DAA Assignment 2

Report

Algorithm 1 - Exact

This algorithm finds a **Connected Dominating Set (CDS)** in a graph **G=(V,E)** using support from an auxiliary graph **Ψ=(VΨ,EΨ)**. It starts by setting two bounds, a lower bound **l** and an upper bound **u,** where **u** is the highest degree of any vertex with respect to **Ψ.** It also finds all **(h−1)** cliques in **G,** stored in a set **Λ.**

The algorithm then runs a binary search. In each step, it picks a midpoint **α** and builds a flow network with a source **s,** a sink **t,** the vertices of **G,** and the cliques in **Λ.** Edges are added with specific capacities based on degrees, clique structures, and the value of **α.** It then finds the minimum **s-t** cut in this network.

If the cut separates only the source node, it means the guess was too large, and the upper bound is reduced. Otherwise, the lower bound is raised, and the vertices on the source side (excluding **s**) form the current candidate for the CDS. This process repeats until the bounds converge, resulting in an exact connected dominating set.

Algorithm 2 - CoreExact

This algorithm finds a **Connected Dominating Set (CDS)** by first performing a **core decomposition** of the graph **G=(V,E)** with help from an auxiliary graph **Ψ=(VΨ,EΨ).** It locates the **(k″,Ψ)-core** by using pruning criteria to focus only on the denser parts of the graph. Then, it initializes empty sets for connected components **C, t**he final solution **D,** and a temporary set **U.** It also sets two bounds, a lower bound **l** starting from the initial core density **ρ",** and an upper bound **u** based on the maximum degree in the graph.

The algorithm collects all connected components of the **(k″,Ψ)-core.** For each component, if needed, it intersects it further with a smaller core. It then builds a flow network by following the edge-building rules from **Algorithm 1 - Check** and finds a minimum **s-t** cut. If the cut is empty, it skips to the next component.

If the cut is meaningful, it uses **binary search** between **l and u** to fine-tune the solution. In each iteration, it builds a new flow network, finds the minimum cut, and updates **l and u** based on whether the cut separates only the source or a useful group of nodes. If a better connected structure is found, it updates the CDS candidate **D.** After all components are processed, the algorithm returns the final connected dominating set.

Algorithm 1 - Exact

Dataset	H-Value	No.of (h -1) cliques	No.of Triangles	No.of edges in densest subgraph	No. of vertices in Densest Subgraph	Runtime	Density
AS-733	3	3132	2503	226	28	61.3549s	8.071
CA - Hepth	3	25973	28339	496	32	619.24s	15.5
Yeast	3	1948	206	19	7	5.26113s	2.714

Algorithm 4 - CoreExact

Dataset	H-Value	No.of (h -1) cliques	No.of Triangles	No.of edges in densest subgraph	No. of vertices in Densest Subgraph	Runtime	Density
AS-733	3	3132	2503	556	168	0.4194s	3.3095
CA - Hepth	3	25973	28339	22842	6882	2.475s	3.3191
Yeast	3	1948	206	406	236	0.07966s	1.7203
AS-caida	3	52861	34265	26549	8393	32.0799s	3.1632