applied in the last L steps. The procedure LegalOp checks if a new operator is legal given the current tabu list. The implementation of this procedure depends on the exact nature of operators we use. As in the basin-flooding approach, tabu search does not stop when it reaches a solution that cannot be improved, but rather continues the search with the hope of reaching a better structure. If this does not happen after a prespecified number of steps, we decide to abandon the search.

Algorithm A.7 Beam search

```
  Procedure Beam-Search (
    $\sigma_0$,   // initial candidate solution
    score,   // Score
    $\mathcal{O}$,   // A set of search operators
    $K$,   // Beam width
  )
1   Beam ← $\{\sigma_0\}$
2   while not terminated
3     $H \leftarrow \emptyset$   // Current successors
4     for each $\sigma \in L$ and each $o \in \mathcal{O}$
5       Add $o(\sigma)$ to $H$
6     Beam ← K-Best(score, $H$, $K$)
7   $\sigma_{\text{best}} \leftarrow$ K-Best(score, $H$, 1)
8   return ($\sigma_{\text{best}}$)
```

beam search

Another variant that forces a more systematic search of the space is *beam search*. In beam search, we conduct a hill-climbing search, but we keep track of a certain fixed number K of states. The value K is called the *beam width*. At each step in the search, we take all of the current states and generate and evaluate all of their successors. The best K are kept, and the algorithm repeats. The algorithm is shown in algorithm A.7. Note that with a beam width of 1, beam search reduces to greedy hill-climbing search, and with an infinite beam width, it reduces to breadth-first search. Note that this version of beam search assumes that the (best) steps taken during the search always improve the score. If that is not the case, we would also have to compare the current states in our beam *Beam* to the new candidates in H in order to determine the next set of states to put in the beam. The termination condition can be an upper bound on the number of steps or on the improvement achieved in the last iteration.

A.4.2.3 Randomization in Search

randomization

Another approach that can help in reducing the impact of local maxima is *randomization*. Here, multiple approaches exist. We note that most randomization procedures can be applied as a wrapper to a variety of local search algorithm, including both hill climbing and tabu search. Most simply, we can initialize the algorithm at different random starting points, and then use a hill-climbing algorithm from each one. Another strategy is to interleave random steps and hill-climbing steps. Here, many strategies are possible. In one approach, we can “revitalize” the search by taking the best network found so far and applying several randomly chosen operators