Algorithm 1 CI-TGO($f(\cdot)$, $v(\cdot)$, $\boldsymbol{l}$, $\boldsymbol{u}$, $\boldsymbol{PS}$, $\boldsymbol{K}$, Max_LS, LS_1, LS_2, ϕ, α, β)

$\boldsymbol{best} \leftarrow random(\boldsymbol{l}, \boldsymbol{u})$;
$iteration \leftarrow 0$;
while $Converged(\boldsymbol{best}) ==$ **False do**
    $Populations \leftarrow CreatePop(\boldsymbol{l}, \boldsymbol{u}, \boldsymbol{PS}(1), 1.0)$;
    for $p \in \{1, ..., |\boldsymbol{PS}|\}$ **do**
        $NewPops \leftarrow \{\}$;
        for $pop \in Populations$ **do**
            $Topo_p \leftarrow TopographicalHeuristic(f(\cdot), v(\cdot), pop, \boldsymbol{K}(p), \alpha)$;
            if $p < |\boldsymbol{PS}|$ **then**
                for $\boldsymbol{x} \in Topo_p$ **do**
                    $NewPops \leftarrow NewPops \cup \{CreatePop(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{u}, \boldsymbol{PS}(p+1), \phi^p)\}$;
                end for
            else
                $TopoBest \leftarrow TopoBest \cup Topo_p$;
            end if
        end for
        if $p < |\boldsymbol{PS}|$ **then**
            $Populations \leftarrow NewPops$;
        end if
    end for
    $TopoBest \leftarrow \{\boldsymbol{x} = \boldsymbol{x}^1, ..., \boldsymbol{x}^{min(|TopoBest|, Max\_LS)} : \boldsymbol{x} \in sort(TopoBest)\}$;
    for $\boldsymbol{x} \in TopoBest$ **do**
        $\boldsymbol{x} \leftarrow LocalSearch(f(\cdot), v(\cdot), \boldsymbol{x}, LS_1)$;
        if $Compare(f(\cdot), v(\cdot), \boldsymbol{x}, \boldsymbol{best})$ **or** $f(\boldsymbol{x}) < f(\boldsymbol{best})$ **then**
            $\boldsymbol{x} \leftarrow LocalSearch(f(\cdot), v(\cdot), \boldsymbol{x}, LS_2)$;
            if $Compare(f(\cdot), v(\cdot), \boldsymbol{x}, \boldsymbol{best})$ **then**
                $\boldsymbol{best} \leftarrow \boldsymbol{x}$;
            end if
        end if
    end for
    $iteration \leftarrow iteration + 1$;
end while
return *best*;

Pseudocode for Constrained I-TGO.

Algorithm 2 TopographicalHeuristic($f(\cdot)$, $v(\cdot)$, $Population$, K, α)

$M \leftarrow |Population|$;
$\boldsymbol{best} \leftarrow Population(0)$;
$\boldsymbol{KNN}_K \leftarrow Build\_KNN(Population, K)$;
$\boldsymbol{R} \leftarrow random([0,1]^{M \times M})$;
$TopoBest \leftarrow \{\}$;
for $i \in \{1, ..., |\boldsymbol{Population}|\}$ **do**
    $insert = \boldsymbol{True}$;
    for $j \in \{1, ..., |\boldsymbol{PS}|\} \cap \{\boldsymbol{x}_j \in \boldsymbol{KNN}_K(\boldsymbol{x}_i)\}$ **do**
        if $\boldsymbol{R}_{i,j} < \alpha$ **then**
            $insert \leftarrow insert \;\&\; Compare(f(\cdot), v(\cdot), \boldsymbol{x}_i, \boldsymbol{x}_j)$;
        else
            $insert \leftarrow insert \;\&\; f(\boldsymbol{x}) < f(\boldsymbol{y})$;
        end if
    end for
    if $insert = \boldsymbol{True}$ **then**
        $TopoBest \leftarrow TopoBest \cup \{\boldsymbol{x}\}$;
    end if
    if $Compare(f(\cdot), v(\cdot), \boldsymbol{x}, \boldsymbol{best})$ **then**
        $\boldsymbol{best} \leftarrow \boldsymbol{x}$;
    end if
end for
if $|TopoBest| = 0$ **then**
    $TopoBest \leftarrow TopoBest \cup \{\boldsymbol{best}\}$;
end if
return $TopoBest$;

Creates topographical ordering and returns the best elements.

Algorithm 3 CreatePop($\boldsymbol{x}$, $\boldsymbol{l}$, $\boldsymbol{u}$, $popSize$, ϕ)

1: $\boldsymbol{l}' \leftarrow max(\boldsymbol{l}, \boldsymbol{x} - (0.5 * \phi) * (\boldsymbol{u} - \boldsymbol{l}))$;
2: $\boldsymbol{u}' \leftarrow min(\boldsymbol{u}, \boldsymbol{x} + (0.5 * \phi) * (\boldsymbol{u} - \boldsymbol{l}))$;

3: $Population \leftarrow \{\boldsymbol{x} = \boldsymbol{x}^1, ..., \boldsymbol{x}^{popSize} : \boldsymbol{x} \in random(\boldsymbol{l}', \boldsymbol{u}')\}$;

4: **return** $Population$;

Create a new population shrinked by ϕ around the point $\boldsymbol{x}$.