The entrypoint of algorithm is **dfs-solution**. Utilize it to find optimal solutions, and find guaranteed feasible solutions if scaled too much.

Algorithm 1: dfs-solution

Input: Jobs $J = [j_1, j_2, \cdots, j_n]$, Available Partitions P, Execution Plan E includes a series of jobs.

Output: Feasible solutions S

```
1  Found Solution
2  t_q = +∞ ;            /* Total quickest finished jobs series */
3  if J = ∅ then
4  |  t ← CalculateTime(E)
5  |  if t < t_q then
6  |  |  S = ∅
7  |  |  t_q ← t
8  |  end
9  |  if t = t_q then
10 |  |  S ←+ E
11 |  end
12 end
13 Searching
14 for j_i in J do
15 |  if dependences[j_i] ∈ P then
16 |  |  L = Heuristic(j_i, P, E) ; /* Ordered slots for job to be
   |  |   executed */
17 |  |  for l_k in L do
18 |  |  |  dfs-solution(P + j_i, J − j_i, E + e_{j_i,l_k})
19 |  |  end
20 |  end
21 end
```

Algorithm 2: Heuristic

Input: All available slots $L = [l_1, l_2, \cdots, l_m]$ in each datacenter(one for each), Job j, Available Partitions P, Execution Plan E includes a series of jobs.

Output: Ordered slots $L^\star$

for l_k *in* L **do**
  $m \leftarrow$ [data center contains same main job, data center contains main job partitions]
  $w_k = \begin{cases} 3 & \text{if } m = [true, true] \\ 2 & \text{if } m = [true, false] \\ 1 & \text{if } m = [false, false] \\ 0 & \text{else} \end{cases}$
end
$L^\star \leftarrow SORT(L, W)$; `/* Sort slots by weight` $W = [w_1, \cdots, w_m]$ `*/`
return $L^\star$

Algorithm 3: CalculateTime

Input: Jobs $J = [j_1, j_2, \cdots, j_n]$, Available Partitions P, Execution Plan $E \in J$ includes a series of jobs.

Output: Time t

$R_f \leftarrow \varnothing$
for j_i *in* J **do**
  $td_i \leftarrow 0$
  for $d_{j_i,k}$ *in* $dependences[j_i]$ **do**
    if $DC[d_{j_i,k}] \neq DC[j_i]$ **then**
      $LINKS \leftarrow ShortestPath(DC[d_{j_i,k}], DC[j_i])$
      $td_i \leftarrow \sum_{l_h \in LINKS} Time[l_h]$
    end
  end
  $avail[j_i] \xleftarrow{+} td_i + duration[j_i]$
  $R_f \xleftarrow{+} avail[j_i]$
end
$t \leftarrow \max_{r_{f,i} \in R_f} r_{f,i}$
return $L^\star$

Optimal Solution. If we want to find the optimal solution, and let the **dfs** algorithm finish all possibilities, the complexity is $O(n!m!)$, which n

is number of jobs, m is total slot number of all datacenters.

Approximation Solution. If we just want to find the approximation solution, which guaranteed at least feasible, the complexity is $O(nm + m \log m)$, which n is number of jobs, m is total slot number of all datacenters.