NEW INSTANCES FOR MAXIMUM WEIGHT INDEPENDENT SET FROM A VEHICLE ROUTING APPLICATION

YUANYUAN DONG, ANDREW V. GOLDBERG, ALEXANDER NOE, NIKOS PAROTSIDIS, MAURICIO G.C. RESENDE, AND QUICO SPAEN

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.

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. This optimization problem is NP-hard (Garey and Johnson, 1979) and it is often solved using heuristic algorithms.

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

Date: May 26, 2021– Parts of this paper were written while the first, third, and fourth authors were employed at Amazon.com.

Key words and phrases. Maximum weight independent set, Problem instances, Experimental algorithms.

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	3 841	228 874 404	238 166 485	MT-D-01.tar.gz	0.03
MT-D-200	10 880	547 529	286 750 411	287 228 467	MT-D-200.tar.gz	1.77
MT-D-FN	10 880	645 026	290 723 959	290 881 566	MT-D-FN.tar.gz	2.07
MT-W-01	1 006	3 140	299 132 358	312 121 568	MT-W-01.tar.gz	0.03
MT-W-200	12 320	554 288	383 620 215	384 099 118	MT-W-200.tar.gz	1.86
MT-W-FN	12 320	593 328	390 596 383	390 869 891	MT-W-FN.tar.gz	1.97
MR-D-01	14 058	60 738	1 664 446 852	1 695 332 636	MR-D-01.tar.gz	0.48
MR-D-03	21 499	168 504	1 739 544 141	1 763 685 757	MR-D-03.tar.gz	0.97
MR-D-05	27 621	295 700	1 775 123 794	1 796 703 313	MR-D-05.tar.gz	1.35
MR-D-FN	30 467	367 408	1 794 070 793	1 809 854 459	MR-D-FN.tar.gz	1.75
MR-W-FN	15 639	267 908	5 386 472 651	5 386 842 781	MR-W-FN.tar.gz	1.18
MW-D-01	3 988	19 522	465 730 126	477 563 775	MW-D-01.tar.gz	0.14
MW-D-20	20 054	718 152	522 485 254	531 510 712	MW-D-20.tar.gz	2.50
MW-D-40	33 563	2 169 909	533 938 531	543 396 252	MW-D-40.tar.gz	7.20
MW-D-FN	47 504	4 577 834	542 182 073	549 872 520	MW-D-FN.tar.gz	15.17
MW-W-01	3 079	48 386	1 268 370 807	1 270 311 626	MW-W-01.tar.gz	0.21
MW-W-05	10 790	789 733	1 328 552 109	1 334 413 294	MW-W-05.tar.gz	2.49
MW-W-10	18 023	2 257 068	1 342 415 152	1 360 791 627	MW-W-10.tar.gz	6.76
MW-W-FN	22 316	3 495 108	1 350 675 180	1 373 020 454	MW-W-FN.tar.gz	10.41
CW-T-C-1	266 403	162 263 516	1 298 968	1 353 493	CW-T-C-1.tar.gz	547.73
CW-T-C-2	194 413	125 379 039	933 792	957 291	CW-T-C-2.tar.gz	417.49
CW-T-D-4	83 091	43 680 759	457 715	463 672	CW-T-D-4.tar.gz	140.88
CW-T-D-6	83 758	44 702 150	457 605	463 946	CW-T-D-6.tar.gz	143.95
CR-T-C-1	602 472	216 862 225	4 605 156	4 801 515	CR-T-C-1.tar.gz	746.32
CR-T-C-2	652 497	240 045 639	4 844 852	5 032 895	CR-T-C-2.tar.gz	828.21
CR-T-D-4	651 861	245 316 530	4 789 561	4 977 981	CR-T-D-4.tar.gz	845.85
CR-T-D-6	381 380	128 658 070	2 953 177	3 056 284	CR-T-D-6.tar.gz	441.42
CR-T-D-7	163 809	49 945 719	1 451 562	1 469 259	CR-T-D-7.tar.gz	168.95
CW-S-L-1	411 950	316 124 758	1 622 723	1 677 563	CW-S-L-1.tar.gz	1 071.34
CW-S-L-2	443 404	350 841 894	1 692 255	1 759 158	CW-S-L-2.tar.gz	1 192.32
CW-S-L-4	430 379	340 297 828	1 709 043	1 778 589	CW-S-L-4.tar.gz	1 156.28
CW-S-L-6	267 698	191 469 063	1 159 946	1 192 899	CW-S-L-6.tar.gz	644.49
CW-S-L-7	127 871	89 873 520	589 723	599 271	CW-S-L-7.tar.gz	294.53
CR-S-L-1	863 368	368 431 905	5 548 904	5 768 579	CR-S-L-1.tar.gz	1 271.78
CR-S-L-2	880 974	380 666 488	5 617 351	5 867 579	CR-S-L-2.tar.gz	1 314.11
CR-S-L-4	881 910	383 405 545	5 629 351	5 869 439	CR-S-L-4.tar.gz	1 323.34
CR-S-L-6	578 244	245 739 404	3 841 538	3 990 563	CR-S-L-6.tar.gz	845.81
CR-S-L-7	270 067	108 503 583	1 969 254	2 041 822	CR-S-L-7.tar.gz	370.47

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:

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

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 cliques Q of size k, we have the set of k-clique inequalities

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

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.

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 $2, \ldots, m+1$ 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 in the solution, the file contains a line with "v" in it.
- `cliques.txt` – Set of cliques in G. For each clique $C = \{c_1, c_2, \ldots, c_k\}$, the file contains one line as "c_1 c_2 ... 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 Gbyte in size. 32-bit integers are insufficient to represent the total weight of a solution. An implementation needs to use 64-bit integers or doubles to represent the weight of these independent sets.

3. Links to Instances

The full set of 38 instances are downloaded at `http://https://registry.opendata.aws/mwis-vr-instances/` as gzipped tar files.

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.

References

M.R. Garey and D.S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Company, San Francisco, 1979.

(Yuanyuan Dong) Dallas, TX USA. The work was done while the author was at Amazon.com
Email address: njdyy03@gmail.com

(Andrew V. Goldberg) Amazon.com, East Palo Alto, CA USA.
Email address: avg@alum.mit.edu

(Alexander Noe) University of Vienna, Vienna, Austria. The work was done while the author was at Amazon.com
Email address: alexander.noe@univie.ac.at

(Nikos Parotsidis) Google, Zurich, Switzerland. The work was done while the author was at Amazon.com
Email address: nickparo1@gmail.com

(Mauricio G.C. Resende) Amazon.com and Industrial & Systems Engineering, University of Washington, Seattle, WA USA.
Email address: mgcr@berkeley.edu, mgcr@uw.edu

(Quico Spaen) Amazon.com, East Palo Alto, CA USA.
Email address: qspaen@berkeley.edu