1 Path Relinking

Given two solutions, s1 and s2, characterized by vertex colors in the range of 1 to k, the path relinking algorithm aims to find a path from one solution to the other. The algorithm iterates through the vertices, changing colors from s1 to s2. The algorithm selects the best solution encountered during this process.

```

function conflicts(graph, solution)
    conflicts_count = 0
    for edge in graph.edges
        if solution[edge[1]] == solution[edge[2]]
            conflicts_count += 1
        end
    end
    return conflicts_count
end

function path_relinking(graph, solution1, solution2)
    current_solution = copy(solution1)
    best_solution = copy(solution1)
    best_cost = conflicts(graph, solution1)

    for i in 1:length(graph.vertices)
        if current_solution[i] != solution2[i]
            current_solution[i] = solution2[i]
            current_cost = conflicts(graph, current_solution)

            if current_cost < best_cost
                best_cost = current_cost
                best_solution = copy(current_solution)
            end
        end
    end

    return best_solution
end
```

Listing 1: Particle Swarm Optimization in Julia