

Bipartite Create a bipartite graph with n nodes and average density p . The graph is split into two approximately equal-sized color classes.

Complete bipartite Create a complete bipartite graph with n nodes.

Grid Create a grid of size $\lfloor \sqrt{n} \rfloor$.

Planar Create a planar graph with n nodes.

Triangulated planar Create a triangulated planar graph with n nodes.

Disk Create a disk graph with n nodes and disk diameters between d_{\min} and d_{\max} . Two nodes v and w are assumed to be adjacent iff their Euclidean distance is at most $\frac{d_v + d_w}{2}$, where, for each v , d_v denotes the disk diameter generated for node v . The value $d = 100$ corresponds to both the width and height of the graph window. Thus a minimum diameter above $\sqrt{2} \times 100$ will produce complete graphs. (However, all calculations are done using double precision floating point values, so rounding errors are possible.)

14.2 Create Services Submenu

This menu contains operations for creating services, requirements and ensemble assignments. These operations are controlled by means of the following parameters in the service options panel: r (number of services), m (default service size), m_1 and m_2 (minimum and maximum service size), r_1 and r_2 (minimum and maximum number of required services per node), n_1 and n_2 (minimum and maximum size of service “supply areas,” where the supply area of a service s is defined as the set V_s of all nodes where s is required, i.e., $V_s = \{v \in V : s \in R_v\}$). Besides this, the order of services can be controlled with the ‘service order’, ‘user order’ and ‘reverse’ options. Services can be ordered by their numbers (‘def’ = default), at random (‘rand’), by sizes (largest first; ‘size’), by supply area size (largest first; ‘area’) or using an arbitrary order specified by the user (‘user’). The order is reversed if the ‘reverse’ option is enabled. Note that the current requirements and ensembles are always kept sorted w.r.t. the current order (the ensemble lists are ordered lexicographically).

New services Create r services with size m .

Random services Create r services with random sizes between m_1 and m_2 .

Random sizes Assign random sizes between m_1 and m_2 to the current services. (Works like ‘Random services’, but keeps the current service set.)

Empty requirements Assign empty requirements to all nodes of the graph.