Operations Research Forum (2021) 2: 48
https://doi.org/10.1007/s43069-021-00084-x

SHORT COMMUNICATION

New Instances for Maximum Weight Independent Set From a Vehicle Routing Application

Yuanyuan Dong[1] · **Andrew V. Goldberg**[2] · **Alexander Noe**[3] · **Nikos Parotsidis**[4] · **Mauricio G. C. Resende**[5] · **Quico Spaen**[6]

Received: 10 June 2021 / Accepted: 6 July 2021 / Published online: 25 September 2021

Abstract
We present a set of new instances of the maximum weight independent set problem. These instances are derived from a real-world vehicle routing problem and are challenging to solve in part because of their large size. We present instances with up to 881 thousand nodes and 383 million edges.

Keywords Maximum weight independent set · Problem instances · Experimental algorithms

Parts of this paper were written, while the first, third, and fourth authors were employed at Amazon.com.

✉ Mauricio G. C. Resende
mgcr@berkeley.edu; mgcr@uw.edu

Yuanyuan Dong
njdyy03@gmail.com

Andrew V. Goldberg
avg@alum.mit.edu

Alexander Noe
alexander.noe@univie.ac.at

Nikos Parotsidis
nickparo1@gmail.com

Quico Spaen
qspaen@berkeley.edu

1 Dallas, TX, USA

2 Amazon.com, East Palo Alto, CA, USA

3 University of Vienna, Vienna, Austria

4 Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

5 Amazon.com and Industrial & Systems Engineering, University of Washington, Seattle, WA, USA

6 Amazon.com, East Palo Alto, CA, USA

1 Vehicle Routing Application of MWIS

Given an undirected graph $G = (V, E)$ where V is its set of nodes and E its set of edges, a subset of nodes $S \subseteq V$ is an *independent set* if the elements of S are pairwise nonadjacent in G. If $w(v)$ is the weight of node $v \in V$, the weight of independent set S is $W(S) = \sum_{v \in S} w(v)$. In the *maximum weight independent set* (MWIS) problem we seek an independent set S^* such that $W(S^*) \geq W(S)$ for all independent sets $S \subseteq V$ in G [1]. This optimization problem is NP-hard [2] and it is often solved using heuristic algorithms [3].

We provide a collection of instances of an MWIS problem that appeared as subproblems in algorithms solving real-life long-haul vehicle routing problems at Amazon. Our goal is to enhance the set of benchmark instances available to algorithm researchers working on MWIS. Our instances differ from other publicly available instances and the new collection includes some large instances.

To gain intuition into the application, consider a stochastic heuristic for the problem. This heuristic produces different solutions for different pseudo-random generator seeds. Each solution consists of a set of routes. We want to recombine routes from multiple solutions to obtain a better solution.

Each *route* consists of a driver and a set of loads assigned to the driver. A subset of routes is *feasible* if no two routes in the subset share a driver or a load. Each route has a weight. The objective function is the sum of route weights. The problem is to find a feasible solution of the maximum total weight.

To state this problem as MWIS, we build a *conflict graph* as follows. Nodes of the graph correspond to routes and weights correspond to route weights. We connect two nodes by an edge if the corresponding routes have a conflict, i.e., they share a driver or a load.

Our application has additional information that one can (optionally) use in an algorithm. First, we have a good initial solution, the best of the solutions we combine. We provide initial solutions for our instances. One can use this solution to possibly warm-start a MWIS algorithm.

Second, we have information about many cliques in the conflict graph. For a fixed load (or driver), nodes corresponding to the routes containing the load (driver) form a clique: every pair of such nodes is connected. This allows us to use the well-known clique integer linear programming (ILP) formulation of the problem:

$$\max \sum_{v \in V} w_v x_v$$

$$\text{subject to}$$

$$C_2, C_3, \ldots, C_k,$$
$$x_v \in \{0, 1\}, \forall\, v \in V,$$

where $C_2, C_3, \ldots, C_k$ are, respectively, the sets of 2-clique, 3-clique, …, and k-clique inequalities. In general, for maximal cliques Q of size k, we have the set of k-clique inequalities

$$\sum_{v \in Q} x_v \leq 1, \text{ for all maximal cliques Q of size k}.$$

One can solve a linear programming (LP) relaxation of the problem, which assigns each node a value in the closed real interval [0, 1]. Note that the objective function of the LP relaxation provides an upper bound on the corresponding MWIS solution value. We provide both the cliques and the relaxed LP solutions with our instances.

Table 1 lists the instances we provide and includes the graph size, the initial solution value, and the relaxed LP bound. To compute the LP bounds we did not make use of all k-clique inequalities, for $k > 2$ but rather only a subset of them, specified in instance file `cliques.txt`.

2 Input Graph Format

We place each instance in a separate directory containing several files with instance name, graph edge set, node weights, clique information, and relaxed LP solution values. Directory names correspond to the instance names. Next we describe the file formats.

For an undirected, node-weighted graph $G = (V, E, w)$ with n nodes, m edges and integral node IDs from $[1, n]$, we give the following files:

- `instance_name.txt` - Name of the instance.
- `conflict_graph.txt` - Edges of G. The file has a total of $m + 1$ lines. The first line gives the numbers of nodes and edges: "n m". Each of the lines $2, \dots, m + 1$ describes an edge $e = (u, v) \in E$ as "u v".
- `node_weights.txt` - Node weights. The file has a total of n lines, each describing the weight of node $v \in V$ as "v $w(v)$". The weights are integers.
- `solution.txt` - Initial solution for warm start. It contains one line per node in the initial solution, giving its node index: if a node v is in the solution, the file contains a line with "v" in it.
- `cliques.txt` - Set of cliques in G. For each clique $C = \{c_1, c_2, \dots, c_k\}$, the file contains one line as "c_1 c_2 $\dots$ c_k".
- `lploads.txt` - Solution for the relaxed LP problem for the MWIS problem on the clique graph, where each node $v \in V$ has a relaxed LP value $l(v) \in [0, 1]$. The file has n lines, each with the LP value of a node $v \in V$ as "v $l(v)$", where $l(v)$ is a floating point number.

The files conflict_graph.txt and node_weights.txt are needed by any MWIS algorithm. The other files are optional.

Note that some of our graphs are large, with the compressed tar file being over 1 GB in size. To represent the total weight of a solution, 32-bit integers are insufficient. An implementation needs to use 64-bit integers or doubles to represent the weight of these independent sets.

Table 1 List of VR instances in the library. For each of the 38 instances, the table lists the instance name, the number of nodes and edges in the conflict graph, the total weight of a starting solution, the linear programming (LP) upper bound, the compressed tar files of the directory with the files that define the instance, and the size (in Mbytes) of the compressed tar file

Instance	$\|V\|$	$\|E\|$	Initial Sol.	LP bound	Filename	Mbytes
MT-D-01	979	3841	228874404	238166485	`MT-D-01.tar.gz`	0.03
MT-D-200	10880	547529	286750411	287228467	`MT-D-200.tar.gz`	1.77
MT-D-FN	10880	645026	290723959	290881566	`MT-D-FN.tar.gz`	2.07
MT-W-01	1006	3140	299132358	312121568	`MT-W-01.tar.gz`	0.03
MT-W-200	12320	554288	383620215	384099118	`MT-W-200.tar.gz`	1.86
MT-W-FN	12320	593328	390596383	390869891	`MT-W-FN.tar.gz`	1.97
MR-D-01	14058	60738	1664446852	1695332636	`MR-D-01.tar.gz`	0.48
MR-D-03	21499	168504	1739544141	1763685757	`MR-D-03.tar.gz`	0.97
MR-D-05	27621	295700	1775123794	1796703313	`MR-D-05.tar.gz`	1.35
MR-D-FN	30467	367408	1794070793	1809854459	`MR-D-FN.tar.gz`	1.75
MR-W-FN	15639	267908	5386472651	5386842781	`MR-W-FN.tar.gz`	1.18
MW-D-01	3988	19522	465730126	477563775	`MW-D-01.tar.gz`	0.14
MW-D-20	20054	718152	522485254	531510712	`MW-D-20.tar.gz`	2.50
MW-D-40	33563	2169909	533938531	543396252	`MW-D-40.tar.gz`	7.20
MW-D-FN	47504	4577834	542182073	549872520	`MW-D-FN.tar.gz`	15.17
MW-W-01	3079	48386	1268370807	1270311626	`MW-W-01.tar.gz`	0.21
MW-W-05	10790	789733	1328552109	1334413294	`MW-W-05.tar.gz`	2.49
MW-W-10	18023	2257068	1342415152	1360791627	`MW-W-10.tar.gz`	6.76
MW-W-FN	22316	3495108	1350675180	1373020454	`MW-W-FN.tar.gz`	10.41
CW-T-C-1	266403	162263516	1298968	1353493	`CW-T-C-1.tar.gz`	547.73
CW-T-C-2	194413	125379039	933792	957291	`CW-T-C-2.tar.gz`	417.49
CW-T-D-4	83091	43680759	457715	463672	`CW-T-D-4.tar.gz`	140.88
CW-T-D-6	83758	44702150	457605	463946	`CW-T-D-6.tar.gz`	143.95
CR-T-C-1	602472	216862225	4605156	4801515	`CR-T-C-1.tar.gz`	746.32
CR-T-C-2	652497	240045639	4844852	5032895	`CR-T-C-2.tar.gz`	828.21
CR-T-D-4	651861	245316530	4789561	4977981	`CR-T-D-4.tar.gz`	845.85
CR-T-D-6	381380	128658070	2953177	3056284	`CR-T-D-6.tar.gz`	441.42
CR-T-D-7	163809	49945719	1451562	1469259	`CR-T-D-7.tar.gz`	168.95
CW-S-L-1	411950	316124758	1622723	1677563	`CW-S-L-1.tar.gz`	1071.34
CW-S-L-2	443404	350841894	1692255	1759158	`CW-S-L-2.tar.gz`	1192.32
CW-S-L-4	430379	340297828	1709043	1778589	`CW-S-L-4.tar.gz`	1156.28
CW-S-L-6	267698	191469063	1159946	1192899	`CW-S-L-6.tar.gz`	644.49
CW-S-L-7	127871	89873520	589723	599271	`CW-S-L-7.tar.gz`	294.53
CR-S-L-1	863368	368431905	5548904	5768579	`CR-S-L-1.tar.gz`	1271.78
CR-S-L-2	880974	380666488	5617351	5867579	`CR-S-L-2.tar.gz`	1314.11
CR-S-L-4	881910	383405545	5629351	5869439	`CR-S-L-4.tar.gz`	1323.34
CR-S-L-6	578244	245739404	3841538	3990563	`CR-S-L-6.tar.gz`	845.81
CR-S-L-7	270067	108503583	1969254	2041822	`CR-S-L-7.tar.gz`	370.47

3 Downloading the Instances

The full set of 38 instances can be downloaded as gzipped tar files from the AWS OpenData site:

https://registry.opendata.aws/mwis-vr-instances/

using the AWS command line interface (CLI) [4].

Instruction on installation of AWS CLI can be found in [4]. As an example, installation on MacOS can be done using Terminal with the commands

`curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg" sudo installer -pkg AWSCLIV2.pkg -target /`

To list the contents of the repository using AWS CLI, run the command

```
aws s3 ls s3://mwis-vr-instances/ --no-sign-request
```

To download an instance, say file MT-W-01.tar.gz, from the repository using AWS CLI, run the command

```
aws s3 cp s3://mwis-vr-instances/MT-W-01.tar.gz .
--no-sign-request
```

4 Concluding Remarks

In this paper, we introduce a set of large-scale maximum weight independent set instances arising in a real-world vehicle routing application. Our goal in making these instances available to other researchers is that progress can be made in the field. Other researchers can try their existing MWIS solvers on these instances and can be motivated to develop new solvers for them.

Acknowledgements On behalf of all authors, the corresponding author (MGCR) states that there is no conflict of interest, no funding, and the data is freely available.

References

1. Butenko S (2003) Maximum independent set and related problems with applications. PhD thesis, U. of Florida, Gainesville, Florida
2. Garey M, Johnson D (1979) Computers and Intractability: A guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco
3. Pelillo M (2009) Heuristics for maximum clique and independent set. In: Floudas C, Pardalos P (eds) Encyclopedia of Optimization. Springer, US, Boston, MA, pp 1508–1520

4. AWS (2021) AWS Command Line Interface. https://docs.aws.amazon.com/cli/index.html. Last visited May 27, 2021