CYTED WORKSHOP

Madrid – 28-29 November 2016

Solving Large-Scale Time Capacitated Arc Routing Problems

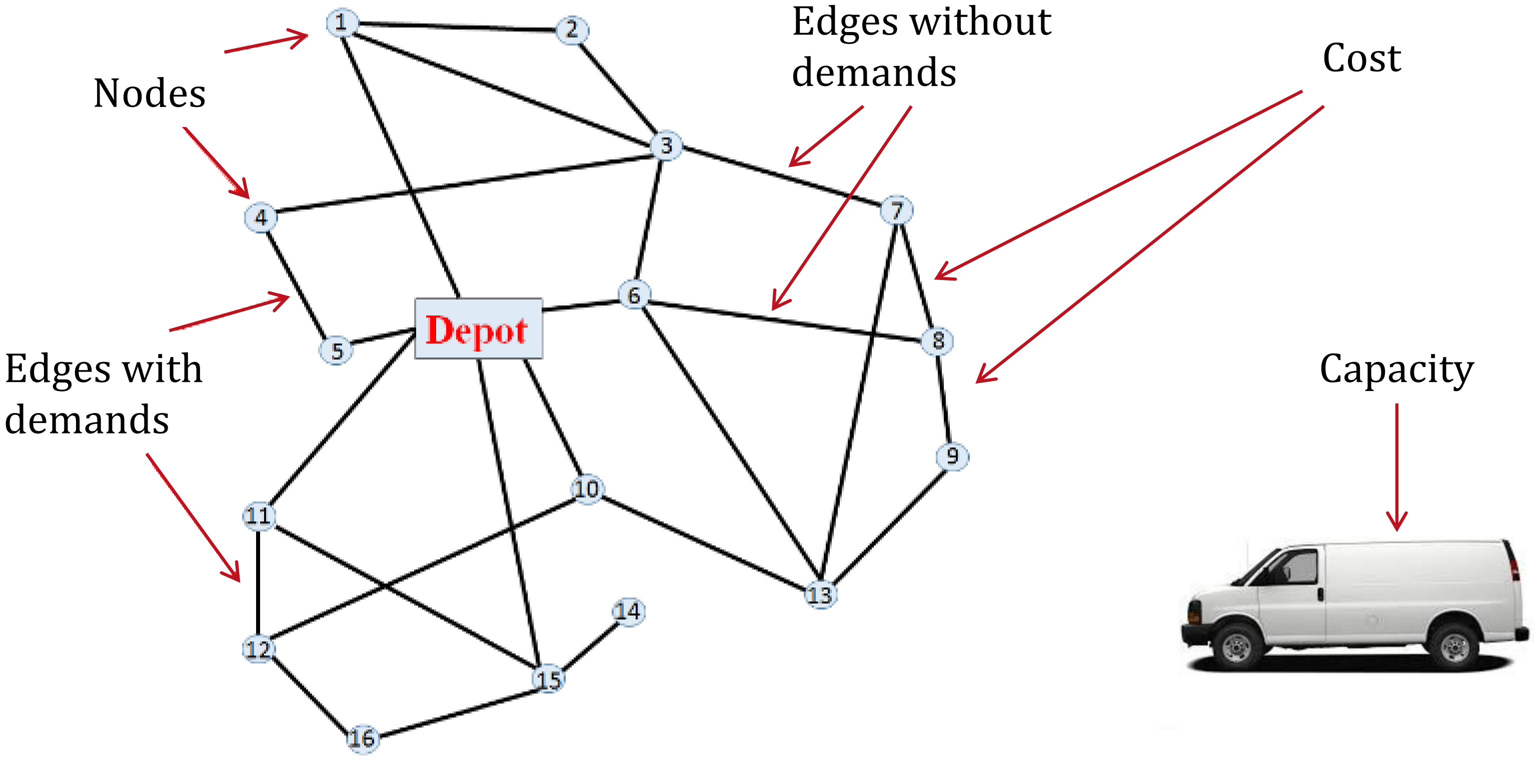
FROM REAL-TIME HEURISTICS TO METAHEURISTICS

Jésica de Armas - jesica.dearmas@upf.edu
Angel A. Juan - ajuanp@uoc.edu
Peter Keenan - peter.keenan@ucd.ie
Sean McGarraghy - sean.mcgarraghy@ucd.ie

Outline

- Time Capacitated Arc Routing Problem
- Our Approaches:
  - TSHARP
  - BR-SA Metaheuristic
- Computational Experiments
- Analysis of Results
- Conclusions
- Future work

Capacitated Arc Routing Problem



Nodes

Edges without demands

Cost

Edges with demands

Capacity

Time Capacitated Arc Routing Problem

- The Time Capacitated Arc Routing Problem (TCARP) represents a problem where a time restriction, rather than a volume limitation, provides a capacity restraint for the problem.

- This typically arises in problems where volume constraints are not relevant, for instance meter reading, rail inspection or road inspection.

Time Capacitated Arc Routing Problem

- In TCARP, either traversing or servicing an arc will exhaust the time capacity of a vehicle. The travel time of an edge is related to its length, and the service demand is equal to the travel time plus the time for visiting all customers on the edge.

- **Both travel and service demands use the vehicle capacity Q**.

- A simple approach to a TCARP problem is to substitute units of time for units of volume and apply techniques devised for volume-constrained problems. However, this approach does not recognize the specic characteristics of TCARP. In a volume-based problem, the capacity of the vehicle is used up by servicing arcs, but the mere transit of an arc does not add any volume to the route.

Previous Works

There has been a very limited set of work directly on the TCARP:

- Some experiments have been performed using heuristics on rural postal delivery problems (Keenan and Naughton, 1996; Keenan, 2001).
- Keenan (2001, 2005) tested graph theory based lower bounds for the TCARP.
- Bartolini et al. (2013) addressed the more general CARPDD, and developed lower bounds and an exact algorithm for this problem based on cut-and-column generation and branch-and-price. Bartolini et al. (2013) tested the datasets used by Keenan (2005) and provided superior lower bounds and introduced a new medium sized TCARP dataset. These authors provided lower bounds and some optimal solutions for this new dataset, but excessive execution time prevented other solutions being found.

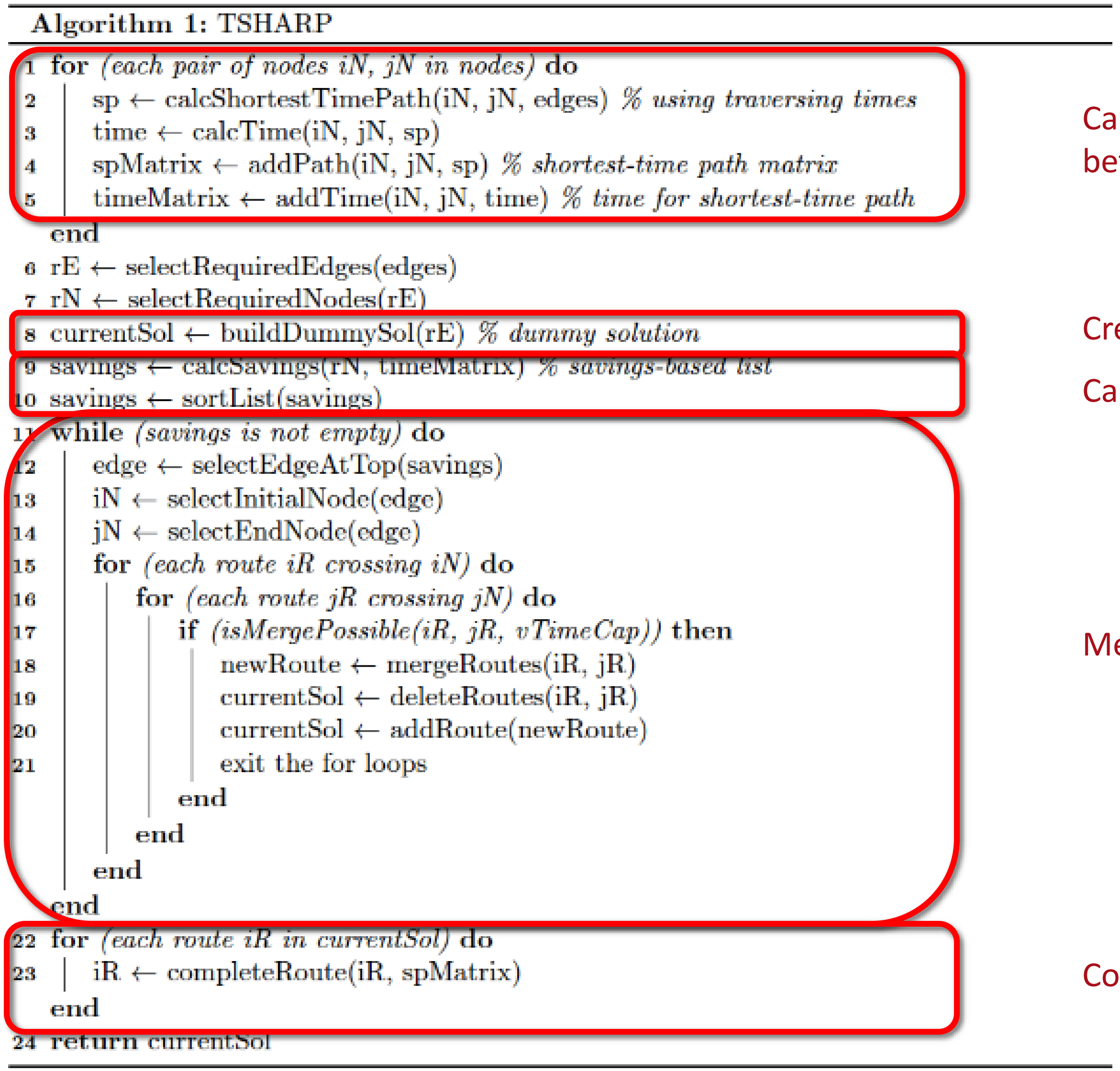
Contributions

Against this background where computation time remains an obstacle, the aim of this paper is to tackle the TCARP using heuristic and metaheuristic algorithms in order to obtain solutions that improve previous results in the literature in short computational times.

Therefore, the **main contributions** of this work are:

1. A savings-based heuristic, TSHARP, which is able to generate reasonably good solutions for the TCARP in milliseconds even for the largest instances considered
2. A Biased-Randomized Simulated Annealing algorithm capable of improving previously published results on the TCARP both in quality as well as in computing times
3. A new set of large-scale instances based on real-life cases.

TSHARP



Algorithm 1: TSHARP

```
for (each pair of nodes iN, jN in nodes) do
    sp ← calcShortestTimePath(iN, jN, edges) % using traversing times
    time ← calcTime(iN, jN, sp)
    spMatrix ← addPath(iN, jN, sp) % shortest-time path matrix
    timeMatrix ← addTime(iN, jN, time) % time for shortest-time path
end
rE ← selectRequiredEdges(edges)
rN ← selectRequiredNodes(rE)
currentSol ← buildDummySol(rE) % dummy solution
savings ← calcSavings(rN, timeMatrix) % savings-based list
savings ← sortList(savings)
while (savings is not empty) do
    edge ← selectEdgeAtTop(savings)
    iN ← selectInitialNode(edge)
    jN ← selectEndNode(edge)
    for (each route iR crossing iN) do
        for (each route jR crossing jN) do
            if (isMergePossible(iR, jR, vTimeCap)) then
                newRoute ← mergeRoutes(iR, jR)
                currentSol ← deleteRoutes(iR, jR)
                currentSol ← addRoute(newRoute)
                exit the for loops
            end
        end
    end
end
for (each route iR in currentSol) do
    iR ← completeRoute(iR, spMatrix)
end
return currentSol
```

Calculate shortest path between any two nodes

Create dummy solutions

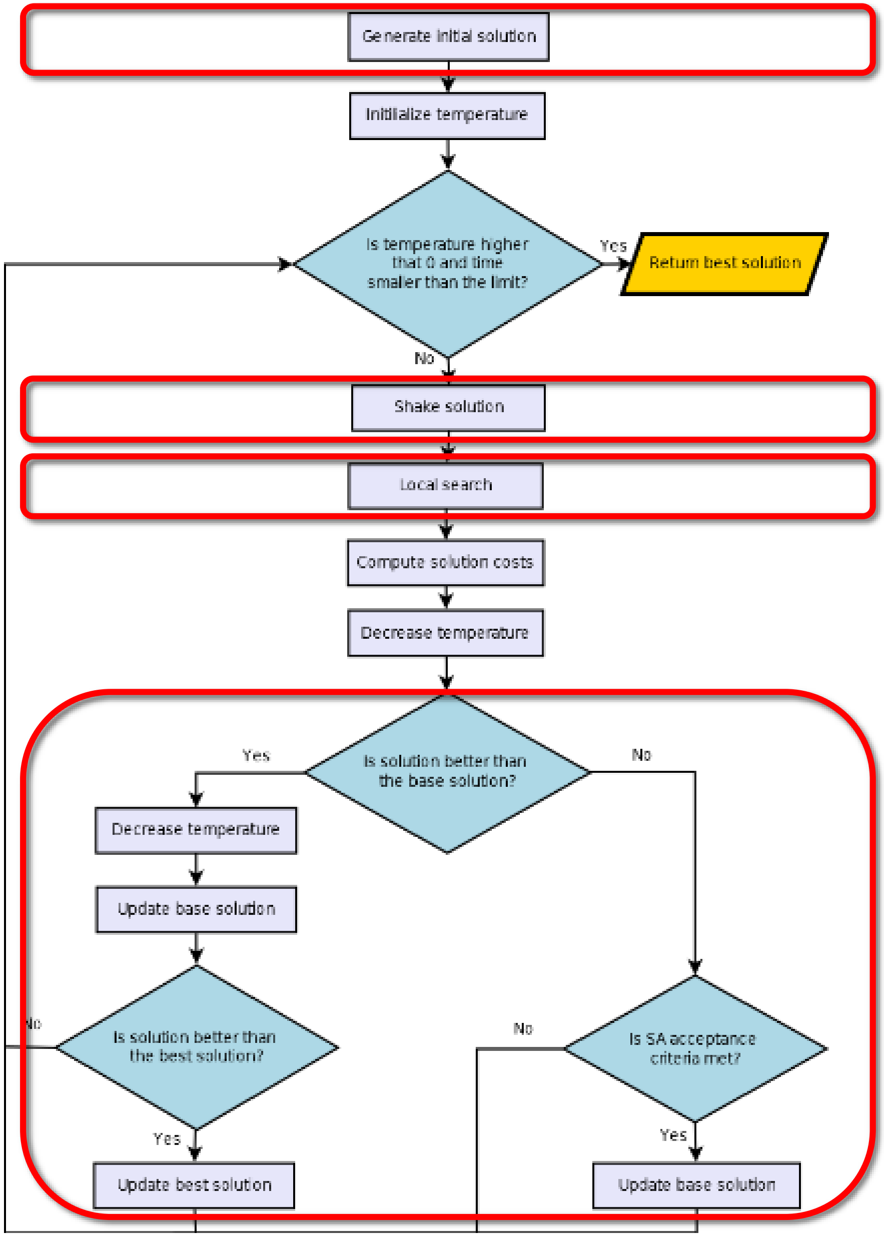
Calculate savings

Merge routes

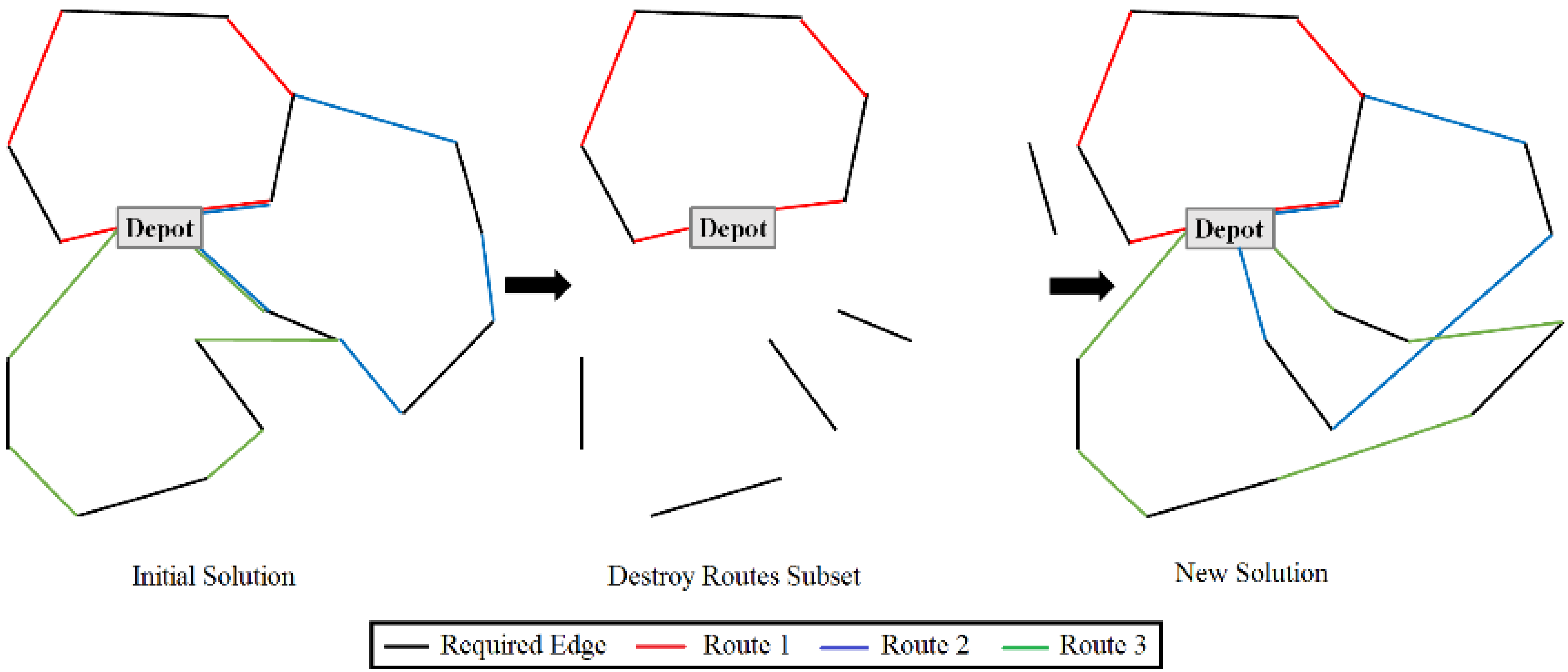
Complete routes

BR-SA

- The biased-randomization of the heuristic is obtained in the following way.
- Firstly, the savings list is constructed and sorted as in the TSHARP procedure. Then, each edge in this list receives some probability of being selected according to a skewed probability distribution (in our case, a geometric distribution is used).
- Thus, at each step, instead of making a greedy choose of the next merging edge, the assigned probabilities are used to select the next edge during the solution-construction process.

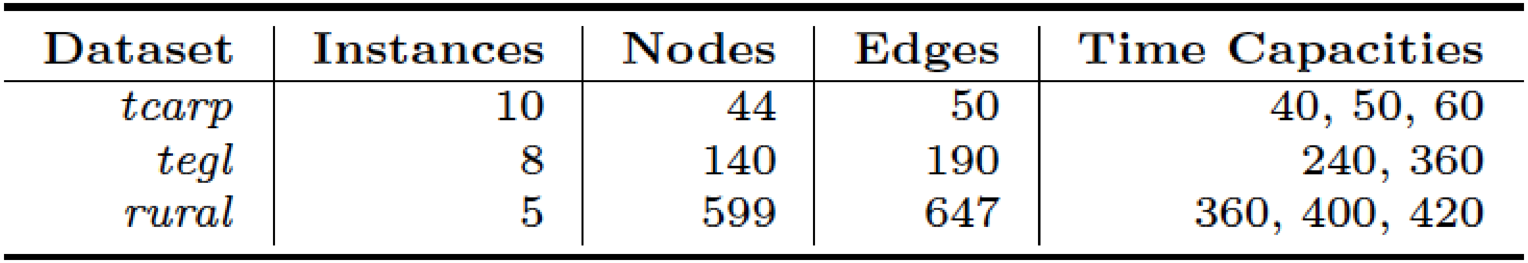

Generate initial solution
Initialize temperature
Is temperature higher that 0 and time smaller than the limit?
Yes
Return best solution
No
Shake solution
Local search
Compute solution costs
Decrease temperature
Is solution better than the base solution?
Yes
No
Decrease temperature
Update base solution
Is solution better than the best solution?
No
Yes
Update best solution
Is SA acceptance criteria met?
No
Yes
Update base solution

Shaking


Depot
Depot
Depot
Initial Solution
Destroy Routes Subset
New Solution
Required Edge
Route 1
Route 2
Route 3

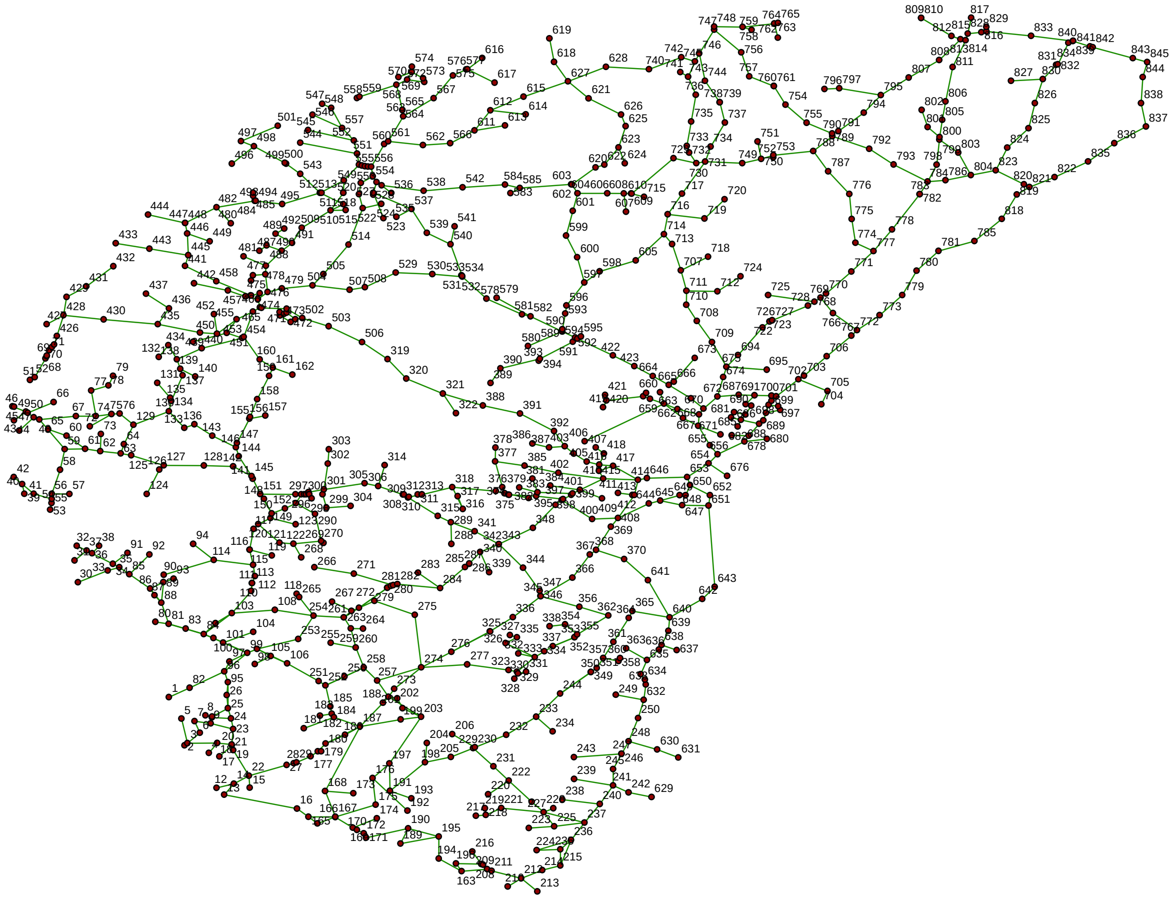
Computational Experiments

Two set of classical TCARP benchmarks have been employed. Additionally, new benchmarks regarding large-scale instances (which are based on real-live cases) have been proposed and solved with our algorithms.

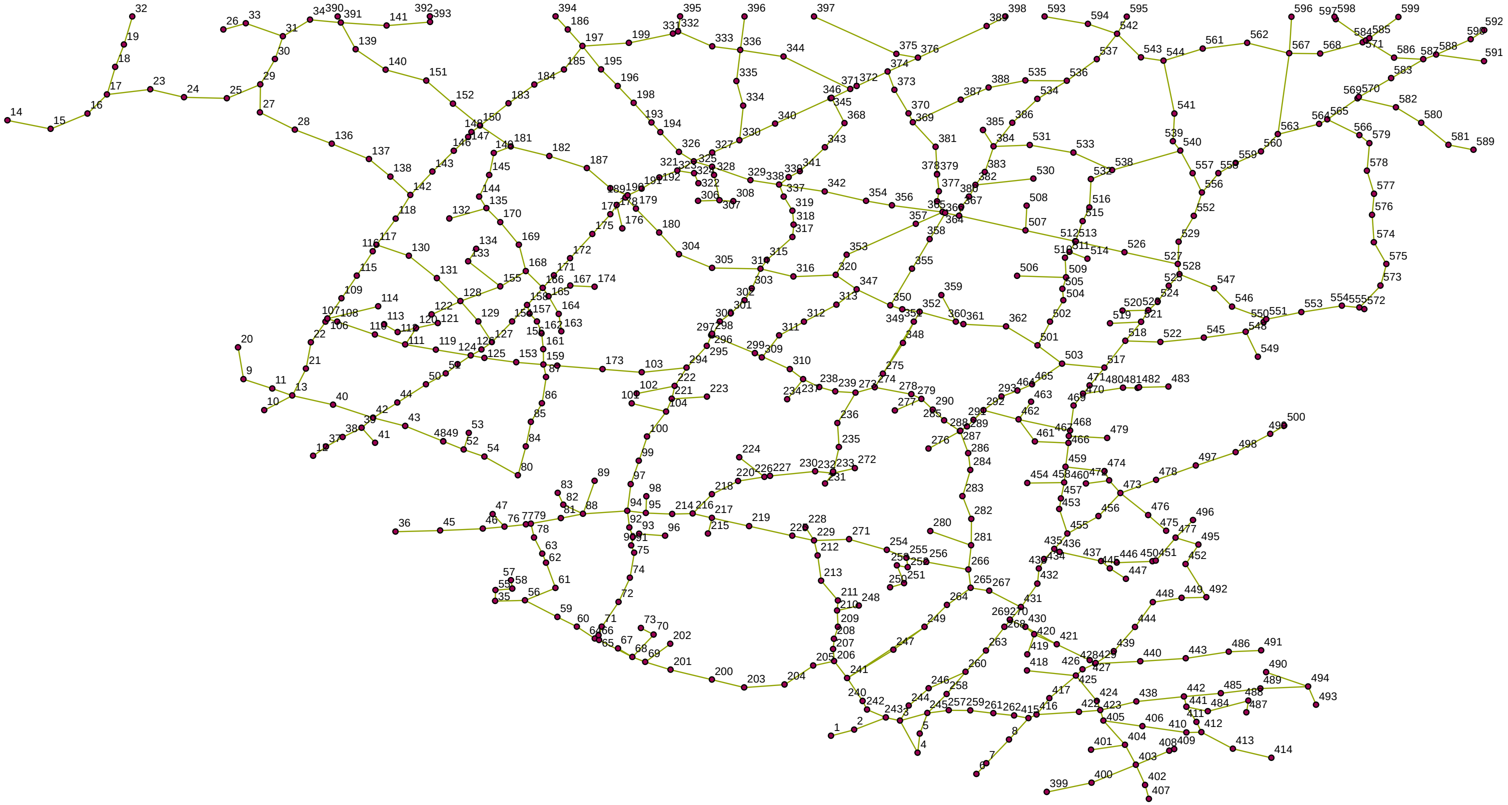


Dataset	Instances	Nodes	Edges	Time Capacities
tcarp	10	44	50	40, 50, 60
tegl	8	140	190	240, 360
rural	5	599	647	360, 400, 420

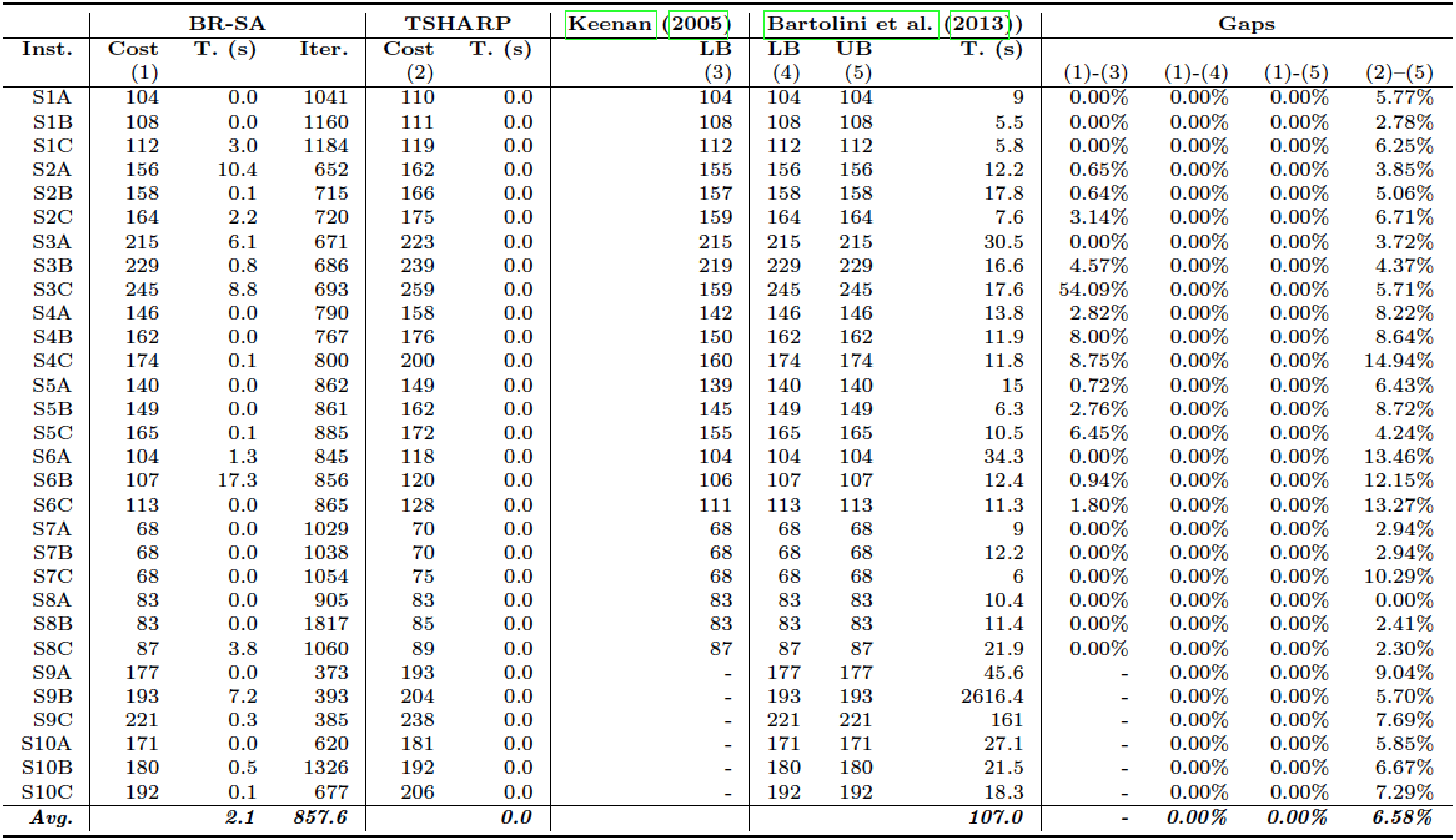
Computational Experiments



Computational Experiments

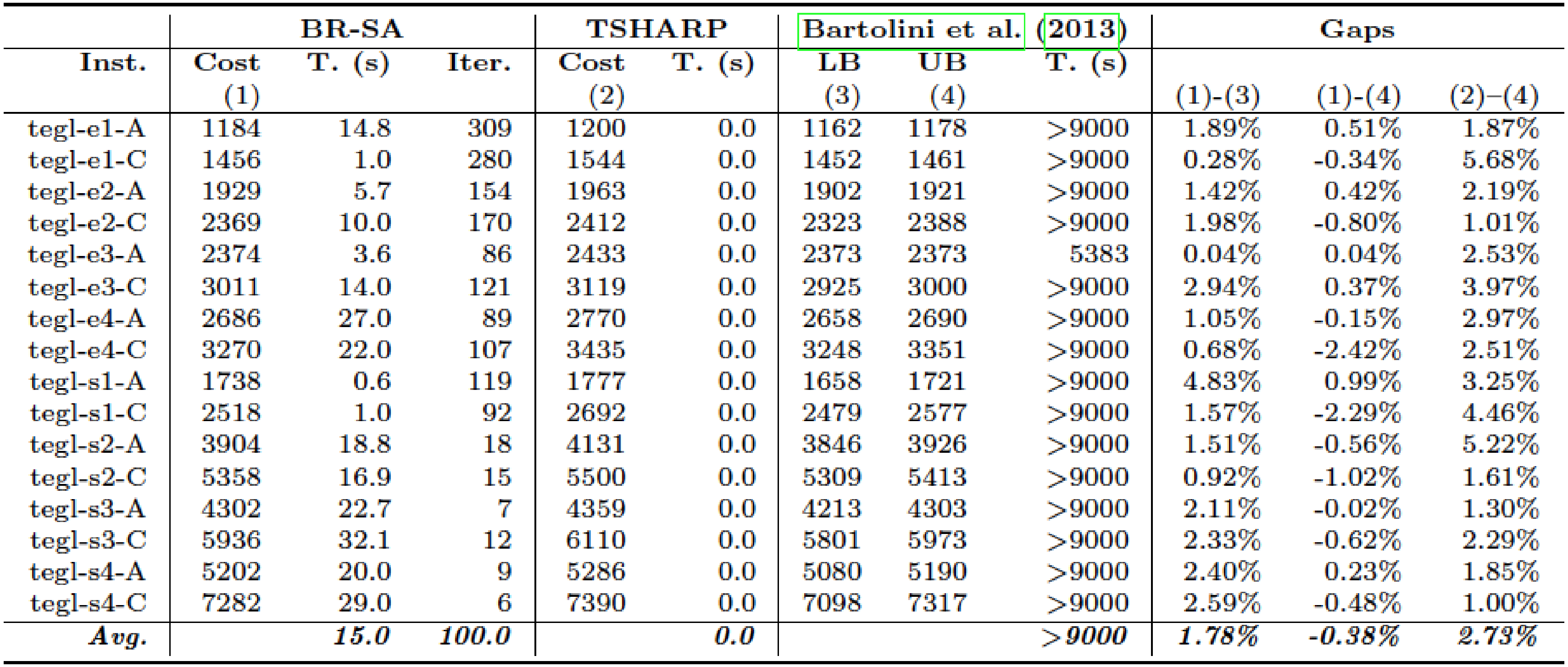


Computational Experiments



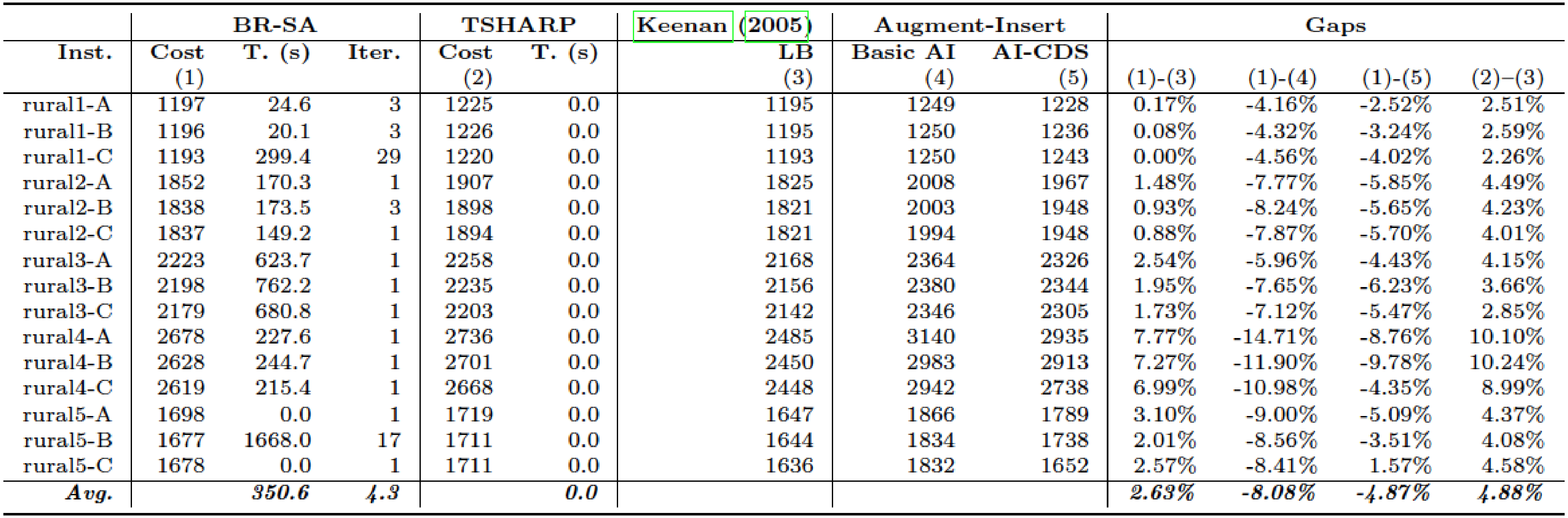
	BR-SA			TSHARP		Keenan (2005)	Bartolini et al. (2013))			Gaps			
Inst.	Cost	T. (s)	Iter.	Cost	T. (s)	LB	LB	UB	T. (s)				
	(1)			(2)		(3)	(4)	(5)		(1)-(3)	(1)-(4)	(1)-(5)	(2)–(5)
S1A	104	0.0	1041	110	0.0	104	104	104	9	0.00%	0.00%	0.00%	5.77%
S1B	108	0.0	1160	111	0.0	108	108	108	5.5	0.00%	0.00%	0.00%	2.78%
S1C	112	3.0	1184	119	0.0	112	112	112	5.8	0.00%	0.00%	0.00%	6.25%
S2A	156	10.4	652	162	0.0	155	156	156	12.2	0.65%	0.00%	0.00%	3.85%
S2B	158	0.1	715	166	0.0	157	158	158	17.8	0.64%	0.00%	0.00%	5.06%
S2C	164	2.2	720	175	0.0	159	164	164	7.6	3.14%	0.00%	0.00%	6.71%
S3A	215	6.1	671	223	0.0	215	215	215	30.5	0.00%	0.00%	0.00%	3.72%
S3B	229	0.8	686	239	0.0	219	229	229	16.6	4.57%	0.00%	0.00%	4.37%
S3C	245	8.8	693	259	0.0	159	245	245	17.6	54.09%	0.00%	0.00%	5.71%
S4A	146	0.0	790	158	0.0	142	146	146	13.8	2.82%	0.00%	0.00%	8.22%
S4B	162	0.0	767	176	0.0	150	162	162	11.9	8.00%	0.00%	0.00%	8.64%
S4C	174	0.1	800	200	0.0	160	174	174	11.8	8.75%	0.00%	0.00%	14.94%
S5A	140	0.0	862	149	0.0	139	140	140	15	0.72%	0.00%	0.00%	6.43%
S5B	149	0.0	861	162	0.0	145	149	149	6.3	2.76%	0.00%	0.00%	8.72%
S5C	165	0.1	885	172	0.0	155	165	165	10.5	6.45%	0.00%	0.00%	4.24%
S6A	104	1.3	845	118	0.0	104	104	104	34.3	0.00%	0.00%	0.00%	13.46%
S6B	107	17.3	856	120	0.0	106	107	107	12.4	0.94%	0.00%	0.00%	12.15%
S6C	113	0.0	865	128	0.0	111	113	113	11.3	1.80%	0.00%	0.00%	13.27%
S7A	68	0.0	1029	70	0.0	68	68	68	9	0.00%	0.00%	0.00%	2.94%
S7B	68	0.0	1038	70	0.0	68	68	68	12.2	0.00%	0.00%	0.00%	2.94%
S7C	68	0.0	1054	75	0.0	68	68	68	6	0.00%	0.00%	0.00%	10.29%
S8A	83	0.0	905	83	0.0	83	83	83	10.4	0.00%	0.00%	0.00%	0.00%
S8B	83	0.0	1817	85	0.0	83	83	83	11.4	0.00%	0.00%	0.00%	2.41%
S8C	87	3.8	1060	89	0.0	87	87	87	21.9	0.00%	0.00%	0.00%	2.30%
S9A	177	0.0	373	193	0.0	-	177	177	45.6	-	0.00%	0.00%	9.04%
S9B	193	7.2	393	204	0.0	-	193	193	2616.4	-	0.00%	0.00%	5.70%
S9C	221	0.3	385	238	0.0	-	221	221	161	-	0.00%	0.00%	7.69%
S10A	171	0.0	620	181	0.0	-	171	171	27.1	-	0.00%	0.00%	5.85%
S10B	180	0.5	1326	192	0.0	-	180	180	21.5	-	0.00%	0.00%	6.67%
S10C	192	0.1	677	206	0.0	-	192	192	18.3	-	0.00%	0.00%	7.29%
Avg.		***2.1***	***857.6***		***0.0***				***107.0***	***-***	***0.00%***	***0.00%***	***6.58%***

Computational Experiments



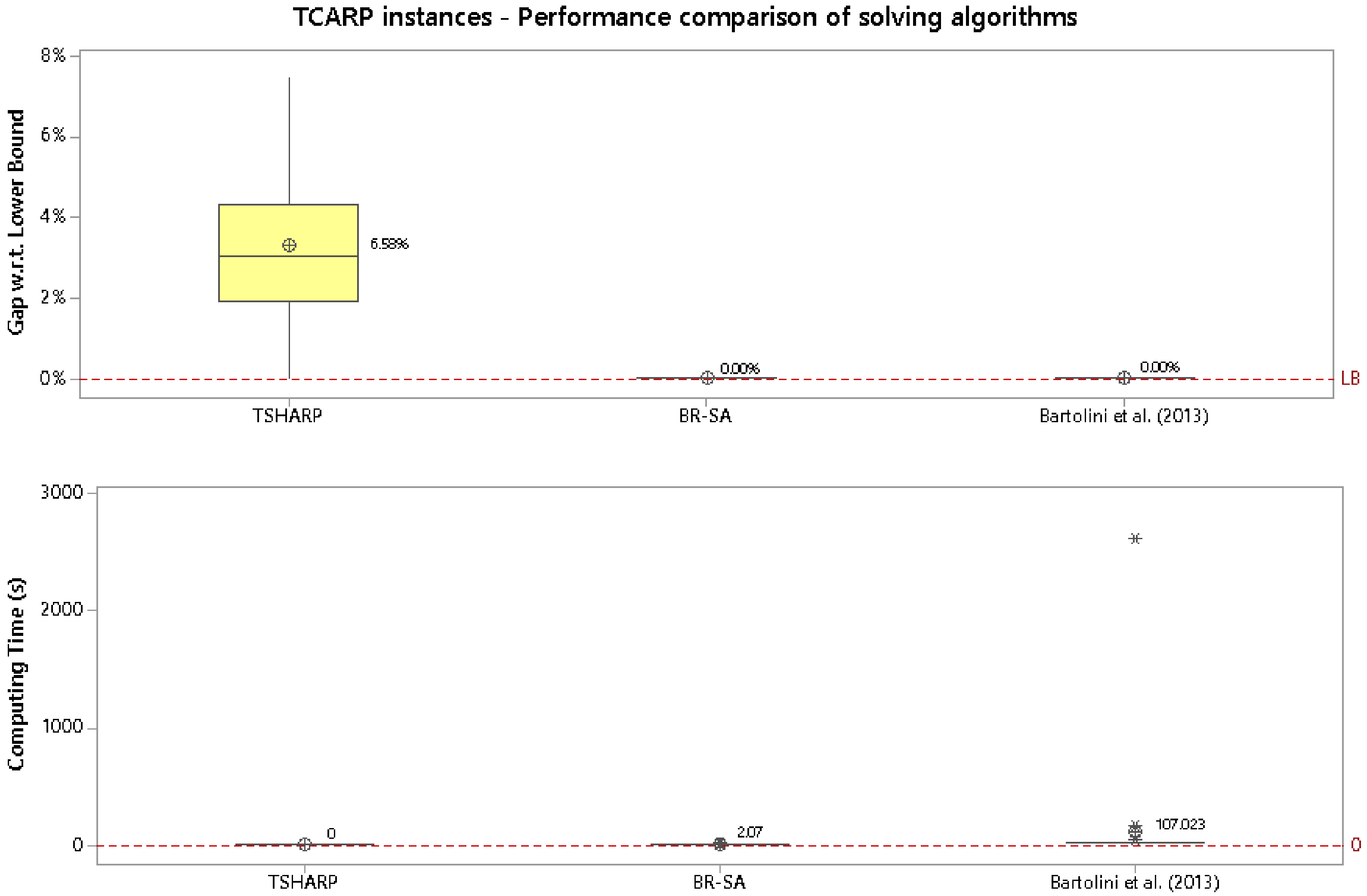
	BR-SA			TSHARP		Bartolini et al. (2013)			Gaps		
Inst.	Cost	T. (s)	Iter.	Cost	T. (s)	LB	UB	T. (s)			
	(1)			(2)		(3)	(4)		(1)-(3)	(1)-(4)	(2)–(4)
tegl-e1-A	1184	14.8	309	1200	0.0	1162	1178	>9000	1.89%	0.51%	1.87%
tegl-e1-C	1456	1.0	280	1544	0.0	1452	1461	>9000	0.28%	-0.34%	5.68%
tegl-e2-A	1929	5.7	154	1963	0.0	1902	1921	>9000	1.42%	0.42%	2.19%
tegl-e2-C	2369	10.0	170	2412	0.0	2323	2388	>9000	1.98%	-0.80%	1.01%
tegl-e3-A	2374	3.6	86	2433	0.0	2373	2373	5383	0.04%	0.04%	2.53%
tegl-e3-C	3011	14.0	121	3119	0.0	2925	3000	>9000	2.94%	0.37%	3.97%
tegl-e4-A	2686	27.0	89	2770	0.0	2658	2690	>9000	1.05%	-0.15%	2.97%
tegl-e4-C	3270	22.0	107	3435	0.0	3248	3351	>9000	0.68%	-2.42%	2.51%
tegl-s1-A	1738	0.6	119	1777	0.0	1658	1721	>9000	4.83%	0.99%	3.25%
tegl-s1-C	2518	1.0	92	2692	0.0	2479	2577	>9000	1.57%	-2.29%	4.46%
tegl-s2-A	3904	18.8	18	4131	0.0	3846	3926	>9000	1.51%	-0.56%	5.22%
tegl-s2-C	5358	16.9	15	5500	0.0	5309	5413	>9000	0.92%	-1.02%	1.61%
tegl-s3-A	4302	22.7	7	4359	0.0	4213	4303	>9000	2.11%	-0.02%	1.30%
tegl-s3-C	5936	32.1	12	6110	0.0	5801	5973	>9000	2.33%	-0.62%	2.29%
tegl-s4-A	5202	20.0	9	5286	0.0	5080	5190	>9000	2.40%	0.23%	1.85%
tegl-s4-C	7282	29.0	6	7390	0.0	7098	7317	>9000	2.59%	-0.48%	1.00%
Avg.		***15.0***	***100.0***		***0.0***			***>9000***	***1.78%***	***-0.38%***	***2.73%***

Computational Experiments



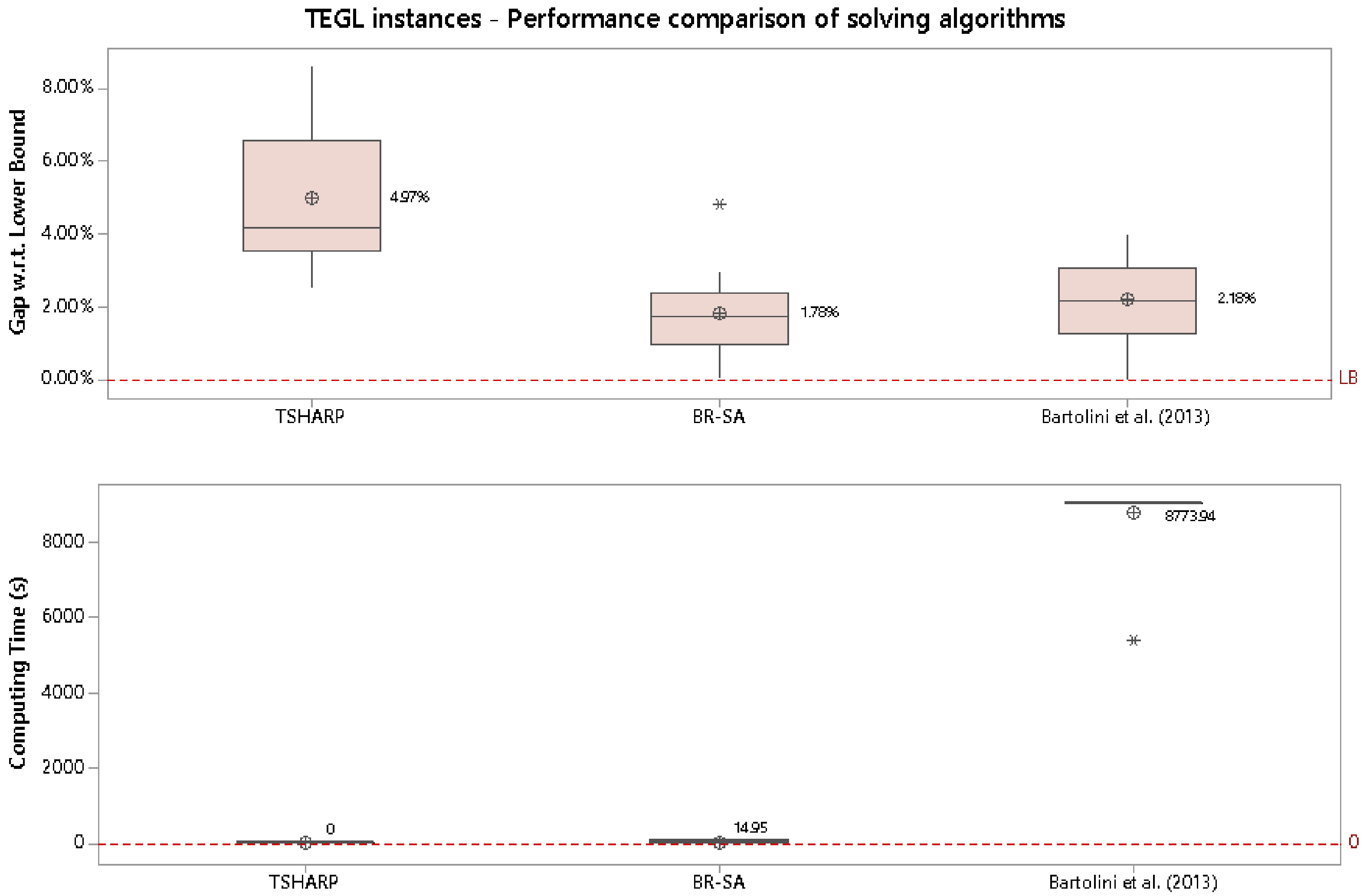
	BR-SA			TSHARP		Keenan (2005)	Augment-Insert		Gaps			
Inst.	Cost	T. (s)	Iter.	Cost	T. (s)	LB	Basic AI	AI-CDS				
	(1)			(2)		(3)	(4)	(5)	(1)-(3)	(1)-(4)	(1)-(5)	(2)–(3)
rural1-A	1197	24.6	3	1225	0.0	1195	1249	1228	0.17%	-4.16%	-2.52%	2.51%
rural1-B	1196	20.1	3	1226	0.0	1195	1250	1236	0.08%	-4.32%	-3.24%	2.59%
rural1-C	1193	299.4	29	1220	0.0	1193	1250	1243	0.00%	-4.56%	-4.02%	2.26%
rural2-A	1852	170.3	1	1907	0.0	1825	2008	1967	1.48%	-7.77%	-5.85%	4.49%
rural2-B	1838	173.5	3	1898	0.0	1821	2003	1948	0.93%	-8.24%	-5.65%	4.23%
rural2-C	1837	149.2	1	1894	0.0	1821	1994	1948	0.88%	-7.87%	-5.70%	4.01%
rural3-A	2223	623.7	1	2258	0.0	2168	2364	2326	2.54%	-5.96%	-4.43%	4.15%
rural3-B	2198	762.2	1	2235	0.0	2156	2380	2344	1.95%	-7.65%	-6.23%	3.66%
rural3-C	2179	680.8	1	2203	0.0	2142	2346	2305	1.73%	-7.12%	-5.47%	2.85%
rural4-A	2678	227.6	1	2736	0.0	2485	3140	2935	7.77%	-14.71%	-8.76%	10.10%
rural4-B	2628	244.7	1	2701	0.0	2450	2983	2913	7.27%	-11.90%	-9.78%	10.24%
rural4-C	2619	215.4	1	2668	0.0	2448	2942	2738	6.99%	-10.98%	-4.35%	8.99%
rural5-A	1698	0.0	1	1719	0.0	1647	1866	1789	3.10%	-9.00%	-5.09%	4.37%
rural5-B	1677	1668.0	17	1711	0.0	1644	1834	1738	2.01%	-8.56%	-3.51%	4.08%
rural5-C	1678	0.0	1	1711	0.0	1636	1832	1652	2.57%	-8.41%	1.57%	4.58%
Avg.		*350.6*	*4.3*		*0.0*				*2.63%*	*-8.08%*	*-4.87%*	*4.88%*

Analysis of Results

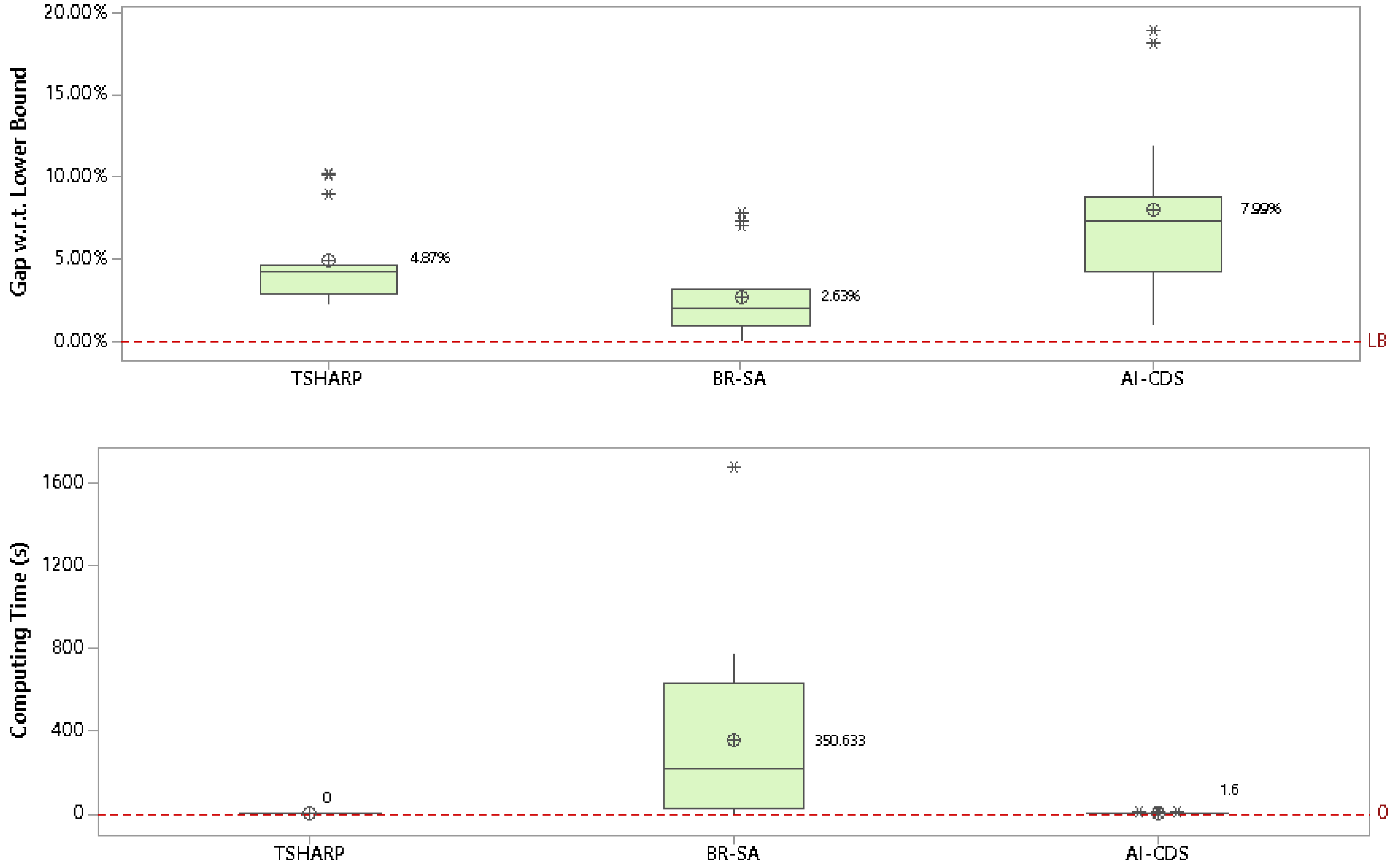


TCARP instances - Performance comparison of solving algorithms

Analysis of Results


TEGL instances - Performance comparison of solving algorithms

Analysis of Results



RURAL instances - Performance comparison of solving algorithms

Conclusions

- This paper discusses realistic applications of the Time Capacitated Arc Routing Problem (TCARP), in which large-scale instances have been solved for the first time in the related literature.

- On the one hand, the TSHARP heuristic is capable of providing reasonably good solutions in just a few milliseconds.

- On the other hand, the BR-SA metaheuristic (which combines a biased-randomized version of the aforementioned TSHARP heuristic with a Simulated Annealing framework) shows an excellent trade-off performance of quality solution versus computing time. Thus, the BR-SA algorithm outperforms previously published approaches in the literature while, at the same time, is relatively easy to implement in practice.

- A new set of well-deffined large scale instances that can be used by other researchers to test their solving approaches.

Future Work

- Additional capacity constraints could be considered, e.g., a time-based as well as a volume based constraint for each vehicle

- Additional realistic goals could be included in the objective function, e.g., route balancing goals

- A multi-depot version of the TCARP, considering a limited capacity per depot, could be explored

- It could be worth considering the effects of parallel and distributed computing techniques on the algorithms' performance

- A stochastic version of the problem, in which either customers' demands or traveling/servicing times were modeled as random variables, would make the problem even more realistic

CYTED WORKSHOP

Madrid – 28-29 November 2016

CYTED

Universitat Pompeu Fabra Barcelona

THANK YOU FOR YOUR ATTENTION

Solving Large-Scale Time Capacitated Arc Routing Problems

FROM REAL-TIME HEURISTICS TO METAHEURISTICS

Jésica de Armas - jesica.dearmas@upf.edu
Angel A. Juan - ajuanp@uoc.edu
Peter Keenan - peter.keenan@ucd.ie
Sean McGarraghy - sean.mcgarraghy@ucd.ie