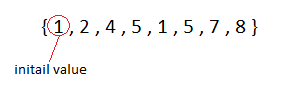
# *Variable neighborhood search (VNS)*

Variable neighborhood search has been proposed be P. Hansen and N. Mladenovic in 1997.

The basic idea of VNS is to successively explore a set of predefined neighborhoods to provide a better solution “note that we don’t say the actual solution”

It explores either at random or systematically a set of neighborhoods to get optimal solution.

* **This algorithm composed for three steps:**
  + Shaking
  + Local search
  + Move
* **What is objective from this algorithm?**
  + This algorithm used for found the optimal solution
* **How this algorithm works?** 
  + First, define a set of neighborhood
  + Take



* + Start searching from the first neighborhoods.
  + 1. Then take a random value **x’** then make local search up on it to produce **x’’** solution.
  + 2. And compare the target function with x’’ with target function with x if satisfied then make x = x’’ and repeat this process until break this condition then move to the next neighborhood set.
  + We move to the next neighborhood with the last value of x in the previse neighborhood.
  + 3. Repeat the process [1] and [2] for all the sets of neighborhood.
  + Because we determine the neighbor randomly in each neighborhood then we can optimize our solution by repeating the process [3] until termination criteria satisfied.
  + 4. The last step is return the optimal solution (x), or the solution that closer for the actual solution.
* **VNS algorithm.**
* **Shaking:** a random neighbor solution x’ is generated in the current neighborhood.
* **Local search:** is procedure that applied to the solution x’ to generate the solution x’’.
* **Moving:** “note that ***solution x’’ is better than x solution*** then the solution x’’ become the new current solution and the search starts from the current solution”. Then the ***moving*** happen when solution x’’ *is not better* than x solution to the next neighborhood , generate a new solution in this neighborhood and try to **improve** it.
* **Some of Problems that this algorithm solves.**