Finally, we wish to remark that the full version of our TS employs some more values besides $b_1(m)$, $b_2(m)$ and $b_3(m)$, but their objective is merely to bring greater refinement into move evaluation.

Furthermore, we have not yet pointed out how the original cost function f is integrated into "$\equiv$" and "$\lhd$". The easiest way to do this would probably be to introduce a fourth value $b_4(m)$, defined as $b_4(m) := f(\mathcal{B}) - f(m(\mathcal{B}))$, and then expanding the definition of "$\equiv$" and "$\lhd$" made above by now choosing i from the set $\{1, \ldots, 4\}$. Unfortunately, calculating $b_4(m)$ would imply coloring the ensemble graphs induced by $\mathcal{B}$ and especially by $m(\mathcal{B})$. Since, in terms of processor speed, this would be very time consuming if done for all $m \in M(\mathcal{B})$, we have utilized a more intelligent way of taking the cost function into account, yet too complex to be described here in detail.

3.5 Setting Tabus

As has already been mentioned in the discussion of the basic TS framework, we will stick close to the standard design of working with tabus, with the one exception that we will use a periodic and not a constant function to generate the tabu tenure values. Before we can go into details, we must first define, what in our problem context the elements are, which are apt to having a tabu set on them (and thus added to the tabu list). We have designed to set tabus on elements of the form (v, s), where $v \in V$ is a vertex and $s \in S$ is a service. Thus, if in a TS iteration we make the step implied by the move $m = (v, u, z, s) \in M(\mathcal{B})$, with $\mathcal{B} \in \mathbb{B}$ being the actual ensemble assignment of the TS iteration, we afterwards set a tabu on the element (v, s), preventing any further move of service s on vertex v for the time the tabu is valid. Note that the ensembles B_u and B_z, which are the source and target ensembles involved in the move m, are of no importance here. Only the action of having moved a service s on a vertex v is taken into account.
Setting a tabu demands assigning it a tabu tenure value expressing the tabu's duration validity. A constant function to generate the tabu tenure values, as done so in traditional TS design, does not suffice in our application. For on the one hand, choosing the constant too small involves the danger of loosing the capability to escape a local optimum once caught in it. On the other hand, choosing the constant too large might bring up the problem of iterating near a global optimum without the possibility of ever reaching it, which would be the case if always some vital moves are blocked by tabus. Experiments showed that we would always face the problem of having either chosen a too small or a too large tabu tenure constant, with always some TS iterations possible, for which the tabu tenure constant is of bad quality. In consequence, we have decided to use a periodic function, generating smaller and larger tabu tenure values in alternating phases. Going deeper into the details of the periodic function underlying our TS design is of no mentionable gain in insight due to the wide variety of possible and advantageous periodic functions thinkable,– though some might be of greater efficiency than others. Until now, there have been no deeper investigations into the nature of an adequate and optimized periodic function for our application.