▯→ Adaptive Particle swarm optimization for Feature Selection on High Dimensional Data , a.k.a Jump Local Optima Particle Swarm Optimization I

Adaptive Particle Swarm Optimization for Feature Selection on High Dimensional Data

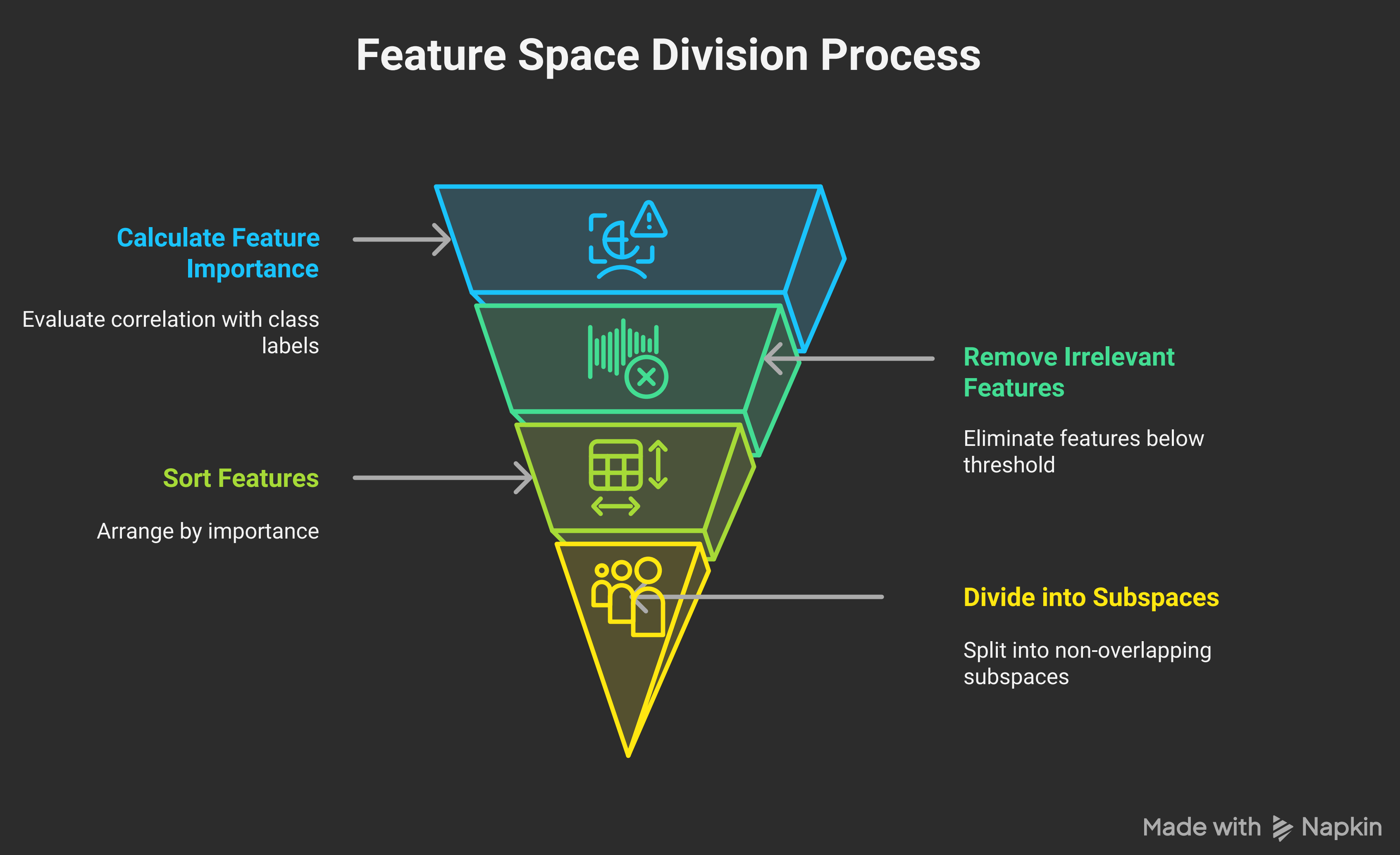
JLOPSO I: Algorithm Workflow

The **Jump Local Optima Particle Swarm Optimization I (JLOPSO I)** is an advanced PSO-based algorithm designed for feature selection on high-dimensional data. It improves upon standard PSO by splitting the problem into manageable sub-problems, adaptively adjusting swarm sizes, and incorporating a local search protocol to escape local optima.

The complete step-by-step workflow is as follows:

Step 1: Feature Space Division (Sub-Problem Splitting)

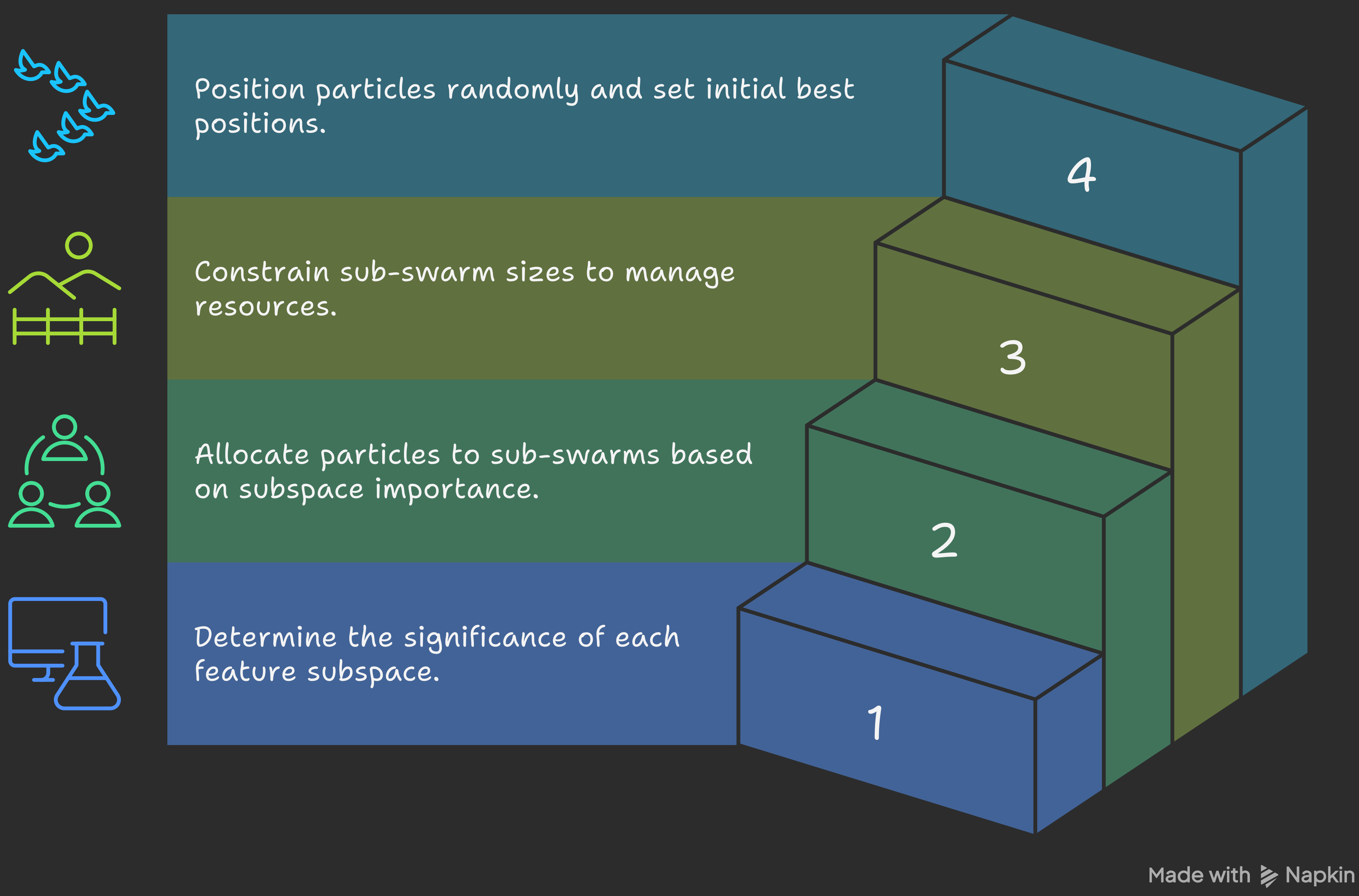
1. **Calculate Feature Importance:** The algorithm first evaluates the importance of every feature by calculating its **Symmetrical Uncertainty (SU)** with respect to the class labels. SU measures the correlation between a feature and the class.
2. **Remove Irrelevant Features:** Features with an SU value below a certain threshold (e.g., SU > 0) are considered weak or irrelevant and are removed from the feature set.
3. **Sort and Divide:** The remaining features are sorted in descending order based on their SU values. This sorted set is then uniformly divided into a predefined number of M non-overlapping feature subspaces. This "divide-and-conquer" strategy splits the high-dimensional problem into smaller, low-dimensional sub-problems.



Step 2: Sub-Swarm Initialization and Sizing

1. **Calculate Subspace Importance:** The importance of each feature subspace FI) is calculated as the average SU of all features within it.
2. **Set Initial Sub-Swarm Sizes:** The initial size (i.e., number of particles) for each sub-swarm is determined proportionally to the importance of its corresponding feature subspace.
3. **Bound Sub-Swarm Sizes:** To manage computational resources, the size of each sub-swarm is constrained within a predefined upper and lower bound.
4. **Initialize Particles:** For each sub-swarm, particles are initialized with random positions. Each particle's best-known position pbest) is set to its initial position, and the swarm's global best gbest) is determined.

Sub-Swarm Initialization Process



Step 3: Iterative Optimization

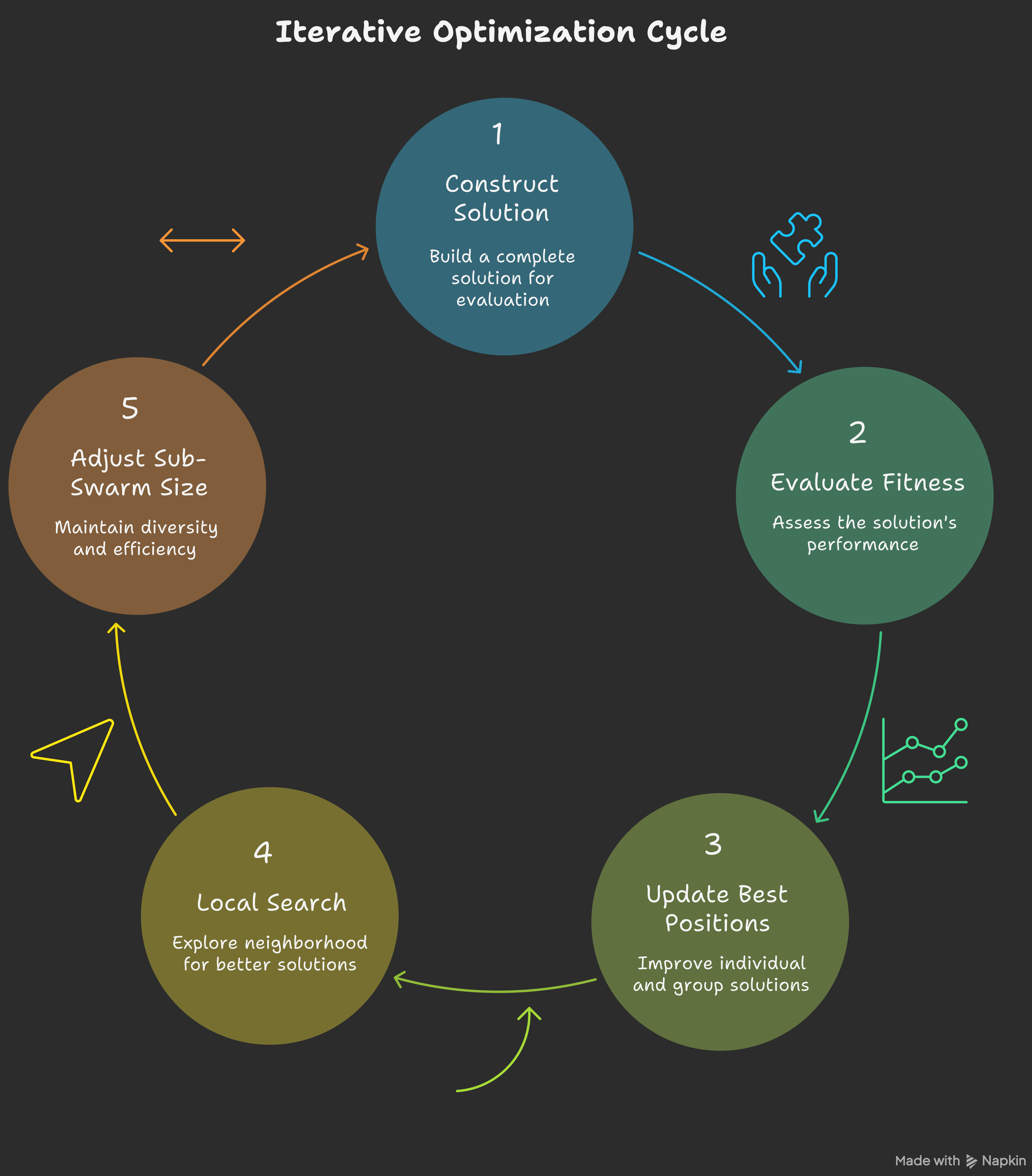
The algorithm iterates until a stopping criterion is met. In each iteration, the following steps are performed for every sub-swarm:

1. **Construct Solution & Evaluate Fitness:**
    * For each particle, a complete solution is constructed using an elite combination strategy. * This solution is decoded into a feature subset.
    The *fitness** of the subset is evaluated using a **K-Nearest Neighbors (KNN) classifier with Leave-One-Out Cross-Validation (LOOCV)**. The resulting classification accuracy serves as the fitness value.

2. **Update Best Positions:** If a particle finds a position with better fitness, its pbest is updated. If any particle's pbest is better than the current gbest of the sub-swarm, the gbest is updated.

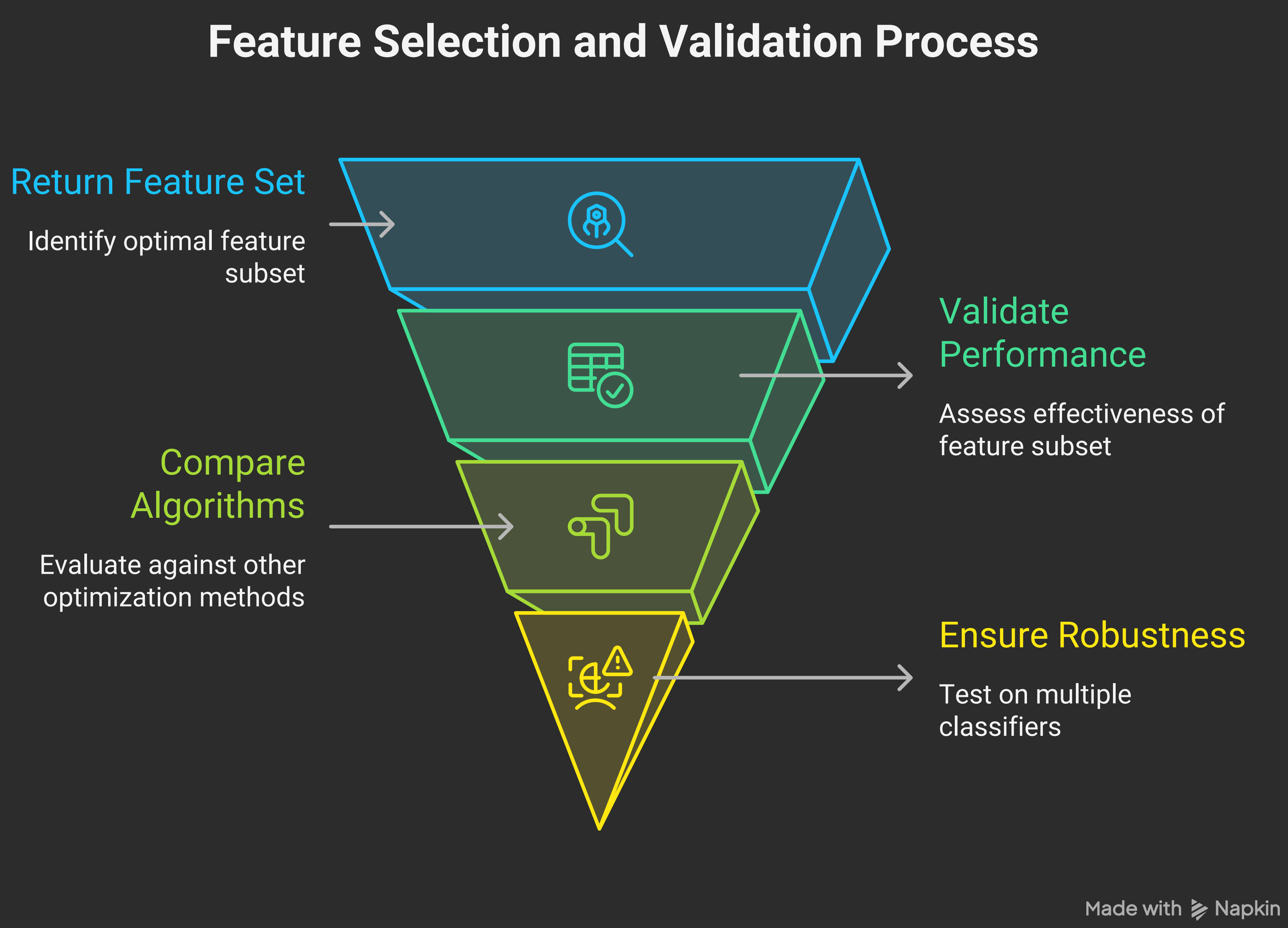
3. **Local Search Protocol (Local Search I):**
   * This strategy is invoked if the global best solution gbest) has not improved for a set number of iterations.
   * It explores the neighborhood of the current gbest by randomly adding potentially relevant features and removing potentially redundant ones to escape local optima. If a better solution is found, the gbest is updated.

4. **Adaptive Sub-Swarm Size Adjustment:**
   Periodically, the algorithm calculates the *relative convergence and divergence** of each sub-swarm.
   * Based on these metrics, particles may be automatically removed from a converging (less diverse) sub-swarm or new particles added to a diverging (more diverse) sub-swarm. This dynamic adjustment maintains diversity and reduces unnecessary computation.



Step 4: Final Output and Evaluation

1. **Return Feature Set:** Once the stopping criterion is met, the final gbest from the optimization process represents the selected optimal feature subset.
2. **Validate Performance:** The effectiveness of the selected feature subset is validated using key metrics: **number of selected features** and **classification accuracy**. The performance is typically compared against other algorithms like standard PSO, GA, and HHO on multiple classifiers (e.g., KNN and SVM) to ensure robustness.



JLOPSO I Feature Selection Process