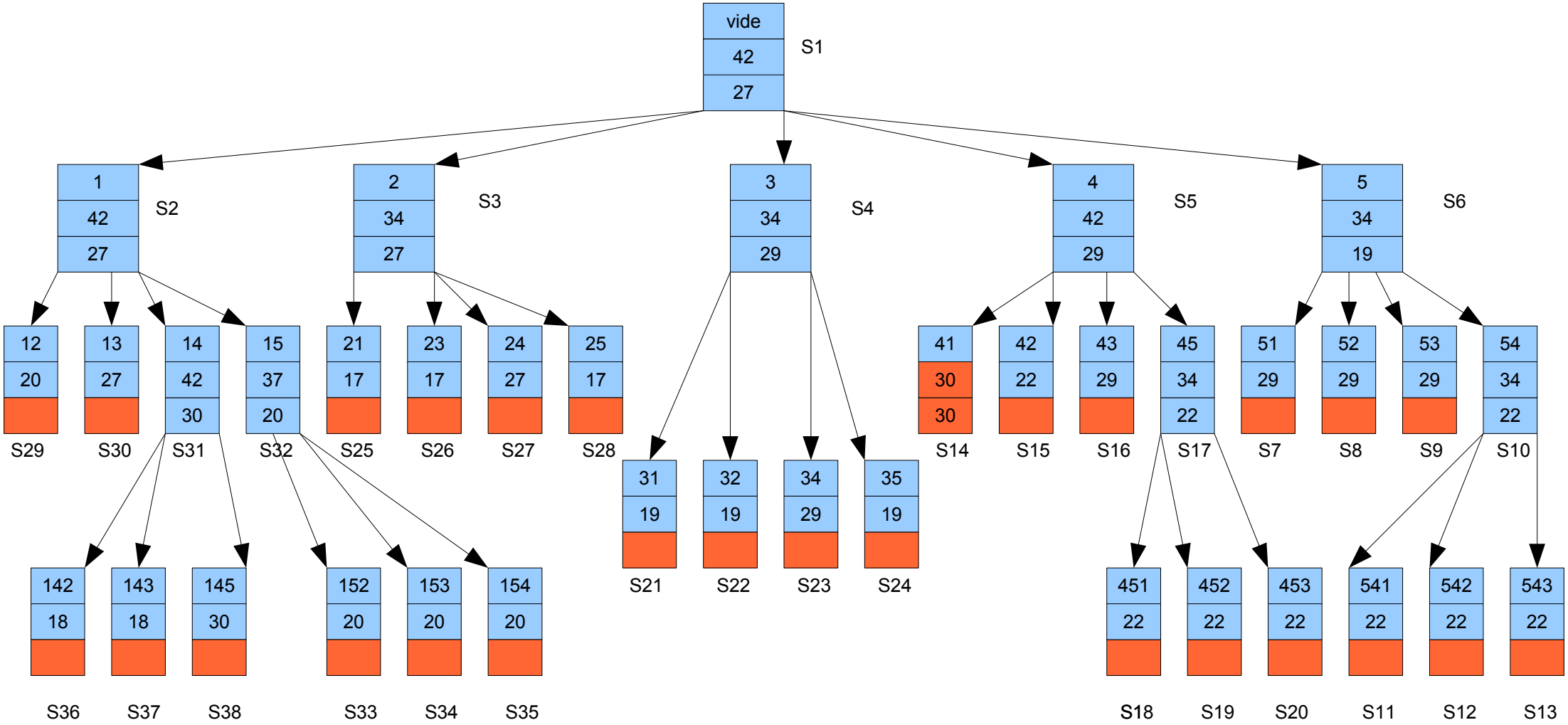
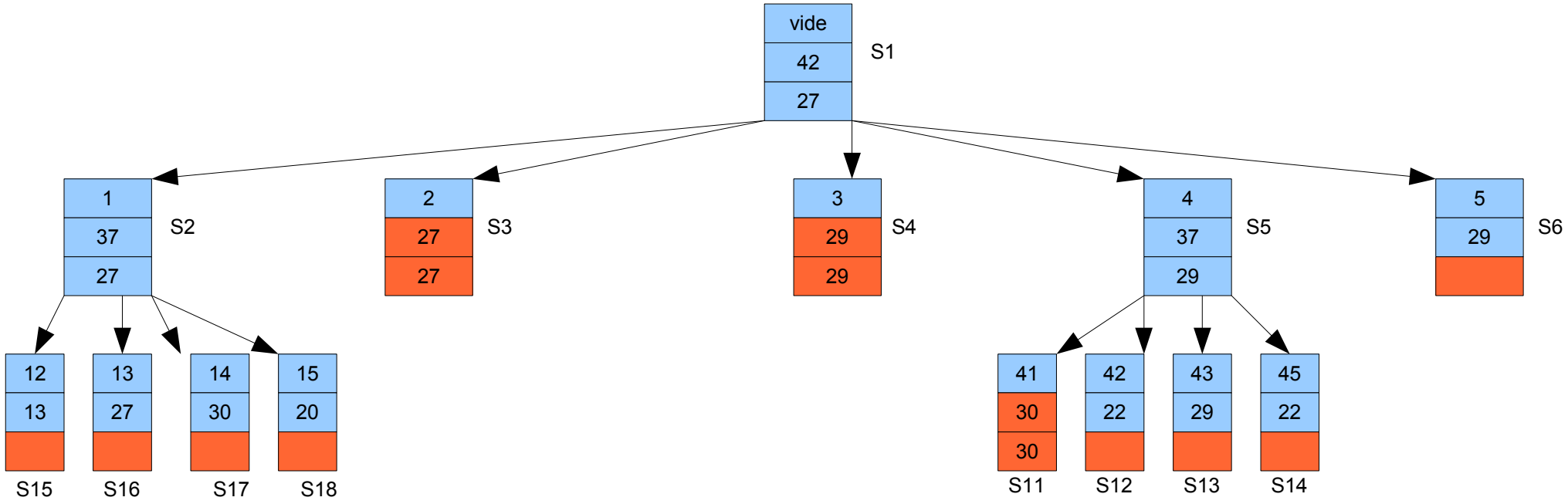
Application de la méthode arborescente n°2: beaucoup de noeuds développés



Les noeuds affichent verticalement Js, h(s), g(s). g(s) n'apparait pas si le noeud peut-être tronqué avant le calcu l.
Dès la première séparation, la meilleure solution rencontrée vaut 29. Lorsqu'on développe le noeud S14, une solution à 30 est trouvée.

Application de la méthode arborescente n°2 avec h modifié:

En chaque noeud on calcule un ensemble R(S) de tâches qui sont nécessairement en retard dans tous les ordonnancements du noeud:
i est dans R(S) si durée de Js+durée de i >échéance de i.
on définit h(S)= profits des tâches à l'heure de Js+ profits des tâches ni dans Js ni dans R(S)



Application de la méthode arborescente vue en cours:

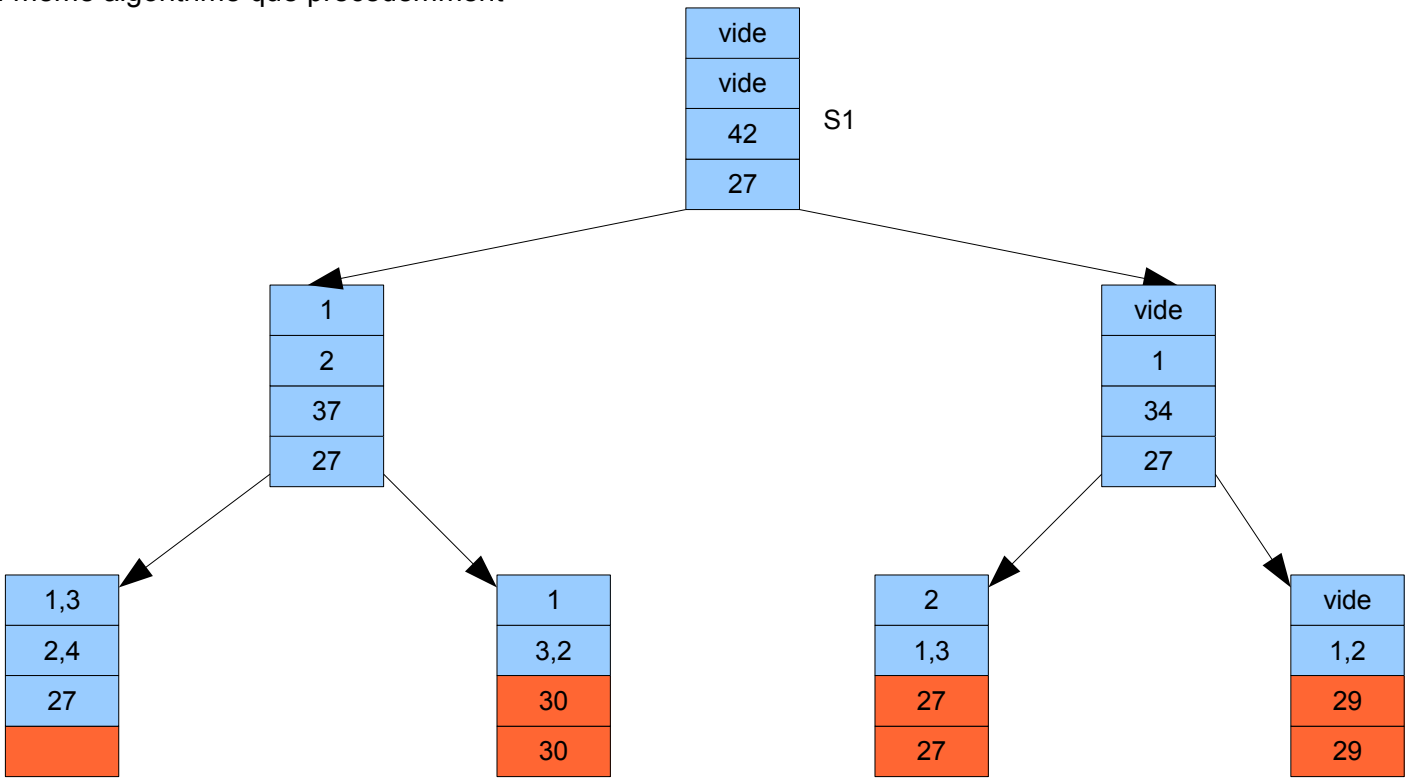
Un noeud S est associé à un ensemble de tâches à l'heure H(S) et un ensemble de tâches en retard R(S).
Il correspond à l'enemble des ordonnancements où les tâches de H(S) sont à l'heure et où les tâches de R(S) sont en retard.

Condition de faisabilité: si les tâches de H(S) mises en ordre croissant des échéances sont toutes à l'heure, alors il y a au moins un ordonnancement faisable associé au noeud S. On dit que H(S) définit une séquence à l'heure.

Dominance: si i n'est pas dans H(S) ni dans R(S), et que H(S) et i ne forment pas une séquence à l'heure, alors dans tous les ordonnancements de S, i sera en retard. On peut donc l'ajouter à R(S)

Séparation on choisit une tâche ni dans H(S) ni dans R(S) et l'on crée deux fils S' et S'', avec H(S')=H(S)u{i}, R(S')=R(S); H(S'')=H(S); R(S'')=R(S)u{i}

Evaluation par excès: somme des profits des tâcheq qui ne sont pas dans R(S).
Evaluation par défaut: même algorithme que précédemment



Les noeuds présentent verticalment, H, R, h, g