Algorithm: Generation of permutations in S_n in lexicographic order and optimization of a cost function over S_n

Input: $n \in \mathbb{N}$, $c: S_n \to \mathbb{R}$

Output: list L of permutations in S_n in lexicographic order; π^* such that $c(\pi^*) = \min_{\pi \in S_n} c(\pi)$

① Set $\pi(i) = i$, $\pi^*(i) = i$, $\forall\ i \in \{1, 2, \ldots, n\}$; $L = (\pi)$ list

Set $i := n-1$

② Set $k := \min \{\pi(i)+1, \ldots, n+1\} \setminus \{\pi(1), \ldots, \pi(i-1)\}$

③ If $k \leq n$ then do

$\pi(i) := k$

if $i = n$ then $L := (L, \pi)$. If $c(\pi) < c(\pi^*)$, set $\pi^* := \pi$

if $i < n$ then $\pi(i+1) := 0$ and $i := i+1$

end do

if $k = n+1$ then $i := i-1$

if $i \geq 1$ then go to ② otherwise STOP

④ Output L, π^*

Efficient implementation of ② for given π, k (~~fixed~~)

$\forall\ j = 1:n$ $\quad aux(j) := 0$

$\forall\ j = 1:i-1$ $\quad aux(\pi(j)) = 1$

$k := \pi(i) + 1$

while $k \leq n$ and $aux(k) = 1$ set $k := k+1$