Algorithm 1: Assign Disassembly Line (Algo-ADL)

```
/* The algorithm must be repeated for each p invidual.    */
Data: nM, nA, nJ, nS, nORSuc, SUC, PRE, sol_p, sc, θ, t, C ;
      /* Algo-ADL inputs */
Result: X, U, tf ;                          /* Algo-ADL output */
begin
    X_{m(i∈sc_m)js} ← [0] ;  /* (0,1) assign of tasks to stations */
    U_{js} ← [0] ;                        /* (0,1) opening stations */
    tf_{m(i∈sc_m)} ← [0] ;   /* ≥ t_{m(i∈sc_m)} finishing time of tasks */
    task_pre, task_theta ← Algo-SPT ;      /* Set selected tasks'
     precedence and theta lists */
    /* Start of Algo-ADL || Repeat for each m model.      */
    for m ← 1 to nM do
        j ← 0 ;       /* Initial value of station number is 0 */
        assignable ← ∅;
        for i ∈ sc_m if task_pre_{mi} = ∅ then Append i in assignable;
        assigned ← ∅;
        while |selected_m| ≠ |assigned| do
            if |assignable| = 1 then
                r ← 0;
                j, X, U, tf, assignable, assigned ← Algo-AR
            else if |assignable| ≥ 2 then
                r ← 0 for l ← 1 to |assignable| do
                    if index (assignable_l, sol_{pm(nORSuc_m+2)}) <
                     index (assignable_r, sol_{pm(nORSuc_m+2)}) then
                        r ← l
                    end
                end
                j, X, U, tf, assignable, assigned ← Algo-AR
            for i ∈ sc_m do
                if i ∉ assigned ∩ i ∉ assignable then
                    for i' ∈ task_pre_{mi} do
                        if i' ∉ assigned then
                            Break
                        else if i' is last element in task_pre_{mi} then
                            Append i in assignable
                    end
                end
            end
        end
    end
    return X, U, tf
end
```