Algorithm 1 ALNS Algorithm for 2E-VRP-PDD

Input:

$\Omega_{instance}$: Problem instance (hub, satellites, customers, capacities)
Ω_{iter}: Total number of ALNS iterations
d_{rate}: Degree of destruction (fraction of customers to remove)
$T_{start}, T_{end}, c_{rate}$: Simulated Annealing parameters
$\sigma_1, \sigma_2, \sigma_3$: Operator scores for new best, better, and accepted solutions
r: Reaction factor for weight updates

Output: s_{best} ▷ The best solution state found

$s_{initial} \leftarrow$ Build Initial Solution($\Omega_{instance}$)
$s \leftarrow s_{initial}$ ▷ s is the current solution
$s_{best} \leftarrow s_{initial}$ ▷ s$_b$$estisthebest - so - farsolution$T $\leftarrow T_{start}$ ▷ Initialize temperature for Simulated Annealing
$\pi \leftarrow$ Initialize Operator Weights() ▷ Corresponds to Simple Weights() setup
for $i_{total} = 1$ to Ω_{iter} **do**
    $s' \leftarrow$ copy(s) ▷ Create a copy to modify
    $\Theta^- \leftarrow$ Select Destroy Operator(π) ▷ e.g., 'random$_c$$ustomer_r$$emoval$' $\Theta^+ \leftarrow$ Select Repair Operator(π) ▷ e.g., 'greedy$_c$$ustomer_i$$nsertion$'
    $s' \leftarrow$ Destroy(Θ^-, s', d_{rate})
    $s' \leftarrow$ Repair(Θ^+, s')
    reward $\leftarrow 0$
    if $f(s') < f(s_{best})$ **then** ▷ $f(s)$ is the total cost from 's.objective()'
        $s_{best} \leftarrow s'$
        $s \leftarrow s'$
        reward $\leftarrow \sigma_1$
    else if $f(s') < f(s)$ **then**
        $s \leftarrow s'$
        reward $\leftarrow \sigma_2$
    else if Acceptance Criteria(s', s, T) **then** ▷ Simulated Annealing check
        $s \leftarrow s'$
        reward $\leftarrow \sigma_3$
    $\pi \leftarrow$ Update Weights(π, Θ^-, Θ^+, reward, r) ▷ Update operator weights
    $T \leftarrow T \cdot c_{rate}$ ▷ Cool down temperature
return s_{best}