```
system := parse_and_load("system.xml")
scl := parse_and_load("scl.xml")
solutions := map(bay : ∅ for all bay in set(get_bays(scl)))

for all bay in set(get_bays(scl)) do
    ieds := ordered_set(get_ieds(scl, bay))
    fbs := ordered_set(get_fbs(system))

    product := cartesian_power( ieds, length(fbs) )
    solution_space := set(map_by_index(val, fbs) for all val in product)

    for all solution in solution_space do
        for all ied in keyset(solution) do
            for all fb in solution[ied] do
                if not check_if_ied_supports_fb(system, ied, fb) then
                    reduce_solution_space(solution_space, ied, fb)
                    break and continue with next solution
                end if
                                          ▷ check_fb_level_constraints(fb, solution)
            end for

            if not check_device_level_constraints(ied, solution) then
                reduce_solution_space(solution_space, solution)
                break and continue with next solution
            end if
        end for

        for all connection in set(get_connections(system)) do
            mapping := get_connection_map(solution, connection)

            if not check_if_ieds_can_communicate(scl, solution, mapping)
then
                reduce_solution_space(solution_space, mapping)
                break and continue with next solution
            end if
                 ▷ check_connection_level_constraints(connection, solution)
        end for

              ▷ check_application_level_constraints(bay, ieds, fbs, solution)

        calculate_factor(solution)
    end for
                              ▷ found solutions for bay if solution_space not empty
    solutions[bay] := solution_space
end for
```