2. Since it is of no special relevance for the understanding of the algorithmic description of our TS, and to keep things simple, we have not worked out the details of tabu list administration here.

3.8 Test Results

To validate our TS solution approach and its behaviour in comparison to SFF and GFF, we have implemented and tested the TS sketched out in this paper. The test scenario employed here will stay close to the test design, test description and test procedures as developed in [5],- with only some slight modifications -, where the characteristics of the SFF and GFF algorithms and their relation to the clique packing number have already been investigated in some detail. Though the test results presented here are still rather preliminary, and will have to be confirmed with more systematic studies in the future, they already show some interesting characteristics on how our TS approach positions in relation to SFF, GFF and the clique packing number. The tests were performed using randomly generated problem instances. As area graphs we chose so-called unit disk (UD) graphs, which, because of their geometric structure, are commonly employed as a simplified model of broadcast networks [4]. UD graphs are intersection graphs of equal-sized disks in the Euclidean plane. They are usually specified by a point set $V \subseteq \mathbb{R}^2$ and a diameter d. The edges of the graph are then those pairs vw of distinct vertices v, w for which $||v - w||_2 \leq d$, where $|| \cdot ||_2$ denotes the Euclidean norm in $\mathbb{R}^2$.

We performed two series of tests with diameters $d = 2500$ (graph density $\approx 15.6\%$) and $d = 4000$ (graph density $\approx 34.4\%$), where the point coordinates were generated uniformly and independently from the set $\{0, \ldots, 10000\} \times \{0, \ldots, 10000\}$. (This is slightly different from [5], where the point coordinates were taken from the set $[0, 1] \times [0, 1]$ and, in consequence, $d = 0.25$ and $d = 0.4$.) For the requirements we first generated a service set with uniformly distributed integer bandwidths in the interval $[1, 5]$. Then, we determined for each $v \in V$ the size of the corresponding requirement set R_v, again uniformly in the range $[1, 5]$, and selected a random set of this size from the service set. The maximum ensemble size was $M = 10$.

In each test series we varied the parameters n (number of vertices) and r (number of services) as follows: $n \in \{50, 100\}$, $r \in \{10, 20, 50, 100, 250\}$. For each of the 10 resulting parameter combinations we generated 20 random problem instances with the same number of vertices and services - i.e., a total of 200 instances in each series - for which we computed the following values :

- TS: number of colors calculated for the best ensemble assignment found by TS.
- GFF: number of colors calculated for a GFF ensemble assignment.
- SFF: number of colors calculated for a SFF ensemble assignment.
- $BEST$: $min\{TS, GFF, SFF\}$.
- S: lower bound on the clique packing number, using the generalized Carraghan/Pardalos algorithm [3] with the sum estimate for the packing number [5].

Both the SFF and GFF algorithms were invoked using a service order by decreasing sizes. Coloring was always done using the saturation algorithm [2]. In the case of SFF and GFF,