```
Generate initial population $P_0$ of size $N$ using Latin hypercube sampling
Rank and sort $P_0$ based on non-domination level
Create offspring population $Q_0$ using $k$ different algorithms. Each algorithm creates $(N/k)$ points
Set $t = 0$, and $T$ = maximum number of generations
**while** $t < T$ **do**
    $R_t = P_t \cup Q_t$
    Partition $R_t$ into different fronts $F_1, F_2, \ldots$
    Set $P_{t+1} = \varnothing$ and $i = 1$
    **while** $|P_{t+1}| < N$ **do**
        Calculate crowding distance in $F_i$
        **if** $|F_i| + |P_{t+1}| = \mathrm{N}$ **then**
            $P_{t+1} = P_{t+1} \cup F_i$
            **else**
                **if** $|F_i| + |P_{t+1}| \leq N$ **then**
                    Sort $F_i$ members in order of decreasing crowding distance
                    $P_{t+1} = P_{t+1}$ + the first $(N - |P_{t+1}|)$ elements of $F_i$
                **end if**
            **end if**
            $i = i + 1$
    **end while**
    Calculate crowded comparison operator $\forall i \in P_{t+1}$
    Calculate the number of offspring points, $P_{t+1}^{k}$ each of the $k$ algorithms contributed to $P_{t+1}$
    Set $j = 1$
    **while** $j \leq k$ **do**
        For adaptive offspring creation, calculate $N_{t+1}^{j} = N \cdot (P_{t+1}^{j} / N_{t}^{j}) / \sum_{w=1}^{k} (P_{t+1}^{w} / N_{t}^{w})$
        $j = j + 1$
    **end**
    Create $Q_{t+1}$ by generating $N_{t+1}^{k}$ offspring points with each of the $k$ individual algorithms
    $t = t + 1$
**end while**
```