15_m14_uncorr_01	2392.996	89.46	0.0	0.0	7.28	3.8	0.0	3.8	0.0
eil51_n15_m14_uncorr-similar-weights_01	637.419	16.35	0.0	0.0	6.86	0.0	1.60e-6	0.0	1.60e-6
eil51_n15_m70_multiple-strongly-corr_01	920.372	3984.29	2.1	1.1	12.11	0.0	2.70e-6	0.0	2.70e-6
eil51_n15_m70_uncorr_01	9922.137	740.22	0.0	0.0	9.67	7	1.20e-6	1.9	0.0
eil51_n15_m70_uncorr-similar-weights_01	4659.623	867.78	0.0	0.0	7.98	0.0	0.0	0.0	0.0
eil51_n16_m15_multiple-strongly-corr_01	794.745	105.5	0.0	0.0	7.7	18.9	1.6e-6	18.9	1.6e-6
eil51_n16_m15_multiple-strongly-corr_10	4498.848	623.4	0.0	0.0	9.1	12.9	0.0	16.6	1.3e-6
eil51_n16_m15_uncorr_01	2490.889	59.5	1.0	0.7	8.4	1.6	2.3e-6	1.6	2.3e-6
eil51_n16_m15_uncorr_10	3601.077	211.5	0.0	0.0	9.0	7.1	1.6e-6	7.1	1.6e-6
eil51_n16_m15_uncorr-similar-weights_01	540.897	36.4	0.0	0.0	8.5	0.0	3.0e-6	0.0	3.0e-6
eil51_n16_m15_uncorr-similar-weights_10	3948.211	245.4	0.0	0.0	8.7	5.8	1.5e-6	13.6	0.0
eil51_n17_m16_multiple-strongly-corr_01	685.565	248.6	0.0	0.0	8.4	0.2	1.5e-6	0.0	1.5e-6
eil51_n17_m16_multiple-strongly-corr_10	3826.098	2190.4	0.0	0.0	9.8	0.0	1.5e-6	0.0	1.5e-6
eil51_n17_m16_uncorr_01	2342.664	134.9	0.0	0.0	8.3	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr_10	2275.279	554.5	0.0	0.0	9.6	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr-similar-weights_01	556.851	70.8	0.0	0.0	8.1	0.0	0.0	0.0	0.0
eil51_n17_m16_uncorr-similar-weights_10	2935.961	787.7	0.0	0.0	9.7	0.0	0.0	0.0	0.0
eil51_n18_m17_multiple-strongly-corr_01	834.031	715.7	7.9	0.8	10.2	9.2	0.0	12.9	1.7e-6
eil51_n18_m17_multiple-strongly-corr_10	5531.373	6252.4	0.0	0.0	10.5	0.4	1.5e-6	0.4	1.5e-6
eil51_n18_m17_uncorr_01	2644.491	366.1	0.0	0.0	9.7	0.2	0.0	1.8	0.0
eil51_n18_m17_uncorr_10	3222.603	1462.7	0.0	0.0	10.3	0.0	1.3e-6	0.2	0.0
eil51_n18_m17_uncorr-similar-weights_01	532.906	148.3	0.0	0.0	8.5	0.0	1.3e-6	0.0	1.3e-6
eil51_n18_m17_uncorr-similar-weights_10	4420.438	1929.3	0.0	0.0	9.9	0.0	2.9e-6	0.3	1.8e-6
eil51_n19_m18_multiple-strongly-corr_01	910.229	1771.6	0.0	0.0	9.3	20.1	1.6e-6	20.1	1.6e-6
eil51_n19_m18_multiple-strongly-corr_10	-	-	-	-	10.4	-	-	-	-
eil51_n19_m18_uncorr_01	2604.844	830.2	0.0	0.0	9.7	0.0	0.0	0.0	0.0
eil51_n19_m18_uncorr_10	4048.408	3884.3	0.0	0.0	10.9	0.0	1.4e-6	0.0	1.4e-6
eil51_n19_m18_uncorr-similar-weights_01	472.186	412.3	0.0	0.0	9.2	0.0	1.5e-6	0.0	1.5e-6
eil51_n19_m18_uncorr-similar-weights_10	5573.695	5878.8	0.0	0.0	10.5	0.0	0.0	0.0	0.0
eil51_n20_m19_multiple-strongly-corr_01	518.189	4533.7	0.6	0.6	11.1	14.1	1.4e-6	12.3	0.0
eil51_n20_m19_multiple-strongly-corr_10	-	-	-	-	12.1	-	-	-	-
eil51_n20_m19_uncorr_01	2092.673	2456.9	0.0	0.0	8.7	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr_10	3044.391	12776.0	0.0	0.0	9.8	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr-similar-weights_01	451.052	1007.7	0.0	0.0	7.9	0.0	0.0	0.0	0.0
eil51_n20_m19_uncorr-similar-weights_10	4169.799	15075.7	0.0	0.0	9.4	0.0	0.0	0.0	0.0

$$Gap = \frac{OPT - Obj}{OPT}\%$$

References:

[1] F. Neumann, S. Polyakovskiy, M. Skutella, L. Stougie, and J. Wu. A Fully Polynomial Time Approximation Scheme for Packing While Traveling. ArXiv e-prints, 2017.

[2] P. Benchimol, W.-J. v. Hoeve, J.-C. Regin, L.-M. Rousseau, and M. Rueher. Improved filtering for weighted circuit constraints. Constraints, 17(3):205– 233, Jul 2012.