INTELIGENTNI MULTIAGENTSKI SUSTAVI

PODSJETNIK v2.0

Racionalni agenti

Algoritam iteracije vrijednosti:

```
VALUE-ITERATION(T, r, γ, ε)
      do
         u ← u'
         δ ← 0
         for s ∈ S
               do u'(s) ← r(s) + γ max_a Σ_{s'} T(s, a, s')u(s')
                  if |u'(s) − u(s)| > δ
                     then δ ← |u'(s) − u(s)|
         until δ < ε(1 − γ)/γ
 return u
```

Očekivana funkcija korisnosti:

$$E[u_i, s, a] = \sum_{s' \in S} T(s, a, s') u_i(s'),$$

Sebična politika:

$$\pi_i^*(s) = \arg\max_{a \in A} E[u_i, s, a]$$

Raspodijeljeno zadovoljavanje ograničenja

Pretraživanje u dubinu:

```
DEPTH-FIRST-SEARCH-CSP(i, g)
 if i > n
    then return g
 for v ∈ D_i
    do if setting x_i ← v does not violate any constraint in P given g
          then g' ← DEPTH-FIRST-SEARCH-CSP(i + 1, g + {x_i ← v})
               if g' ≠ ∅
                  then return g'

 return ∅
```

Rezolucijsko zaključivanje:

$$\frac{\begin{array}{c} A_1 \vee A_2 \vee \cdots \vee A_m \\ \neg(A_1 \wedge A_{11} \wedge \cdots) \\ \neg(A_2 \wedge A_{21} \wedge \cdots) \\ \vdots \\ \neg(A_m \wedge A_{m1} \wedge \cdots) \end{array}}{\neg(A_{11} \wedge \cdots \wedge A_{21} \wedge \cdots \wedge A_{m1} \wedge \cdots)}.$$

Algoritam filtriranja:

```
FILTERING()
 for j ∈ {neighbors of i}
       do REVISE(x_i, x_j)
```

```
HANDLE-NEW-DOMAIN(j, D')
 D_j ← D'
 REVISE(x_i, x_j)
```

```
REVISE(x_i, x_j)
 old-domain ← D_i
 for v_i ∈ D_i
       do if there is no v_j ∈ D_j consistent with v_i
             then D_i ← D_i − v_i
 if old-domain ≠ D_i
    then ∀_{k∈{neighbors of i}} k.HANDLE-NEW-DOMAIN(i, D_i)
```

Algoritam asinkronog vraćanja:

```
HANDLE-OK?(j, x_j)
 local-view ← local-view + (j, x_j)
 CHECK-LOCAL-VIEW()
```

```
CHECK-LOCAL-VIEW()
 if local-view and x_i are not consistent
    then if no value in D_i is consistent with local-view
            then BACKTRACK()
            else select d ∈ D_i consistent with local-view
                 x_i ← d
                 ∀_{k∈neighbors} k.HANDLE-OK?(i, x_i)
```

```
HANDLE-NOGOOD(j, nogood)
 record nogood as a new constraint
 for (k, x_k) ∈ nogood where k ∉ neighbors
       do k.HANDLE-ADD-NEIGHBOR(i)
          neighbors ← neighbors + k
          local-view ← local-view + (k, x_k)
 old-value ← x_i
 CHECK-LOCAL-VIEW()
 if old-value ≠ x_i
    then j.HANDLE-OK?(i, x_i)
```

```
HANDLE-ADD-NEIGHBOR(j)
 neighbors ← neighbors + j
```

```
BACKTRACK()
 nogoods ← {V | V = inconsistent subset of local-view
                    using hyper-resolution rule}
 if an empty set is an element of nogoods
    then broadcast that there is no solution
         terminate this algorithm
 for V ∈ nogoods
       do select (j, x_j) where j has lowest priority in V
          j.HANDLE-NOGOOD(i, V)
          local-view ← local-view − (j, x_j)
 CHECK-LOCAL-VIEW()
```

Teorija igara

Maxmin strategija: $s_i^* = \max_{s_i} \min_{s_j} u_i(s_i, s_j)$

Strategija iterativne dominacije: $\forall_{s_{-i}} \forall_{r_i \neq s_i} u_i(s_{-i}, s_i) \geq u_i(s_{-i}, r_i)$

Strategija društvene dobrobiti: $s^* = \arg\max_{s} \sum_{i} u_i(s)$

Nashova ravnoteža: $\{s \mid \forall_i \forall_{a_i \neq s_i} u_i(s_{-i}, s_i) \geq u_i(s_{-i}, a_i)\}$

Pareto optimalna strategija: $\{s \mid \neg\exists_{s' \neq s}(\exists_i u_i(s') > u_i(s) \wedge \neg\exists_{j \in -i} u_j(s) > u_j(s'))\}$

Teorija igara — dodatna razmatranja

Dodjeljivanje uloga:

For each agent i in parallel
  $I = \{\}$.
  For each role $j = 1, \ldots, n$
    Compute the potential r_{ij} of agent i for role j.
    Broadcast r_{ij} to all agents.
  End
  Wait until all $r_{i'j}$, for $j = 1, \ldots, n$, are received.
  For each role $j = 1, \ldots, n$
    Assign role j to agent $i^* = \arg\max_{i' \notin I}\{r_{i'j}\}$.
    Add i^* to I.
  End
End

Shapleyeva vrijednost:

$$\phi(A, i) = \frac{1}{A!} \sum_{\pi \in \Pi_A} v(B(\pi, i) \cup i) - v(B(\pi, i))$$

Koordinacija grafova:

For each agent i in parallel
  If $i \neq 1$
    Wait until agent $i-1$ sends OK.
  End
  Let $f_j(a_i, a_{-i})$ be all local payoff functions (initial and communicated) that involve agent i.
  Compute $B_i(a_{-i}) = \arg\max_{a_i} \sum_j f_j(a_i, a_{-i})$.
  Compute $f^*(a_{-i}) = \max_{a_i} \sum_j f_j(a_i, a_{-i})$.
  Send $f^*(a_{-i})$ to agent $j = \min\{i+1, \ldots, n\}$, $j \in -i$.
  If $i \neq n$
    Send OK to agent $i+1$.
    Wait until all a^*_{-i} are received.
  End
  Choose any $a^*_i \in B_i(a^*_{-i})$.
  Broadcast a^*_i to all agents j such that $a_i \in \text{domain}(B_j)$.
End

Traženje koalicije:

FIND-COALITION(i)

$L_i \leftarrow$ set of all coalitions that include i.
$S^*_i \leftarrow \arg\max_{S \in L_i} v_i(S)$
Broadcast S^*_i and wait for all other broadcasts, put these into S^*
$S_{max} \leftarrow \arg\max_{s \in S^*} v_i(s)$
if $i \in S_{\max}$
  then join $S_{\max}$
    return
for $j \in S_{\max}$
  do Delete all coalitions in L_i that contain j
if L_i is not empty
  then goto 2
return

Nukleolus:

$\theta(\vec{u}) = \langle e(S_1^{\vec{u}}, \vec{u}), e(S_2^{\vec{u}}, \vec{u}), \ldots, e(S_{2^{|A|}}^{\vec{u}}, \vec{u}) \rangle$ $\{\vec{u} \mid \theta(\vec{u}) \not\succ \theta(\vec{v})$ *for all* $\vec{v}$*, given that* $\vec{u}$ *and* $\vec{v}$ *are feasible.*$\}$

Učenje u višeagentskim sustavima

Algoritam AWESOME:

AWESOME

$\pi \leftarrow$ equilibrium strategy
repeat
  playing-equilibrium $\leftarrow$ TRUE
  playing-stationary $\leftarrow$ TRUE
  playing-equilibrium-just-rejected $\leftarrow$ FALSE
  $\phi \leftarrow \pi_i$
  $t \leftarrow 0$
  while *playing-stationary*
    do play ϕ for N^t times in a row (an epoch)
      $\forall_j$ update s_j given what they played in these N^t rounds.
      if *playing-equilibrium*
        then if some player j has $\max_a(s_j(a), \pi_j(a)) > \epsilon_e$
          then *playing-equilibrium-just-rejected* $\leftarrow$ TRUE
            $\phi \leftarrow$ random action
        else if *playing-equilibrium-just-rejected* = FALSE
          and some j has $\max_a(s_j^{\text{old}}(a), s_j(a)) > \epsilon_s$
          then *playing-stationary* $\leftarrow$ FALSE
      playing-equilibrium-just-rejected $\leftarrow$ FALSE
      $b \leftarrow \arg\max_a u_i(a, s_{-i})$
      if $u_i(b, s_{-i}) > u_i(\phi, s_{-i}) + n|A_i|\epsilon_s^{t+1}\mu$
        then $\phi \leftarrow b$
      $\forall_j s_j^{\text{old}} \leftarrow s_j$
      $t \leftarrow t + 1$

Fiktivna igra:

$$k_i^t(s_j) = k_i^{t-1}(s_j) + \begin{cases} 1 & \text{if } s_j^{t-1} = s_j, \\ 0 & \text{if } s_j^{t-1} \neq s_j. \end{cases} \quad \mathbf{Pr}_i^t[s_j] = \frac{k_i^t(s_j)}{\sum_{\tilde{s}_j \in S_j} k_i^t(\tilde{s}_j)}$$

Dinamika replikatora:

$$\theta^t(s) = \frac{\phi^t(s)}{\sum_{s' \in S} \phi^t(s')} \qquad u^t(s) = \sum_{s' \in S} \theta^t(s') u(s, s') \qquad \phi^{t+1}(s) = \phi^t(s)(1 + u^t(s))$$

Friend — or — foe:

FRIEND-OR-FOE

$t \leftarrow 0$
$s_0 \leftarrow$ current state
$\forall_{s \in S} \forall_{a_j \in A_j} Q_i^t(s, a_1, \ldots, a_n) \leftarrow 0$
Choose action a_i^t
$s \leftarrow s'$
Observe $r_1^t, \ldots, r_n^t$; $a_1^t, \ldots, a_n^t$; s'
$Q_i^{t+1}(s, a_1, \ldots, a_n) \leftarrow$
  $(1 - \lambda^t) Q_i^t(s, a_1, \ldots, a_n) + \lambda^t(r_i^t + \gamma\, NashQ_i^t(s'))$
  where $NashQ_i^t(s') = \max_{\pi \in \Pi(X_1 \times \cdots \times X_k)} \min_{y_i, \ldots, y_l \in Y_1 \times \cdots \times Y_l}$
    $\sum_{x_1, \ldots, x_k \in X_1 \times \cdots \times X_k} \pi(x_1) \cdots \pi(x_k) Q_i(s, x_1, \ldots, x_k, y_1, \ldots y_l)$
  and X are actions for i's friends and Y are for the foes.
$t \leftarrow t + 1$
goto 4

COIN:

$$P_i(s, \vec{a}) = \frac{\sum_{\vec{a}' \in \vec{A}} \Theta[u_i(s, \vec{a}) - u_i(s, \vec{a}')]}{|\vec{A}|} \qquad \Omega_i(s, \vec{a}) = \sum_{\vec{a}' \in \vec{A}} \mathbf{Pr}[\vec{a}'] \frac{|u_i(s, \vec{a}) - u_i(s, \vec{a}'_{-i}, \vec{a}_i)|}{|u_i(s, \vec{a}) - u_i(s, \vec{a}_{-i}, \vec{a}'_i)|}$$

Q-učenje:

Q-LEARNING

$\forall_s \forall_a Q(s, a) \leftarrow 0$; $\lambda \leftarrow 1$; $\epsilon \leftarrow 1$
$s \leftarrow$ current state
if RAND() $< \epsilon$ ▷ Exploration rate
  then $a \leftarrow$ random action
  else $a \leftarrow \arg\max_a Q(s, a)$
Take action a
Receive reward r
$s' \leftarrow$ current state
$Q(s, a) \leftarrow \lambda(r + \gamma \max_{a'} Q(s', a')) + (1 - \lambda) Q(s, a)$
$\lambda \leftarrow .99\lambda$
$\epsilon \leftarrow .98\epsilon$
goto 2

NashQ-učenje:

NASHQ-LEARNING

$t \leftarrow 0$
$s \leftarrow$ current state
$\forall_{s \in S} \forall_{j \leftarrow 1, \ldots, n} \forall_{a_j \in A_j} Q_j^t(s, a_1, \ldots, a_n) \leftarrow 0$
Choose action a_i^t
$s \leftarrow s'$
Observe $r_1^t, \ldots, r_n^t$; $a_1^t, \ldots, a_n^t$; s'
for $j \leftarrow 1, \ldots, n$
  do $Q_j^{t+1}(s, a_1, \ldots, a_n) \leftarrow$
    $(1 - \lambda^t) Q_j^t(s, a_1, \ldots, a_n) + \lambda^t(r_j^t + \gamma\, NashQ_j^t(s'))$
    where $NashQ_j^t(s') = Q_j^t(s', \pi_1(s') \cdots \pi_n(s'))$
    and $\pi_1(s') \cdots \pi_n(s')$ are Nash EP calculated from Q values
$t \leftarrow t + 1$
goto 4

Operatori modalnih logika:

Logika	Modaliteti	Značenje modaliteta
Modalna	$\Box\phi$	Nužno je ϕ
	$\Diamond\phi$	Moguće je ϕ
Deontička	$O\phi$	Obavezno je ϕ
	$P\phi$	Dozvoljeno je ϕ
	$F\phi$	Zabranjeno je ϕ
Vremenska	$G\phi$	Uvijek će biti ϕ
	$F\phi$	Biti će ϕ
	$H\phi$	Uvijek je bilo ϕ
Doksastička	$Bx\phi$	x vjeruje da ϕ

Prikaz mentalnih stavova agenata

Aksiomi K-logika:

(**NEC**) $\vdash A \models \vdash \Box A$
(**K**) $\Box(A \to B) \to (\Box A \to \Box B)$
(**D**) $\Box A \to \neg\Box\neg A$
(**4**) $\Box A \to \Box\Box A$
(**5**) $\neg\Box A \to \Box\neg\Box A$
(**T**) $\Box A \to A$