

Exportar Grafos en formatos: GDF, GraphSON, GraphML , GEXF

Roberto Alejandro Gutiérrez Guillén
Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Santa Fe
Mexico
A01019608@itesm.mx

1. GITHUB

<https://github.com/axgtz/Analisis-y-Diseño-de-Algoritmos/tree/master/ExportSNAP/Snap-4.0/examples/testGuti>

2. INTRODUCCIÓN

Los grafos son una estructura de datos de la computación. Son un conjunto de nodos (vértices) que representan a un objeto, y están unidos por aristas. Puede que las uniones sean dirigidas o no, también, pueden tener peso y otros atributos. Son usados para muchas cosas, por ejemplo para mostrar un mapa de amistades en Facebook o una red de computadoras.

3. PROCEDIMIENTO

4. VENTAJAS Y DESVENTAJAS

4.1. GDF

Este formato está basado en el formato CSV (Comma Separated File), por lo tanto es muy fácil de leer y de convertir a CSV. Además fue el formato más rápido de exportar por su simple estructura.

4.2. GEXF

Este formato es muy usado en la industria, por lo tanto es muy maduro y robusto, por lo que da confianza. Es muy parecido a GraphML ya que también se basa en XML pero es menos utilizado en la industria.

4.3. GraphML

GraphML es un formato para almacenar datos basado en XML, y soporta menos información adicional que sus rivales, esto ayuda en su facilidad de exportar pero no se pueden incluir grafos complejos.

4.4. GraphSON

Una gran desventaja es que la última versión de Gephi, una aplicación usada para visualizar grafos, no soporta este formato en su última versión. Es el formato más complejo de exportar porque no existe un estándar pero esto tiene la ventaja de que puede ser adaptado a cualquier situación.

5. COMPLEJIDADES TEMPORAL Y ESPACIAL

5.1. Complejidad Temporal

Los cuatro formatos de exportación tienen la misma complejidad temporal, ya que guardan todos los nodos y todos los vertices, y se tienen que recorrer todos para ser exportados, por lo tanto complejidad temporal de los cuatro algoritmos es de $O(V+E)$.

5.2. Complejidad Espacial

La complejidad espacial es $O(V+E)$, la suma de la cantidad de nodos y uniones.

6. TIEMPOS DE EJECUCIÓN

Tiempo de ejecución GDF:	0.0369601 segundos
Tiempo de ejecución GEXF:	0.0503775 segundos
Tiempo de ejecución GraphML:	0.0511828 segundos
Tiempo de ejecución GraphSON:	0.0392701 segundos

7. GEPHI

8. APENDICE

A. FUNCIÓN PARA EXPORTAR A GDF

```
void exportarGDF(PNGraph Graph){
    ofstream archivo_salida;
    archivo_salida.open("grafoGnutella.gdf");

    if (archivo_salida.is_open()){
        archivo_salida << "nodedef>name\n";
        for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++){
            archivo_salida << NI.GetId() << ",\n";
        }

        archivo_salida << "edgedef>node1,node2\n";
        for (TNGraph::TEdgeI EI = Graph->BegEI(); EI < Graph->EndEI(); EI++){
            archivo_salida << EI.GetSrcNId() << "," << EI.GetDstNId() << "\n";
        }

        archivo_salida.close();
    }
}
```

B. FUNCIÓN PARA EXPORTAR A GRAPHSON

```
void exportarGrapSON(PNGraph Graph){
    ofstream archivo_salida;
    archivo_salida.open("grafoGnutella.json");

    if (archivo_salida.is_open()){
        archivo_salida << "{ \"graph\": {\n";
    }
}
```

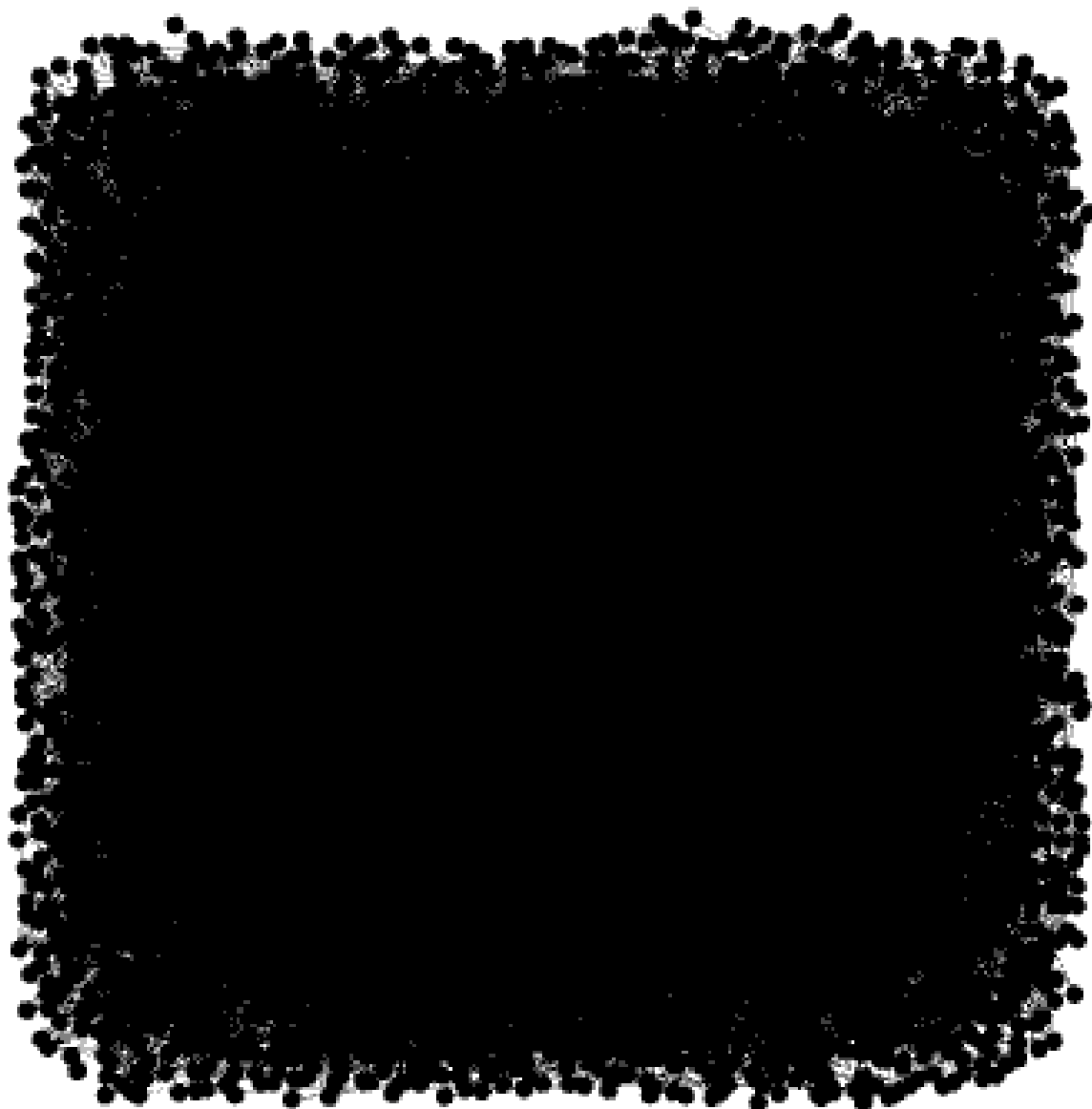


Fig. 1. Resultado de visualizacin en Gephi

```

archivo_salida << "\\nodes\\": [\\n";

for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++){
    archivo_salida << "{\\n \\\"_id\\\": \\\"\" << NI.GetId() << "\\\" }";

    if (NI == Graph->EndNI()){//Si es el ultimo nodo
        archivo_salida << " ],\\n";
    }else{//Todos los demas nodos
        archivo_salida << ",\\n";
    }
}

archivo_salida << "\\edges\\": [\\n";
for (TNGraph::TEdgeI EI = Graph->BegEI(); EI < Graph->EndEI(); EI++){
    archivo_salida << "{ \\\"source\\\": \\\"\" << EI.GetSrcNId() << "\\\", \\\"target\\\"";

    if (EI == Graph->EndEI()){
        archivo_salida << " ]\\n";
    }else{
        archivo_salida << ",\\n";
    }
}
archivo_salida << "} }";

}
}

```

C. FUNCIÓN PARA EXPORTAR GRAPHML

```

void exportarGraphMl(PNGraph Graph){
    ofstream archivo_salida;
    archivo_salida.open("grafoGnutella.graphml");

    if (archivo_salida.is_open()){
        archivo_salida << "<?xml version=\\\"1.0\\\" encoding=\\\"UTF-8\\\"?>\\n";
        archivo_salida << "<graphml xmlns=\\\"http://graphml.graphdrawing.org/xmlns\\\"";
        archivo_salida << "<graph id=\\\"G\\\" edgedefault=\\\"directed\\\">\\n";

        for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++){
            archivo_salida << "<node id=\\\"n\\\" << NI.GetId() << "\\\"/>\\n";
        }
        int i = 0;
        for (TNGraph::TEdgeI EI = Graph->BegEI(); EI < Graph->EndEI(); EI++,i++){
            archivo_salida << "<edge id=\\\"e\\\" << i << "\\\" source=\\\"n\\\" << EI.GetSrcNId()";
        }

        archivo_salida << "</graph>\\n";
        archivo_salida << "</graphml>\\n";
        archivo_salida.close();
    }
}

```

```
}
```

D. FUNCIÓN PARA EXPORTAR A GEXF

```
void exportarGEXF(PNGraph Graph){//
    ofstream archivo_salida;
    archivo_salida.open("grafoGnutella.gexf");

    if (archivo_salida.is_open()){
        archivo_salida << "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n";
        archivo_salida << "<gexf xmlns=\"http://www.gexf.net/1.2 draft\" version=\"1.2\"";
        archivo_salida << "<graph mode=\"static\" defaultedgetype=\"directed\">\n";
        archivo_salida << "<nodes>\n";

        for (TNGraph::TNodeI NI = Graph->BegNI(); NI < Graph->EndNI(); NI++){
            archivo_salida << "<node id=\"" << NI.GetId() << "\" />\n";
        }

        archivo_salida << "</nodes>\n";
        archivo_salida << "<edges>\n";
        int i = 0;
        for (TNGraph::TEdgeI EI = Graph->BegEI(); EI < Graph->EndEI(); EI++,i++){
            archivo_salida << "<edge id=\"" << i << "\" source=\"" << EI.GetSrcNI()
        }
        archivo_salida << "</edges>\n";
        archivo_salida << "</graph>\n";
        archivo_salida << "</gexf>\n";
        archivo_salida.close();
    }
}
```

REFERENCES

- Stanford. 2002. Gnutella peer-to-peer network: August 4 2002. Retrieved October,2017 from <https://snap.stanford.edu/data/p2p-Gnutella04.html>.
- Dan LaRocque. 2014. GraphSON Format. Retrieved October,2017 from <https://github.com/thinkaurelius/faunus/wiki/GraphSON-Format>
- Charles Iliya Krempeaux. 2013. GDF: A CSV Like Format For Graphs. Retrieved October,2017 from <http://changelog.ca/log/2013/03/09/gdf>.
- Gephi. 2017. GraphML Format. Retrieved October, 2017 from <https://gephi.org/users/supported-graph-formats/graphml-format/>
- Gephi. 2017. GEXF File Formatt. Retrieved October, 2017 from <https://gephi.org/gexf/format/>