MDP ONLINE

Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming

Bonet, B, Geffner, H. (2003)

Chafik EL MAHDAOUI

16 novembre 2017

UQÀM | Faculté des sciences
Université du Québec à Montréal

Introduction

- Résoudre des MDPs
- Algorithme long : itération par valeur, itération par politique
- RTDP :
  - ne parcourt pas tout l'ensemble d'états
  - peut retourner une solution rapidement
- L-RTDP :

  améliore la convergence de RTDP

Sommaire

1. Rappels
   a) MDP
   b) Itération par valeur
   c) Exploration en ligne / hors ligne
2. RTDP
   a) Principe
   b) Exemple
   c) Limite
3. L-RTDP
   a) Principe
   b) Exemple

UQÀM | Faculté des sciences
Université du Québec à Montréal

Processus Décisionnel de Markov

- Systèmes stochastiques :
  - Une action = plusieurs états possibles
- Problème d'optimisation
- Formulation par
  - Coûts : minimiser le cout
  - Récompenses : maximiser la récompense
- Une décision : choix d'une action pour un état
- Un plan : ensemble de decision

UQÀM | Faculté des sciences
Université du Québec à Montréal

Processus Décisionnel de Markov

Équation de Bellman (par coût)

$$V^*(s) = \min_{a \in A} \left(C(a) + \sum_{s' \in S} P(s', s, a) \times V(s') \right)$$

But : résoudre cette équation pour chaque états

Itération par valeur

- Itère sur tous les sommets
- Mets à jour la valeur des sommets grâce à l'équation de Bellman
- Jusqu'à ce que la fonction V converge

Exploration en ligne / hors ligne

- Hors ligne
  - Calculs tout les états avant de prendre une décision
  - Exemples : VI, PI …
- En ligne
  - Prend une décision en fonction de l'état actuel et d'une fonction heuristique
  - Exemples : RTDP, L-RTDP

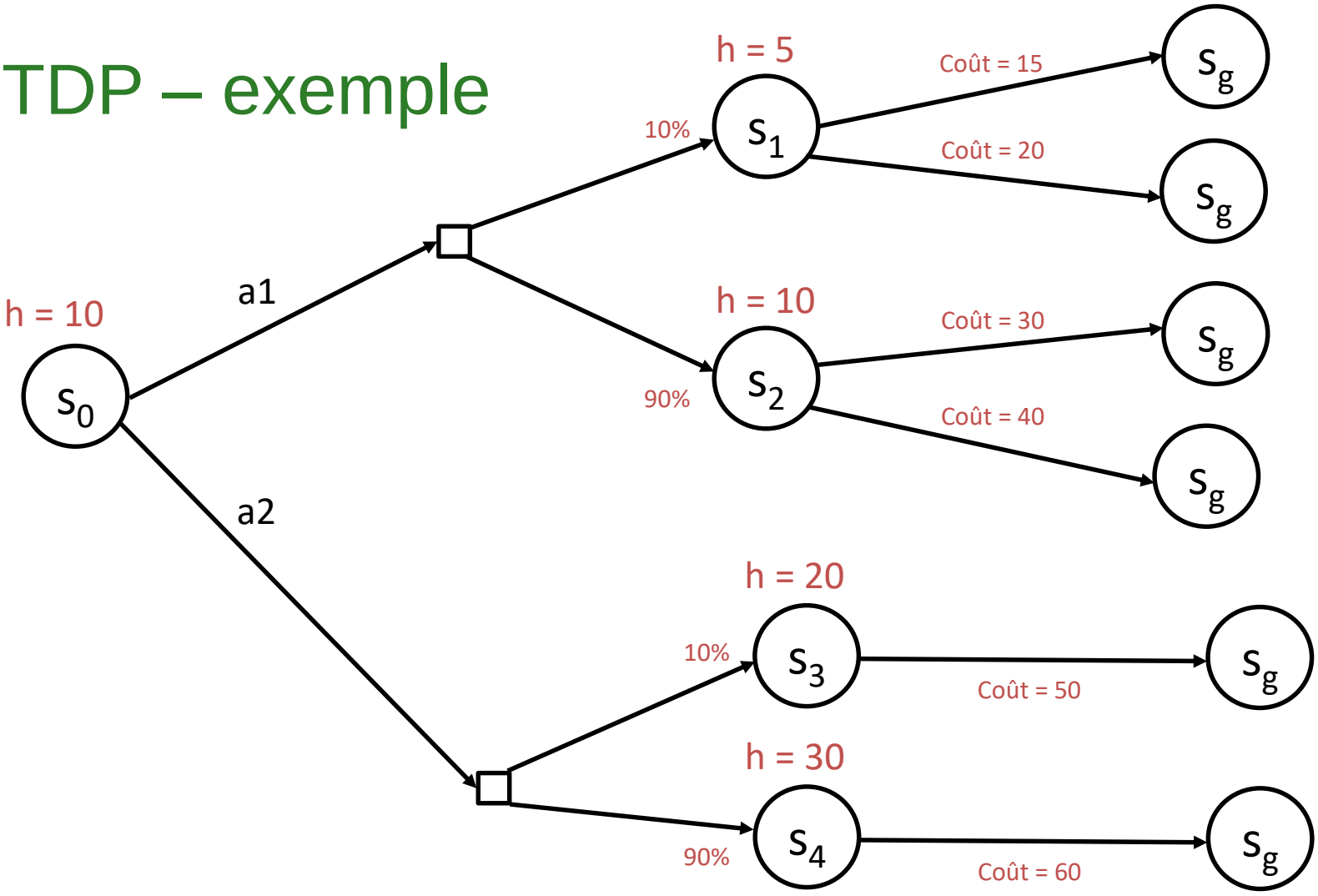
RTDP (Barto et al., 1995)

- Motivations :
  - MDP de grande taille
  - Obtenir un résultat « à tout moment »
- Algorithme glouton
- Réalise des essais
- Favorise les trajectoires les plus probables

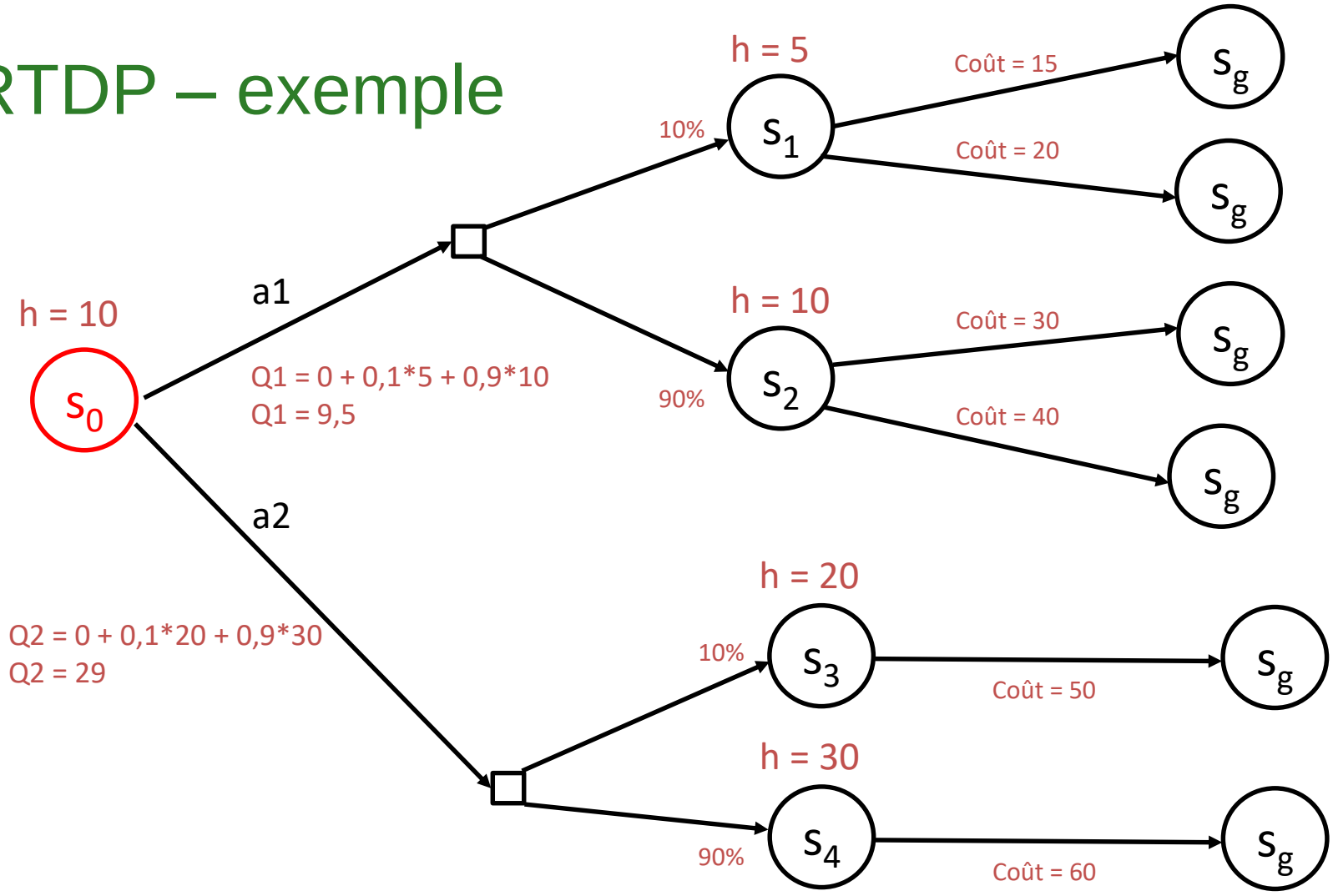
RTDP - algorithme

```
RTDP(s : state)
begin
    repeat RTDPTRIAL(s)
end

RTDPTRIAL(s : state)
begin
    while ¬s.GOAL() do
        // pick best action and update hash
        a = s.GREEDYACTION()
        s.VALUE = c(s, a) + Σ_{s'} P_a(s'|s) · s'.VALUE
        // stochastically simulate next state
        s = s.PICKNEXTSTATE(a)
end
```

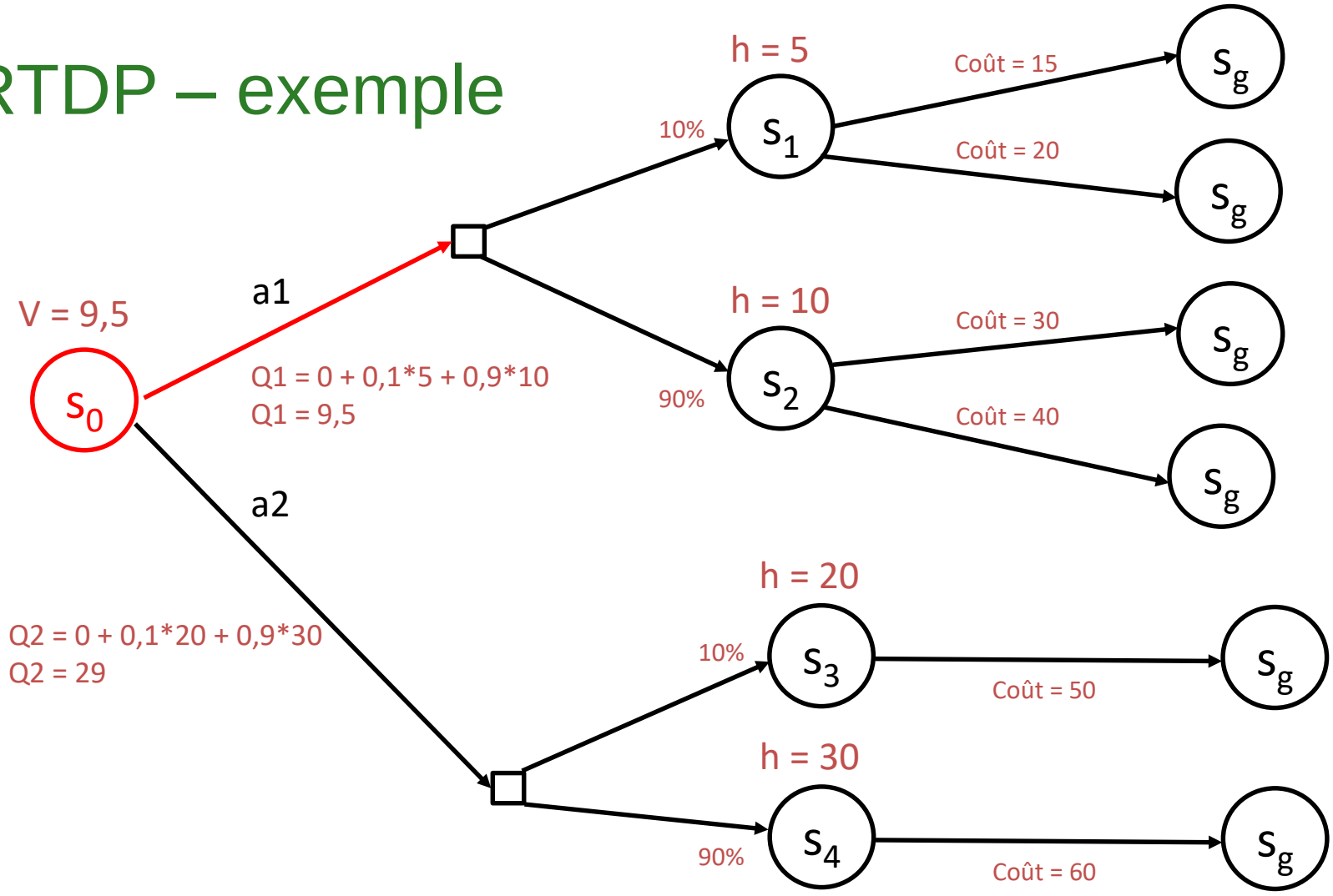
RTDP – exemple



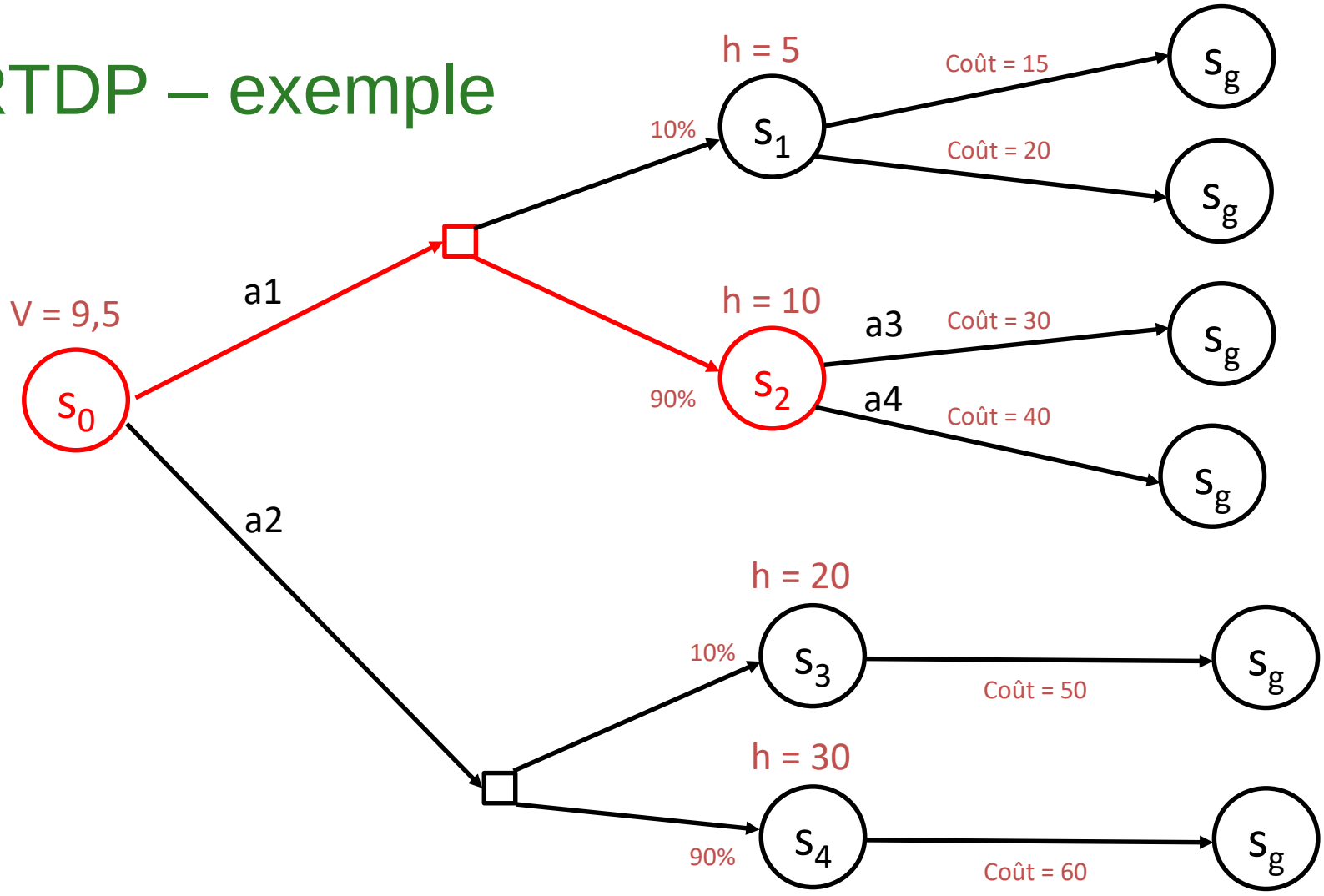
RTDP – exemple



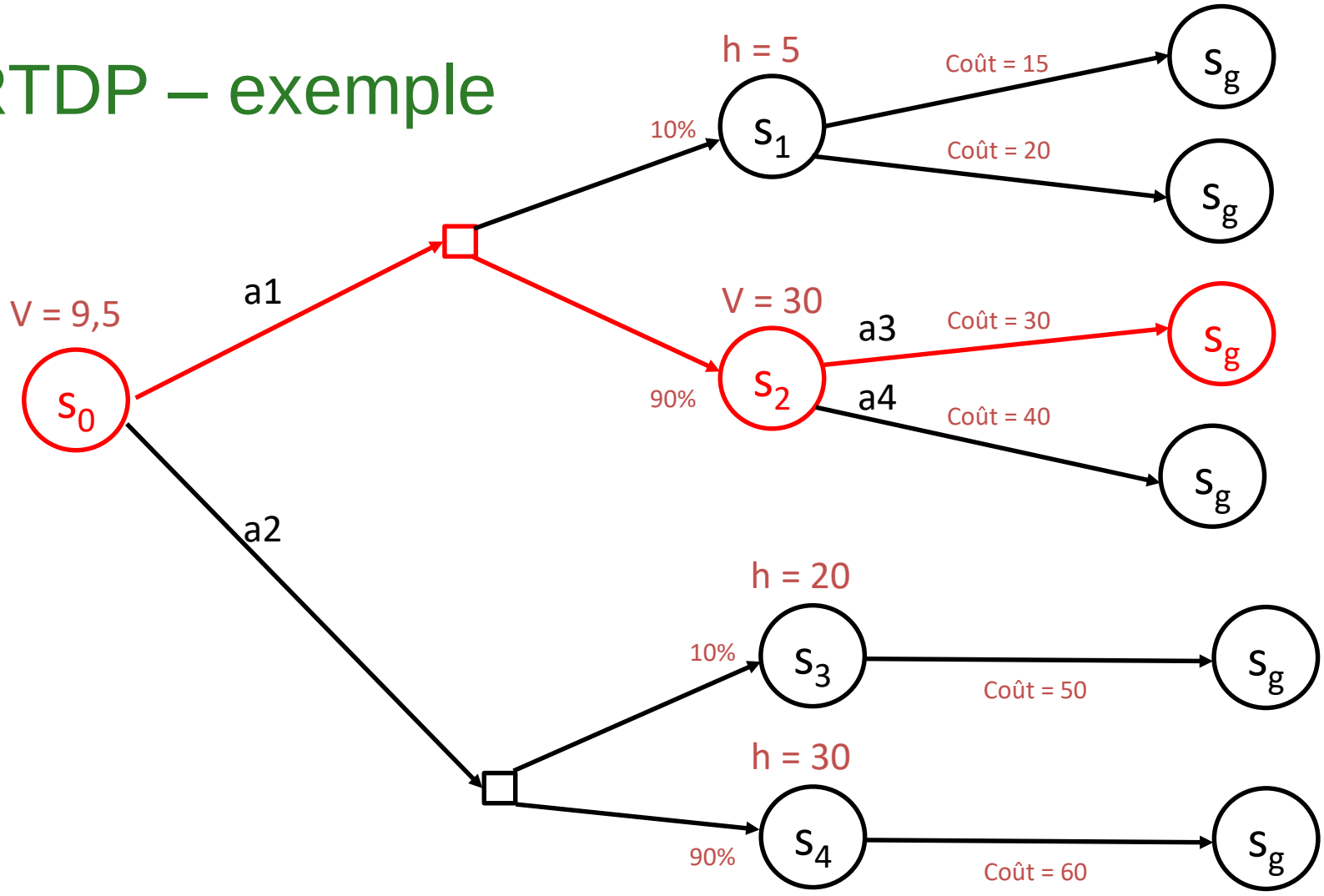
RTDP – exemple



RTDP – exemple



RTDP – exemple



Comparaison de RTDP et VI

- Résultats en fonction du temps
- Convergence vers un optimal
  - Quelques secondes pour VI
  - Plus de 10 minutes pour RTDP

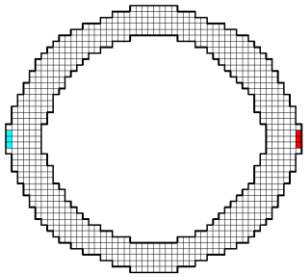
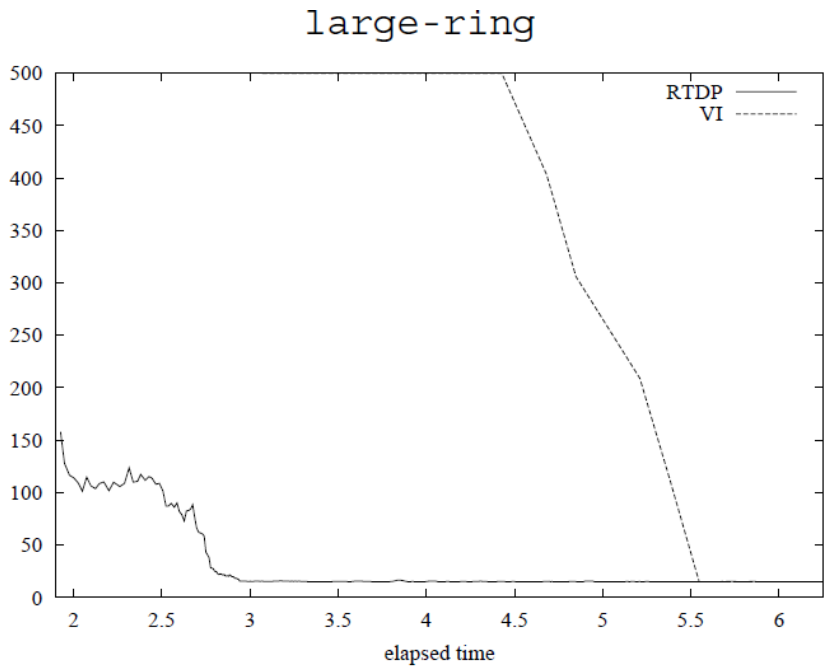
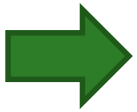


Figure 2: Racetrack for `large-ring`. The initial and goal positions marked on the left and right



L-RTDP (Bonet and Geffner. 2003)

- RTDP + label « résolu » sur les états
- Évite les états déjà visités
- Visites plus d'états différents
- Fini lorsque l'état initial est « résolu »



Améliorer la convergence de RTDP

L-RTDP - algorithme

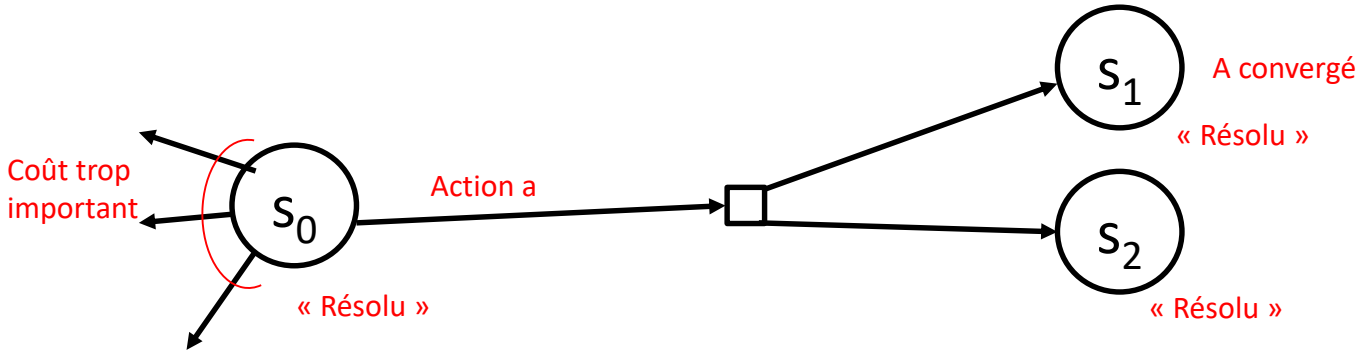
```
LRTDP(s : state, ε : float)
begin
    while ¬s.SOLVED do LRTDPTRIAL(s, ε)
end

LRTDPTRIAL(s : state, ε : float)
begin
    visited = EMPTYSTACK
    while ¬s.SOLVED do
        // insert into visited
        visited.PUSH(s)
        // check termination at goal states
        if s.GOAL() then break
        // pick best action and update hash
        a = s.GREEDYACTION()
        s.UPDATE()
        // stochastically simulate next state
        s = s.PICKNEXTSTATE(a)
    // try labeling visited states in reverse order
    while visited ≠ EMPTYSTACK do
        s = visited.POP()
        if ¬CHECKSOLVED(s, ε) then break
end
```

Algorithm 4: LRTDP.

L-RTDP – algorithme (suite)

CHECKSOLVED :

Vérifie que tous les états accessibles à partir de l'état s en appliquant la meilleure action a sont résolus ou ont convergé



Si oui, l'algorithme marque résolu l'état courant et les états accessibles par l'action a
Si non, il les mets à jour (applique l'équation de Bellman)

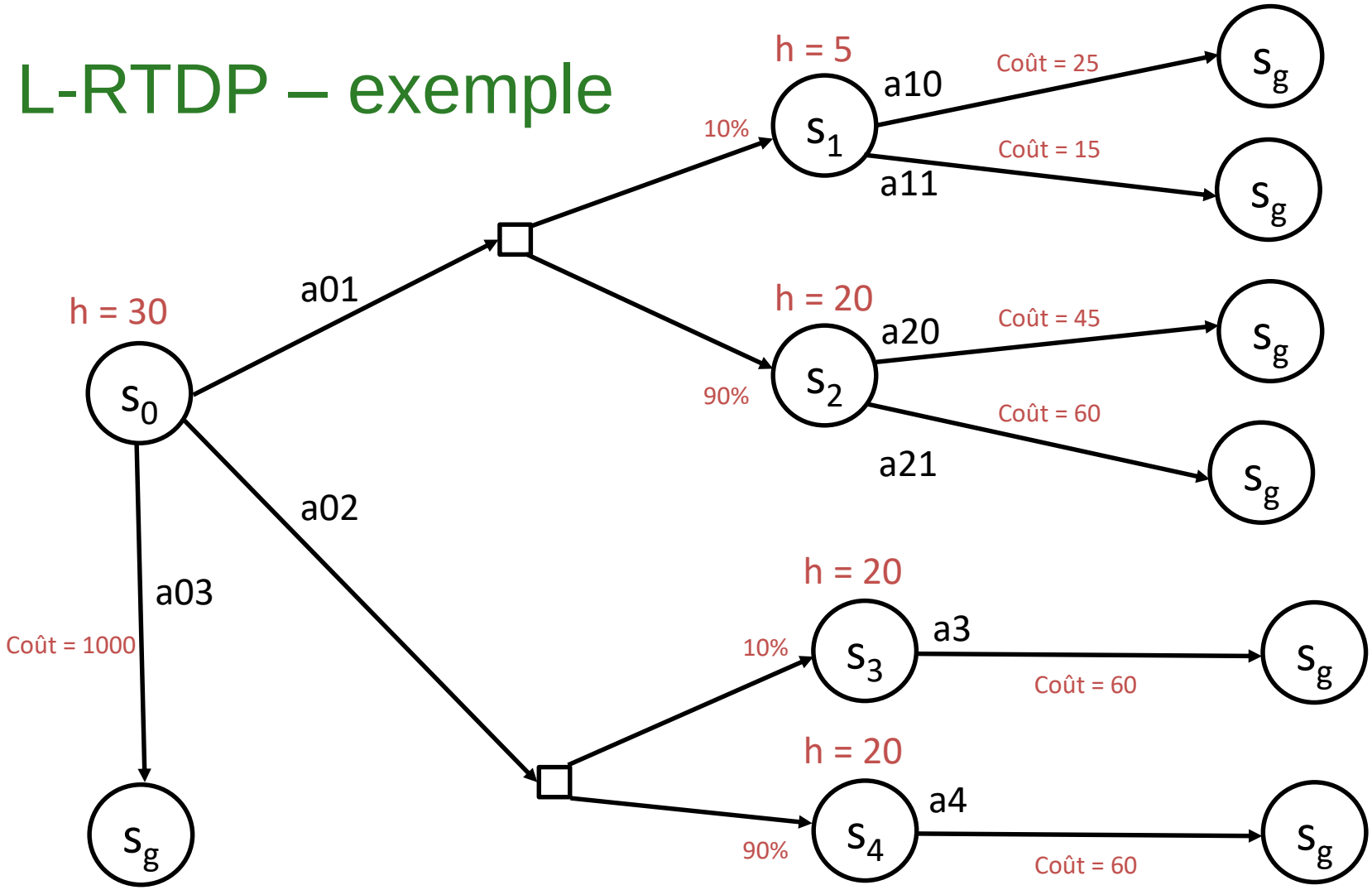
L-RTDP – algorithme (suite)

```
CHECKSOLVED(s : state, ε : float)
begin
    rv = true
    open = EMPTYSTACK
    closed = EMPTYSTACK
    if ¬s.SOLVED then open.PUSH(s)
    while open ≠ EMPTYSTACK do
        s = open.POP()
        closed.PUSH(s)
        // check residual
        if s.RESIDUAL() > ε then
            rv = false
            continue
        // expand state
        a = s.GREEDYACTION()
        foreach s' such that P_a(s', s) > 0 do
            if ¬s'.SOLVED & ¬IN(s', open ∪ closed)
                open.PUSH(s')
[• • •]
```

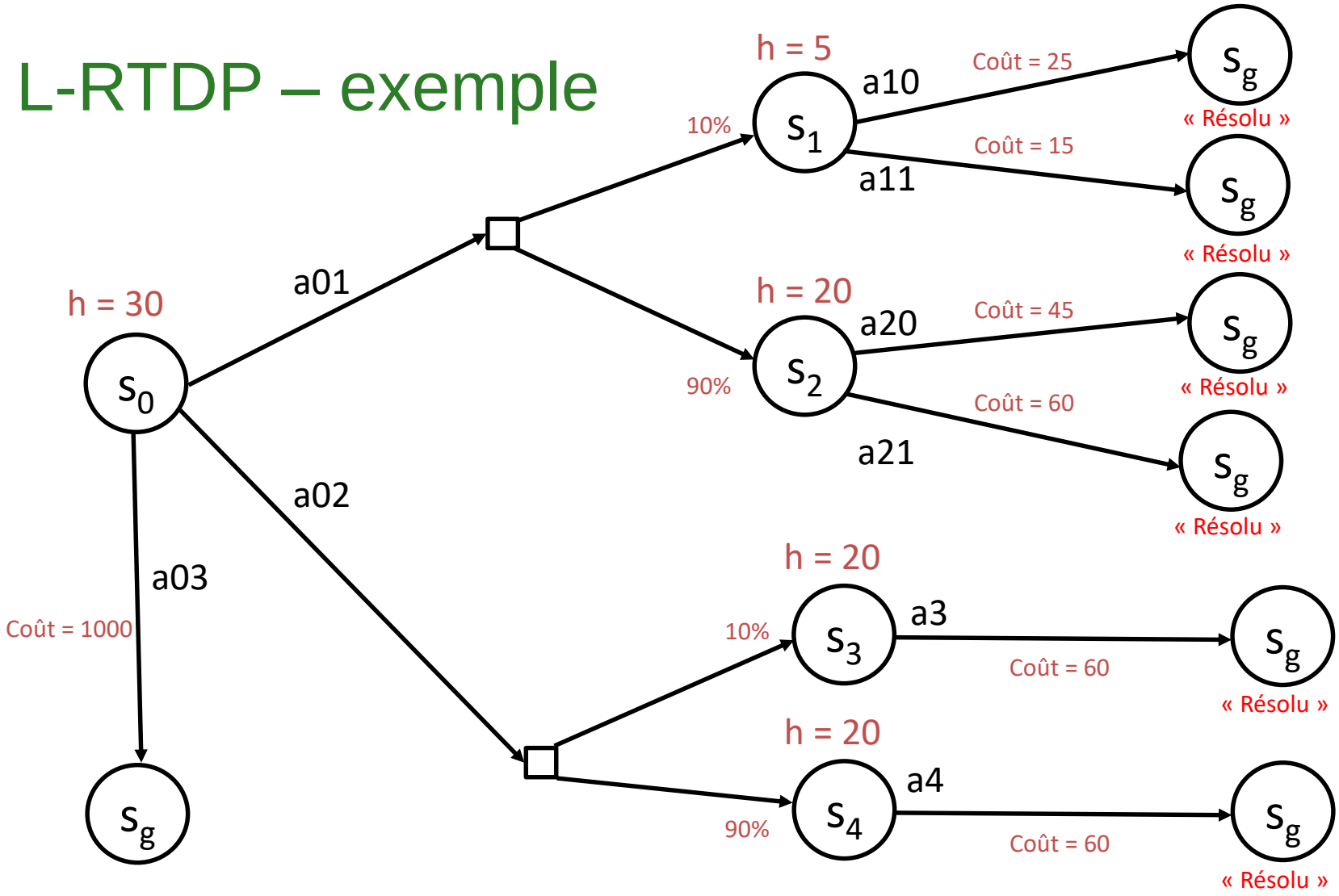
Vérifie que tous les états accessibles à partir de l'état s en appliquant la meilleure action a sont résolus ou ont convergé

```
[• • •]
    if rv = true then
        // label relevant states
        foreach s' ∈ closed do
            s'.SOLVED = true
    else
        // update states with residuals and ancestors
        while closed ≠ EMPTYSTACK do
            s = closed.POP()
            s.UPDATE()
    return rv
end
```

L-RTDP – exemple

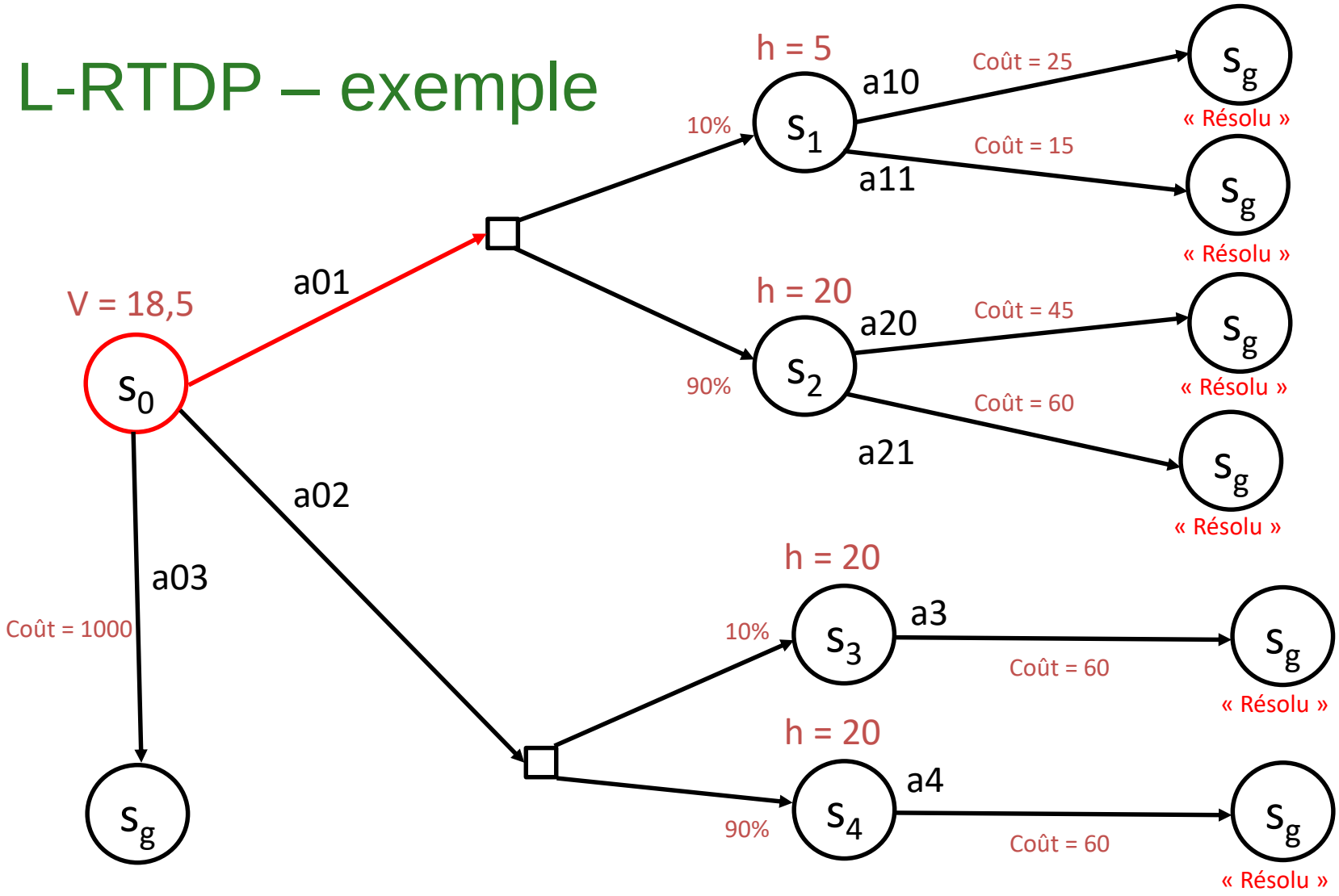


L-RTDP – exemple

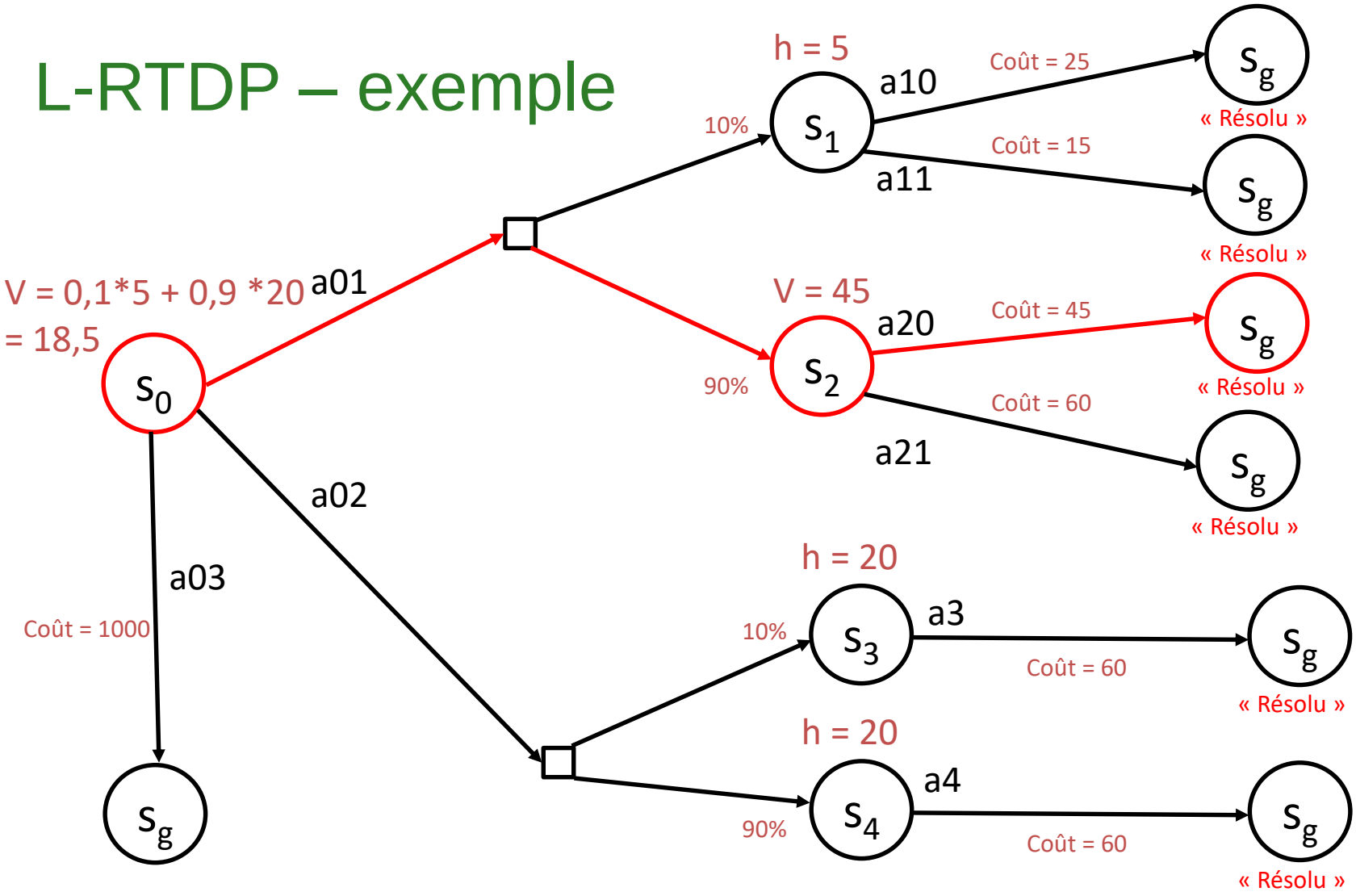


UQÀM | Faculté des sciences
Université du Québec à Montréal

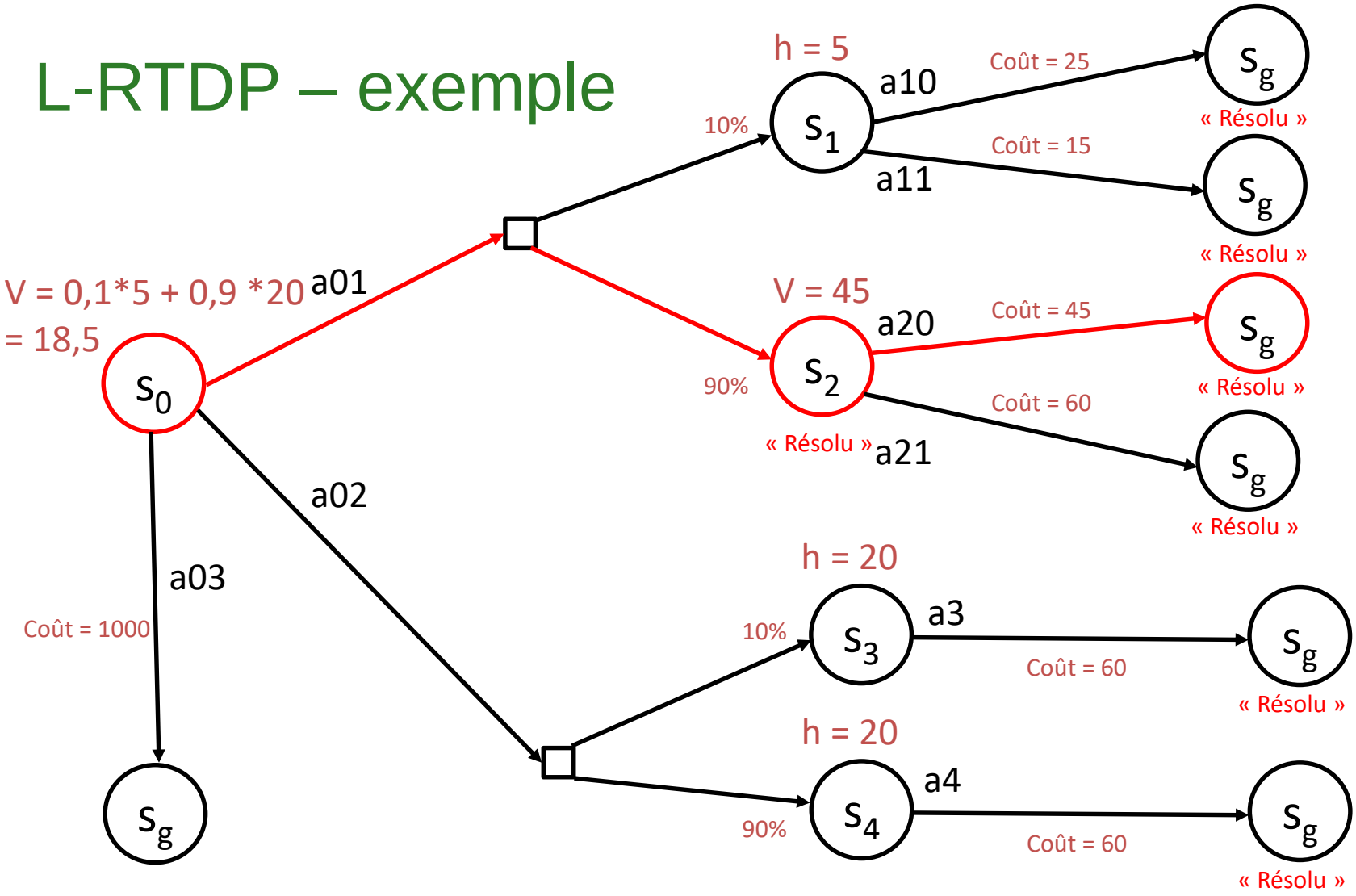
L-RTDP – exemple



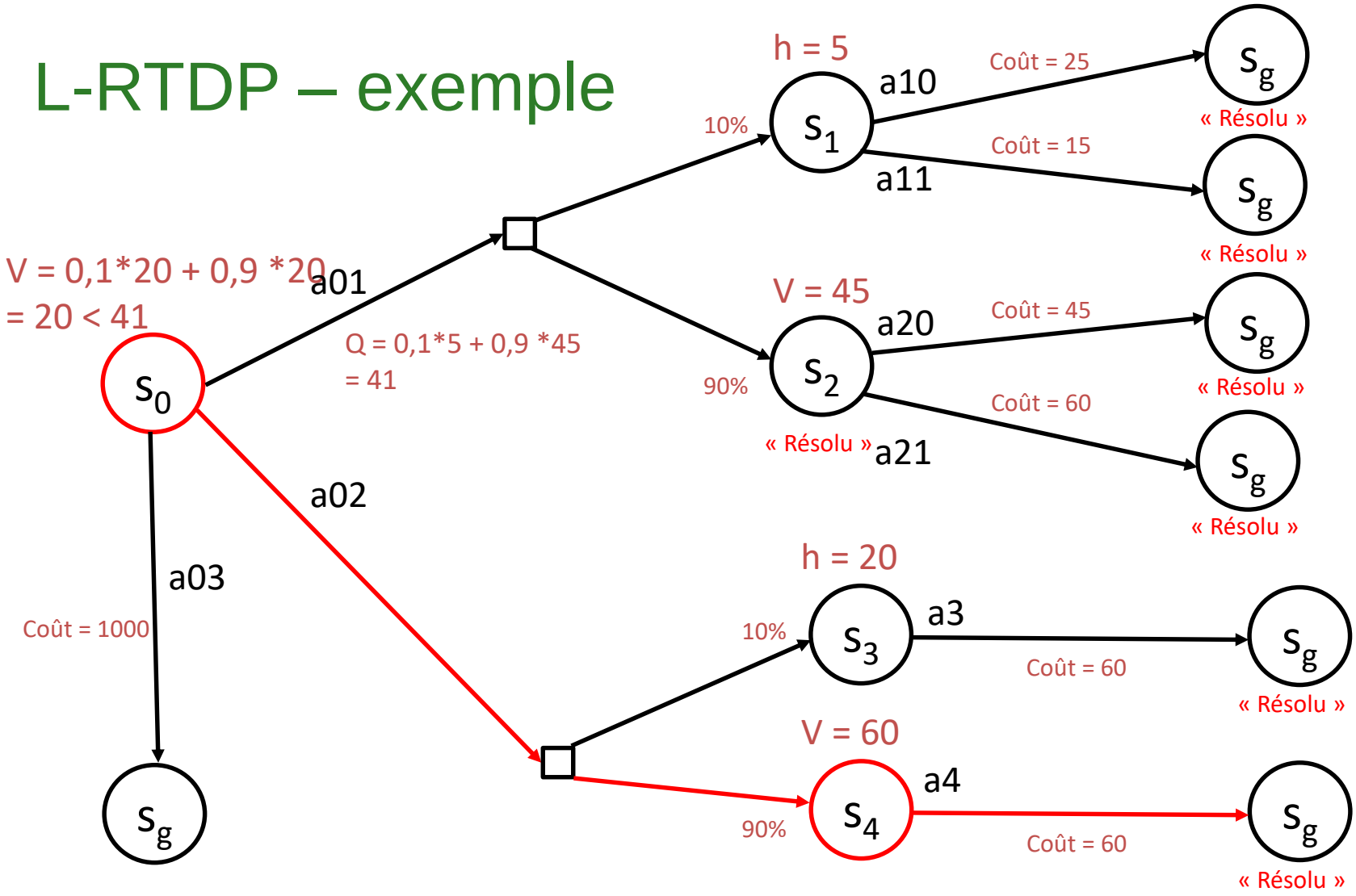
L-RTDP – exemple



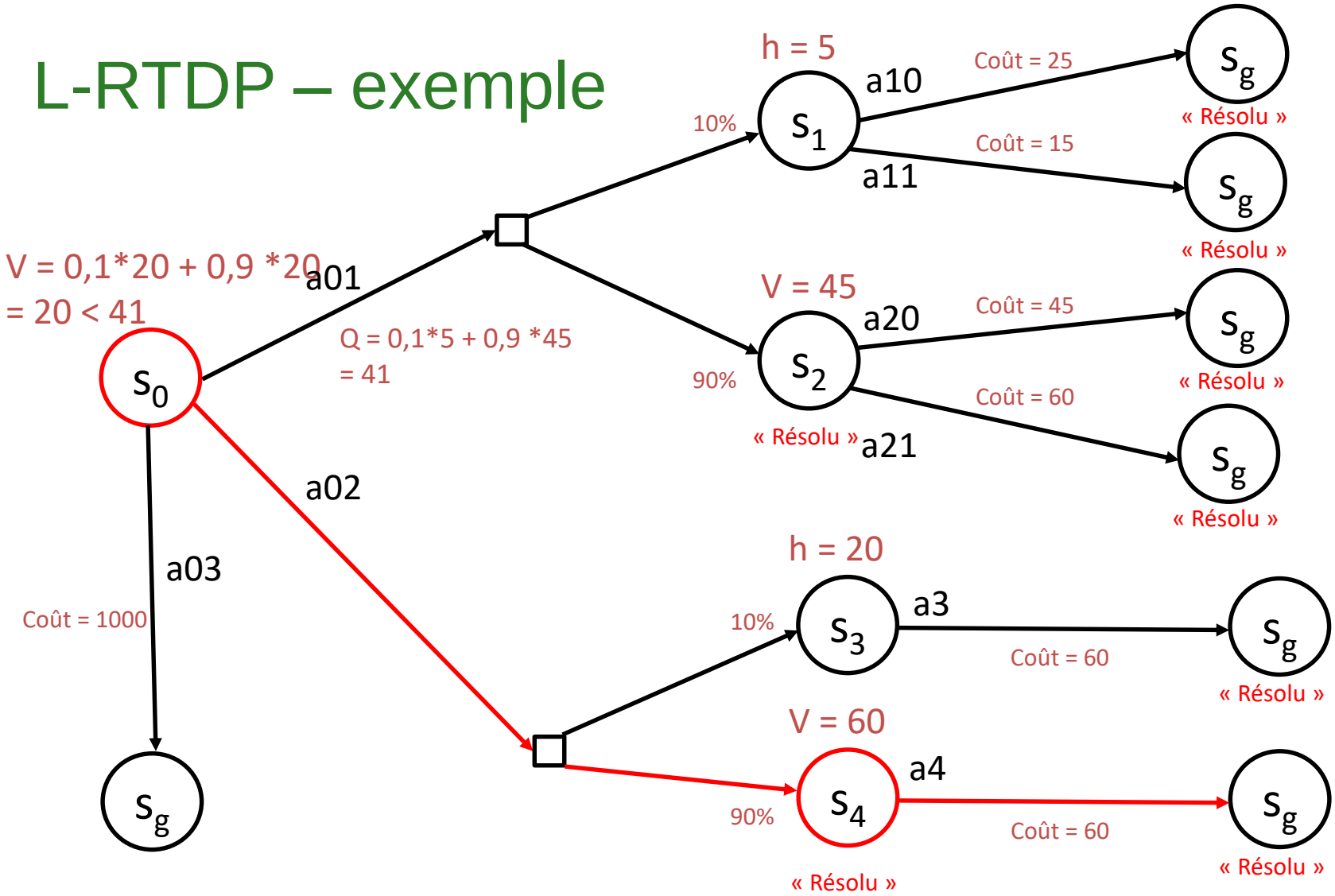
L-RTDP – exemple



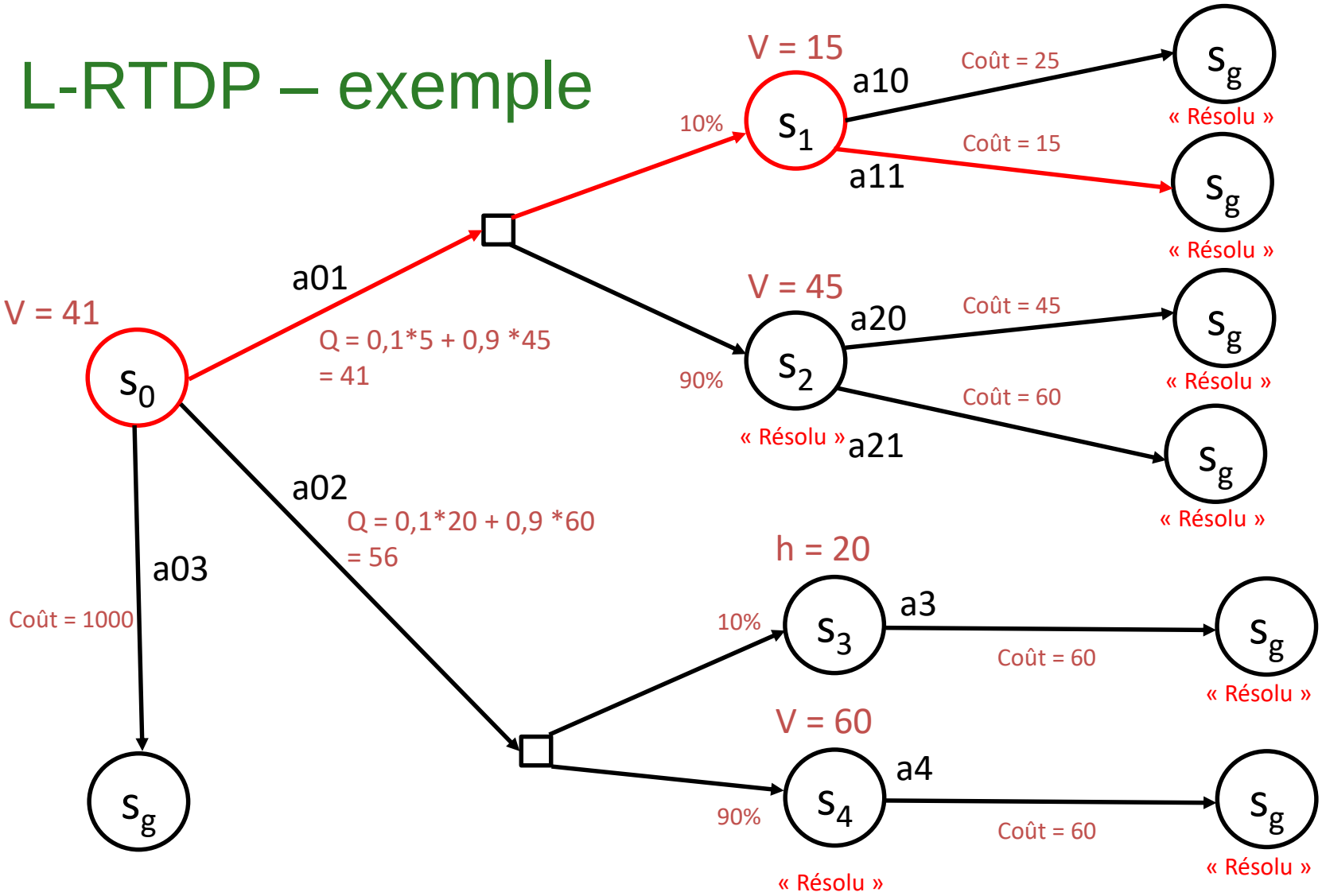
L-RTDP – exemple



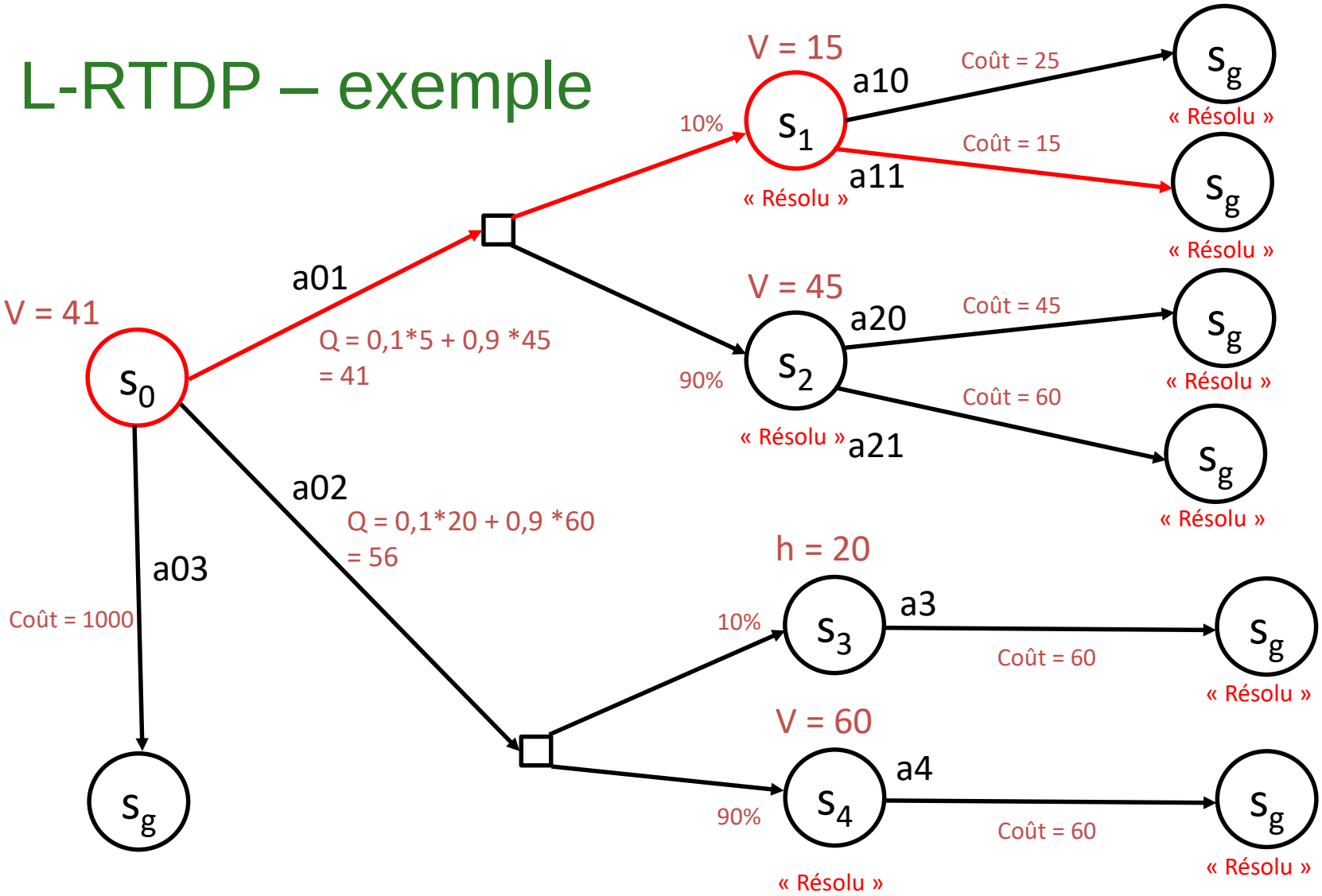
L-RTDP – exemple



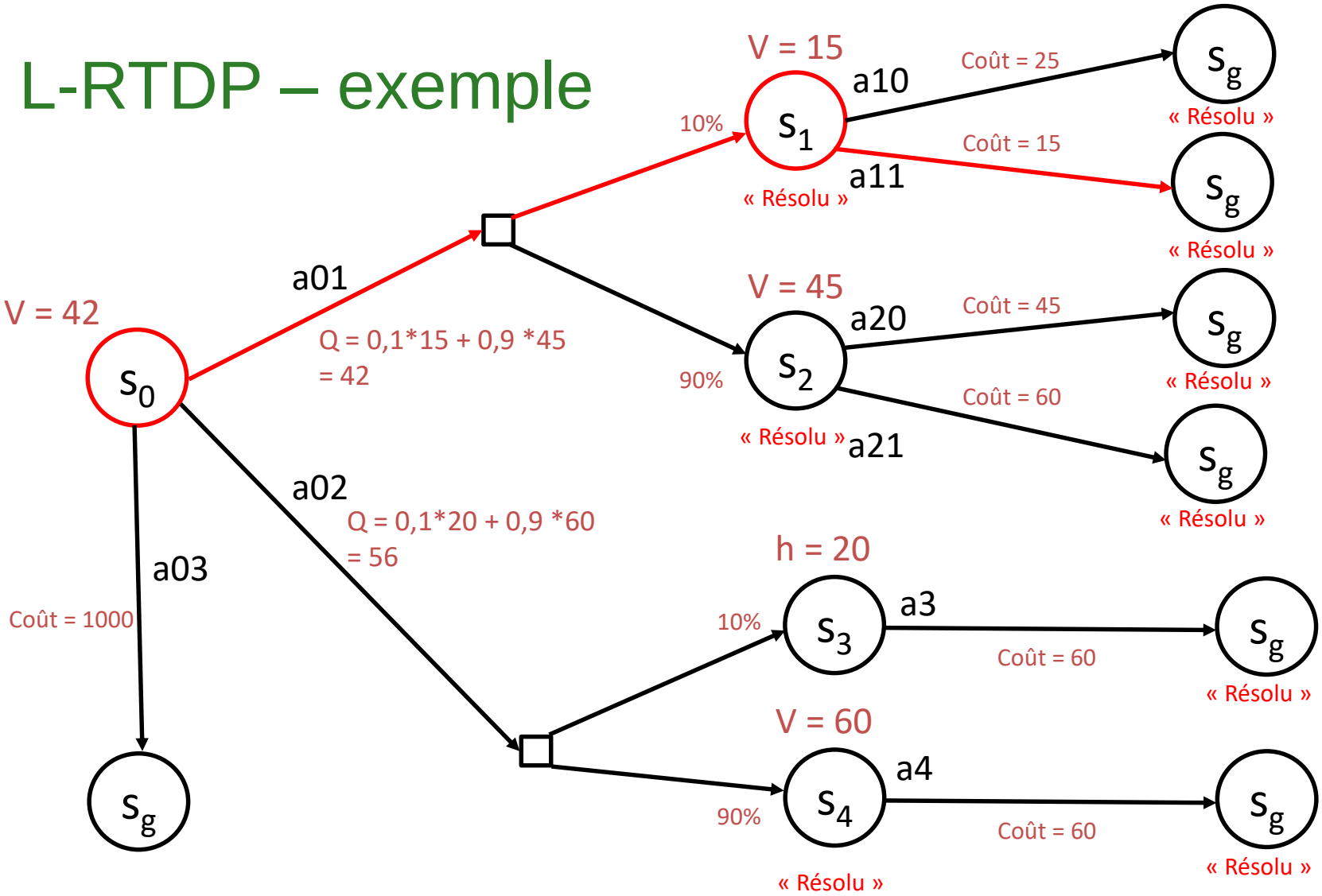
L-RTDP – exemple



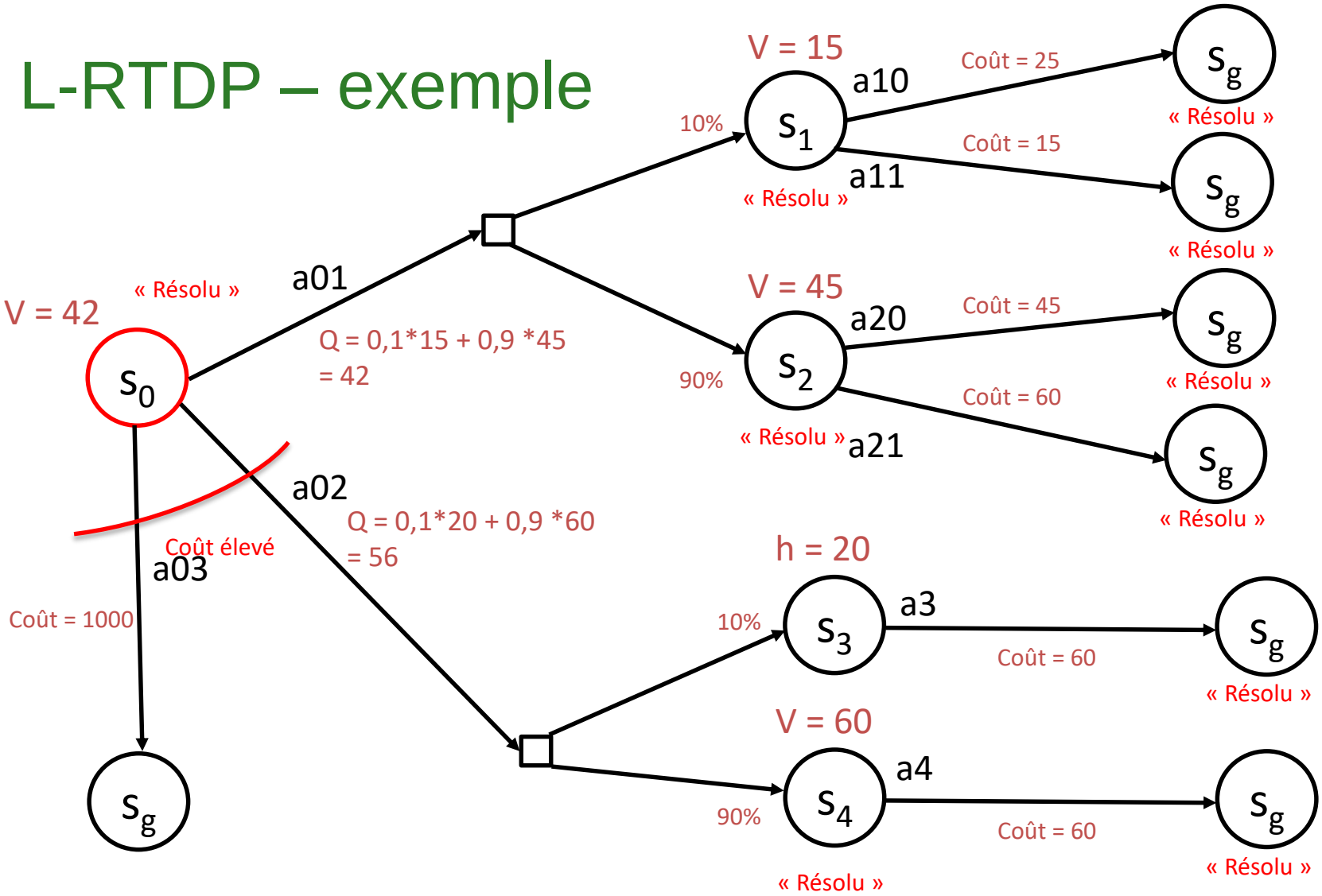
L-RTDP – exemple



L-RTDP – exemple



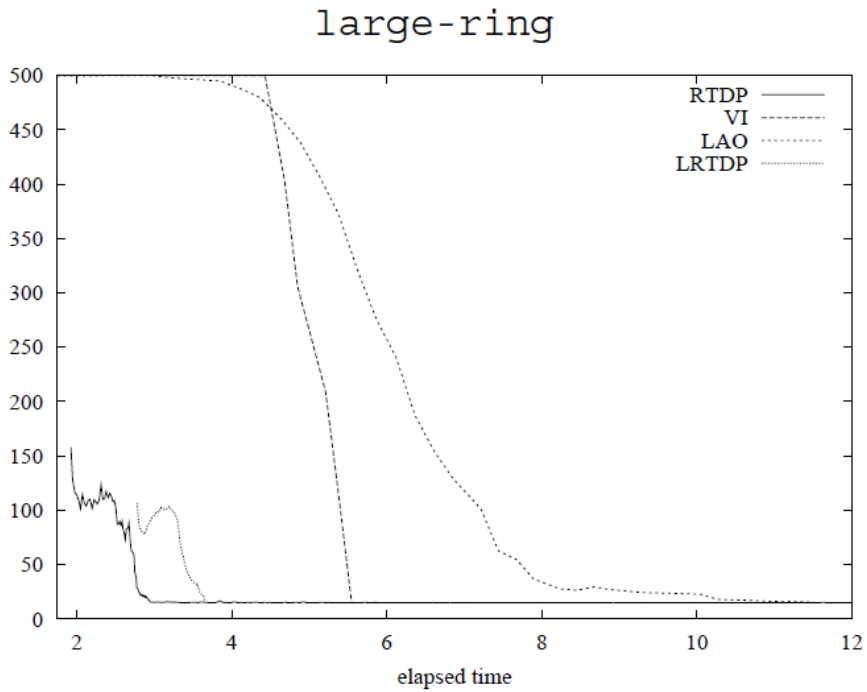
L-RTDP – exemple



UQÀM | Faculté des sciences
Université du Québec à Montréal

Résultats de L-RTDP

- Coût moyen en fonction du temps



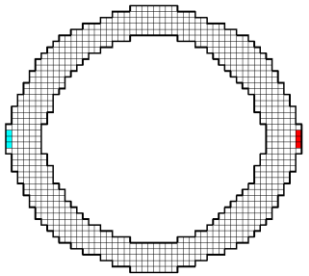


Figure 2: Racetrack for `large-ring`. The initial and goal positions marked on the left and right

Résultats de L-RTDP

- Convergence vers un optimal améliorée

algorithm	small-b	large-b	h-track	small-r	large-r	small-s	large-s	small-y	large-y
VI$(h=0)$	1.101	**4.045**	**15.451**	0.662	5.435	5.896	78.720	16.418	61.773
ILAO*$(h=0)$	2.568	11.794	43.591	1.114	11.166	12.212	250.739	57.488	182.649
LRTDP$(h=0)$	**0.885**	7.116	15.591	**0.431**	**4.275**	**3.238**	**49.312**	**9.393**	**34.100**

Table 2: Convergence time in seconds for the different algorithms with initial value function $h = 0$ and $\epsilon = 10^{-3}$. Times for RTDP not shown as they exceed the cutoff time for convergence (10 minutes). Faster times are shown in bold font.

algorithm	small-b	large-b	h-track	small-r	large-r	small-s	large-s	small-y	large-y
VI(h_{min})	1.317	4.093	12.693	0.737	5.932	6.855	102.946	17.636	66.253
ILAO*(h_{min})	1.161	2.910	11.401	0.309	3.514	0.387	1.055	0.692	1.367
LRTDP(h_{min})	**0.521**	**2.660**	**7.944**	**0.187**	**1.599**	**0.259**	**0.653**	**0.336**	**0.749**

Table 3: Convergence time in seconds for the different algorithms with initial value function $h = h_{min}$ and $\epsilon = 10^{-3}$. Times for RTDP not shown as they exceed the cutoff time for convergence (10 minutes). Faster times are shown in bold font.

Conclusion

- Ajouter la notion de « label » à RTDP
  - Améliore la convergence
  - Trouve un résultat rapidement
  - converge en temps fini
- Processus de labélisation est complexe

UQÀM | Faculté des sciences
Université du Québec à Montréal

MERCI DE VOTRE ATTENTION

UQÀM | Faculté des sciences
Université du Québec à Montréal