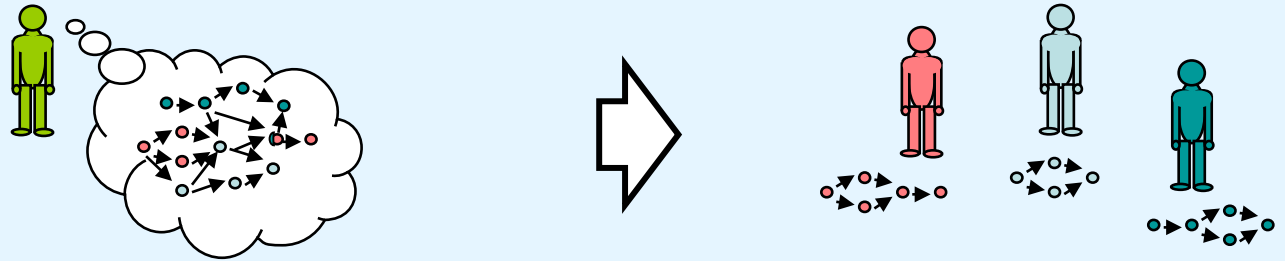


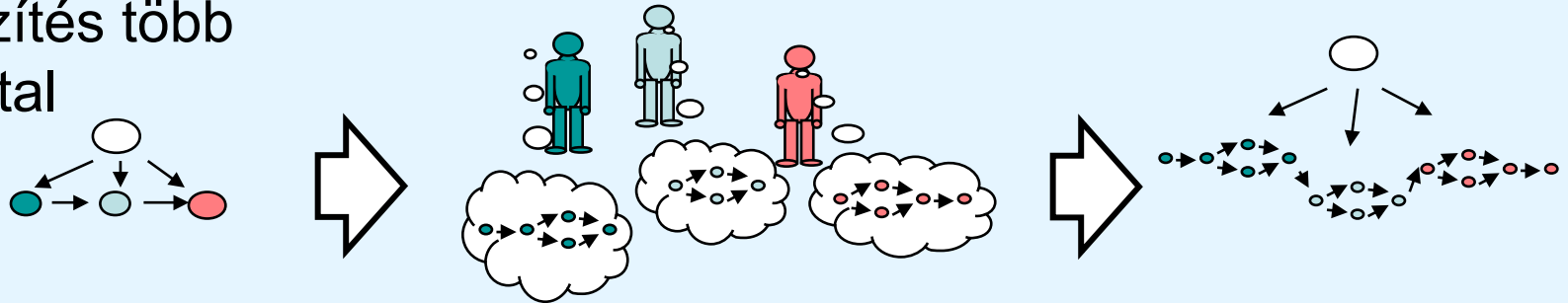
Kooperáció és intelligencia

MAP Multi Agent Planning/2

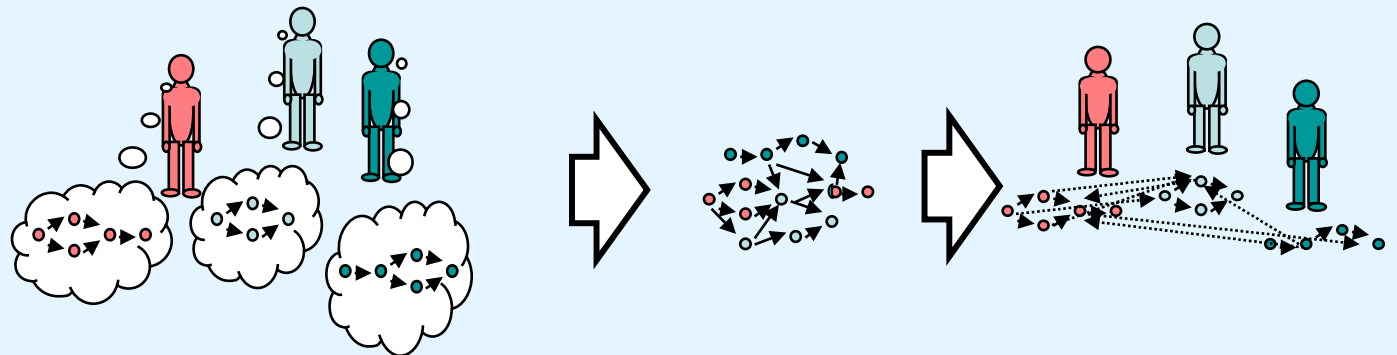
Tervkészítés
több ágens
számára



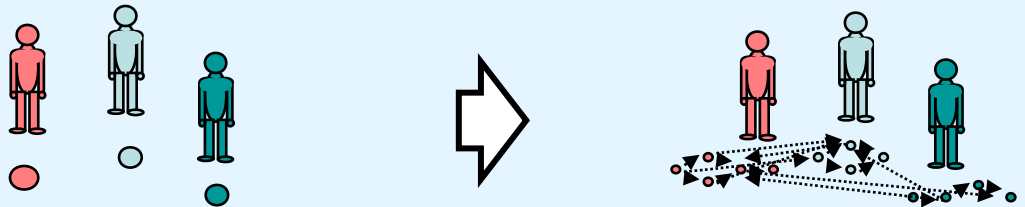
Tervkészítés több
ágens által



Ágens tervek
koordinálása
(tervfúzió)



Tervkészítés és
koordinálás
(elosztott tervekészítés)

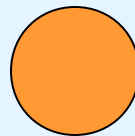
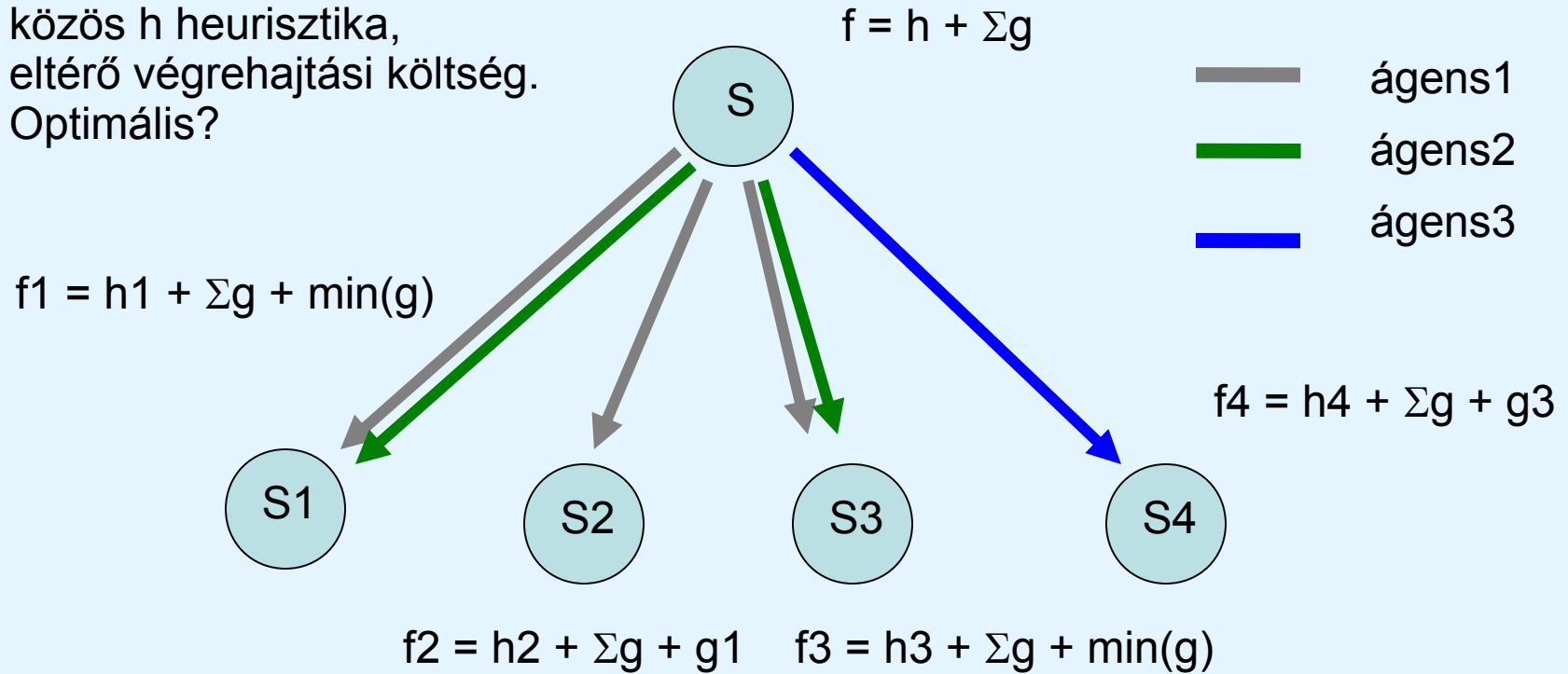


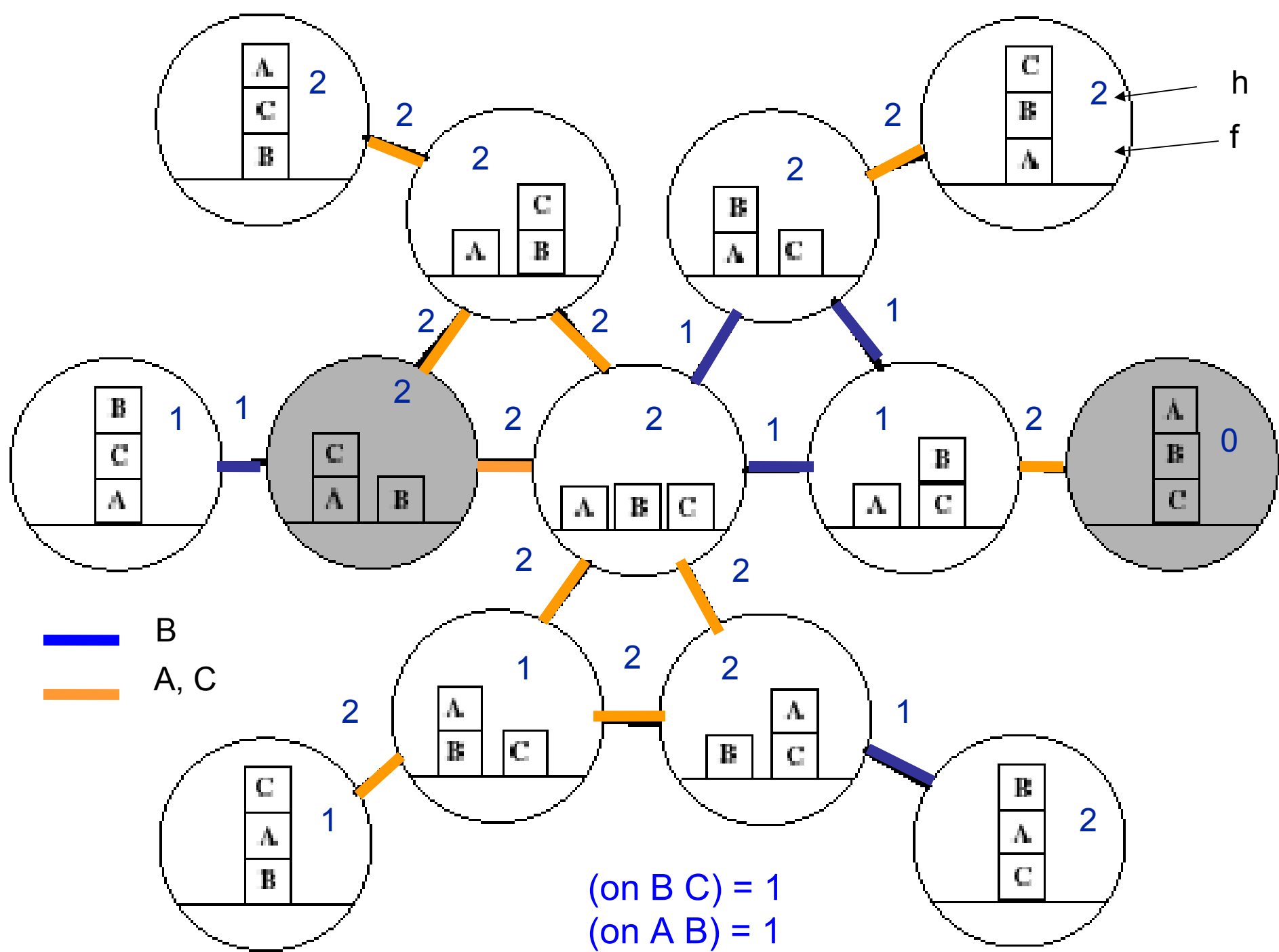
Több ágenses tervek egy ágenses technikákkal - MAS A* - informálisan

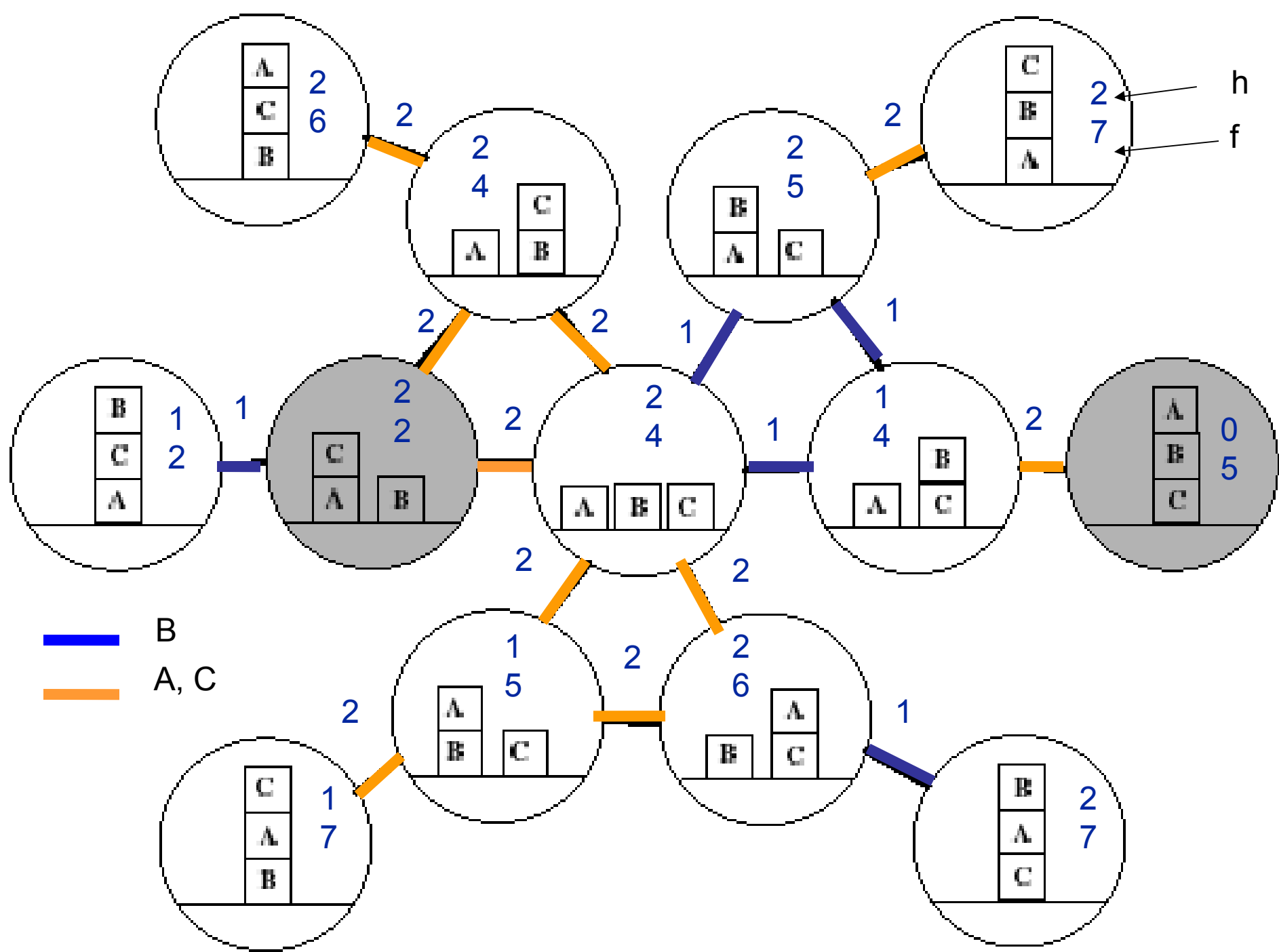
Modell:

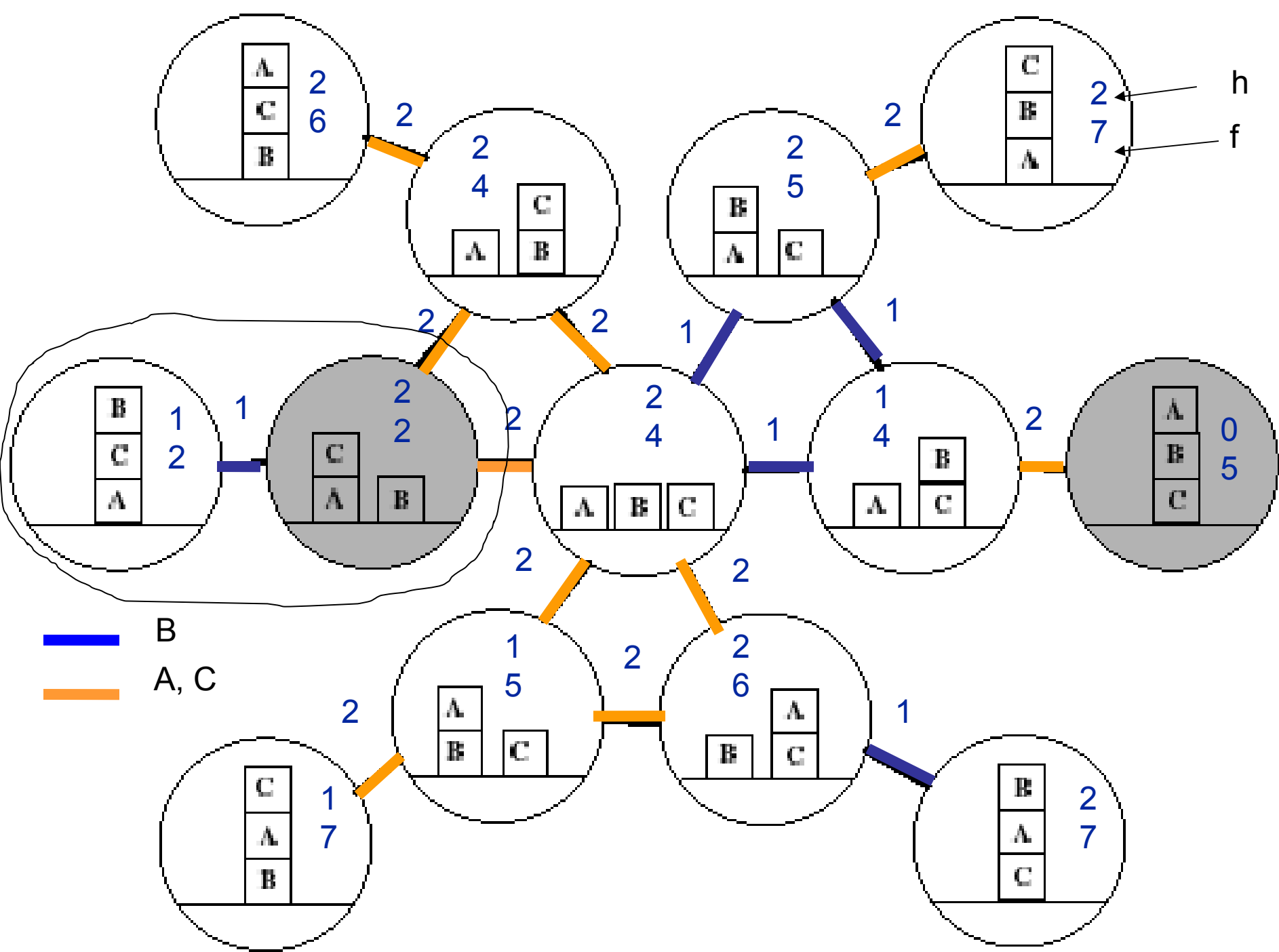
közös h heurisztika,
eltérő végrehajtási költség.

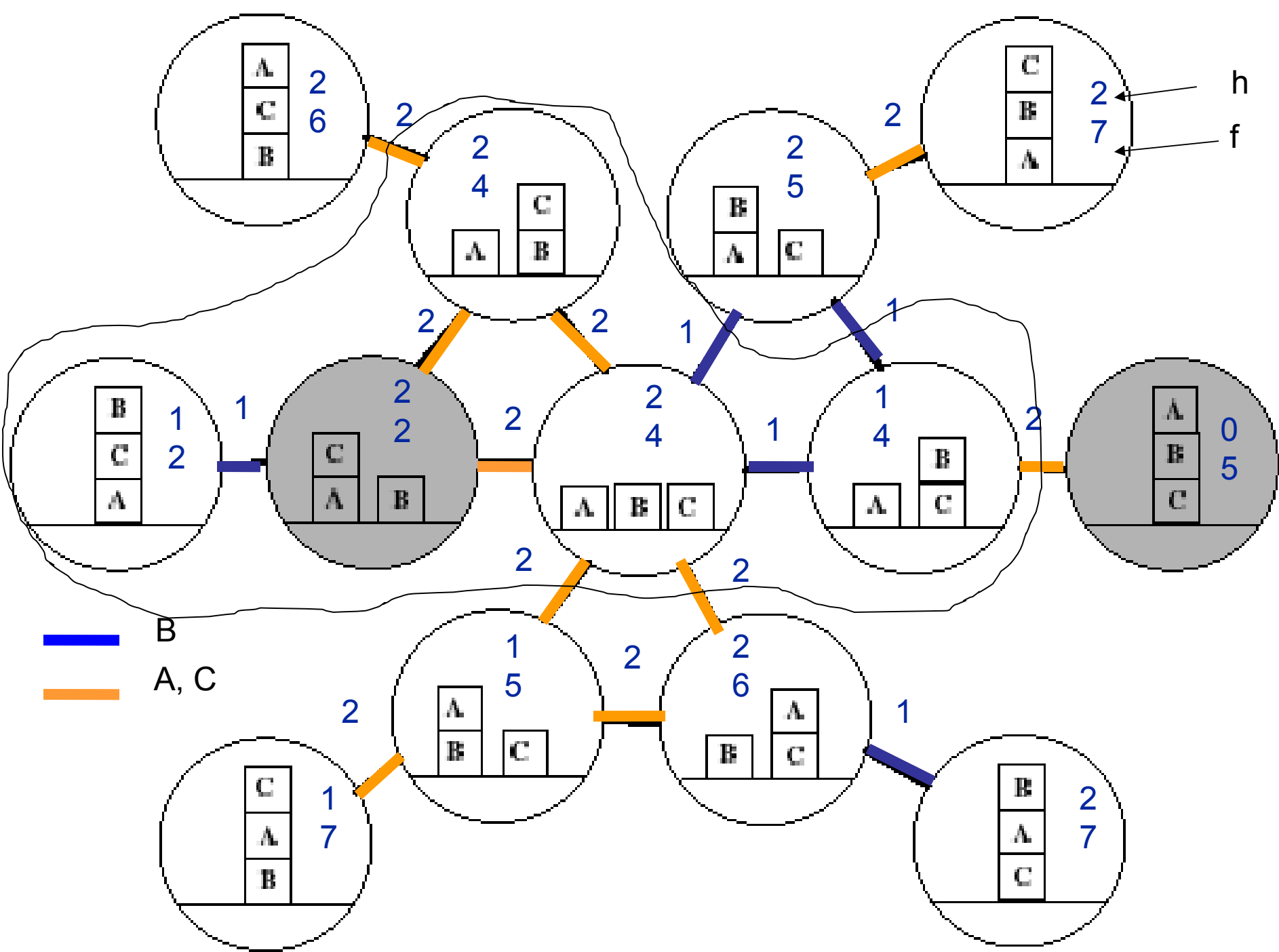
Optimális?

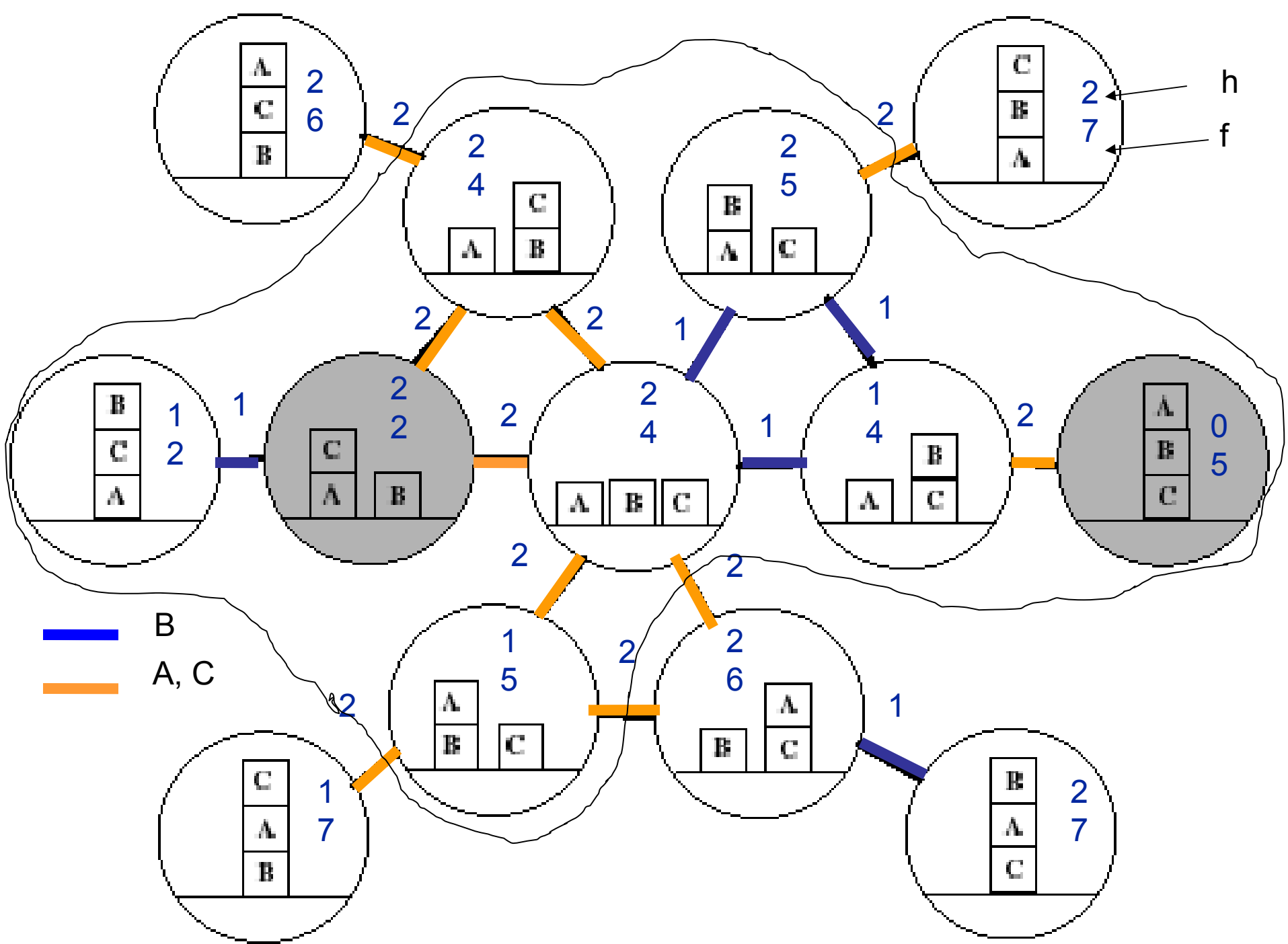


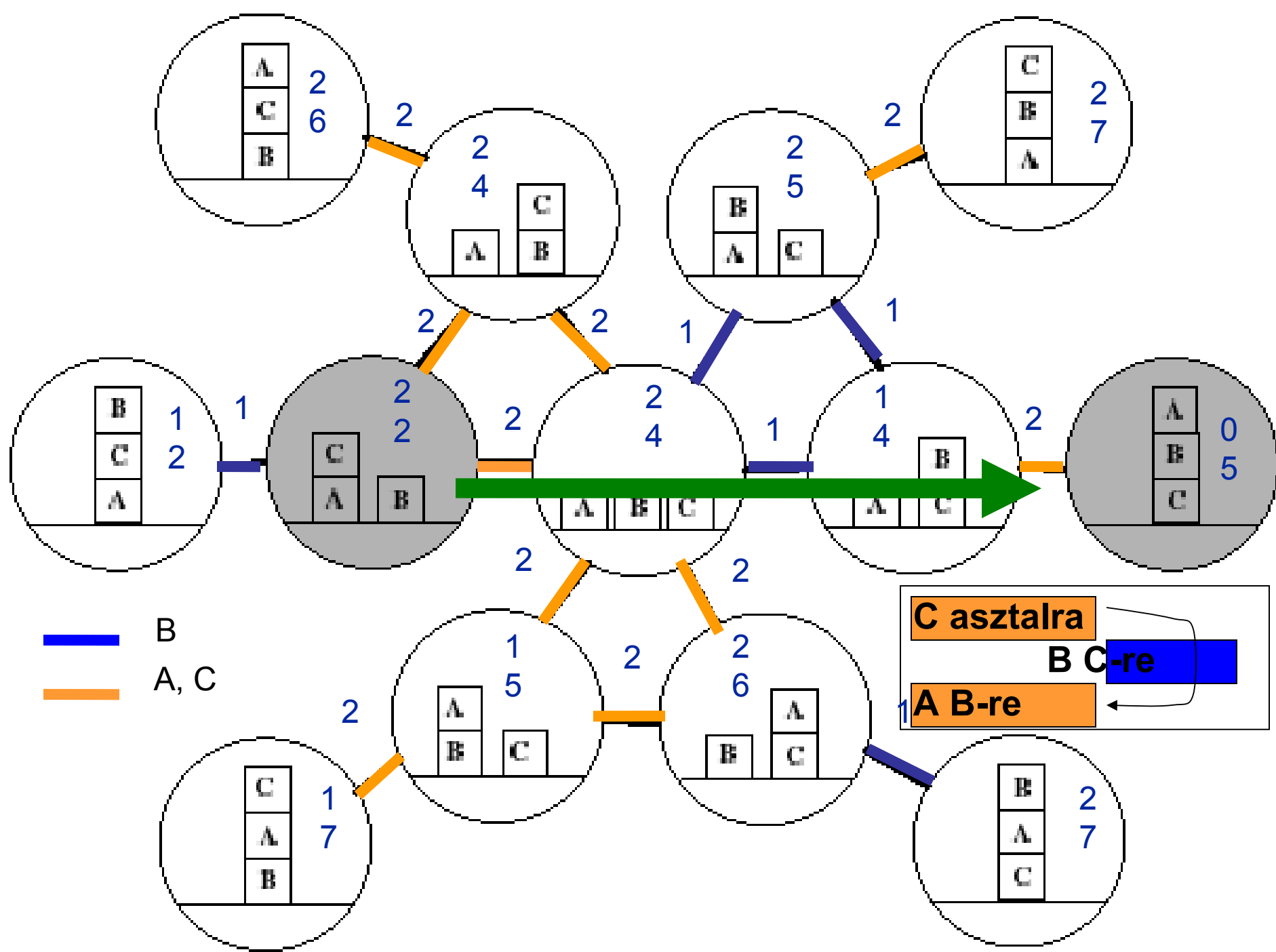


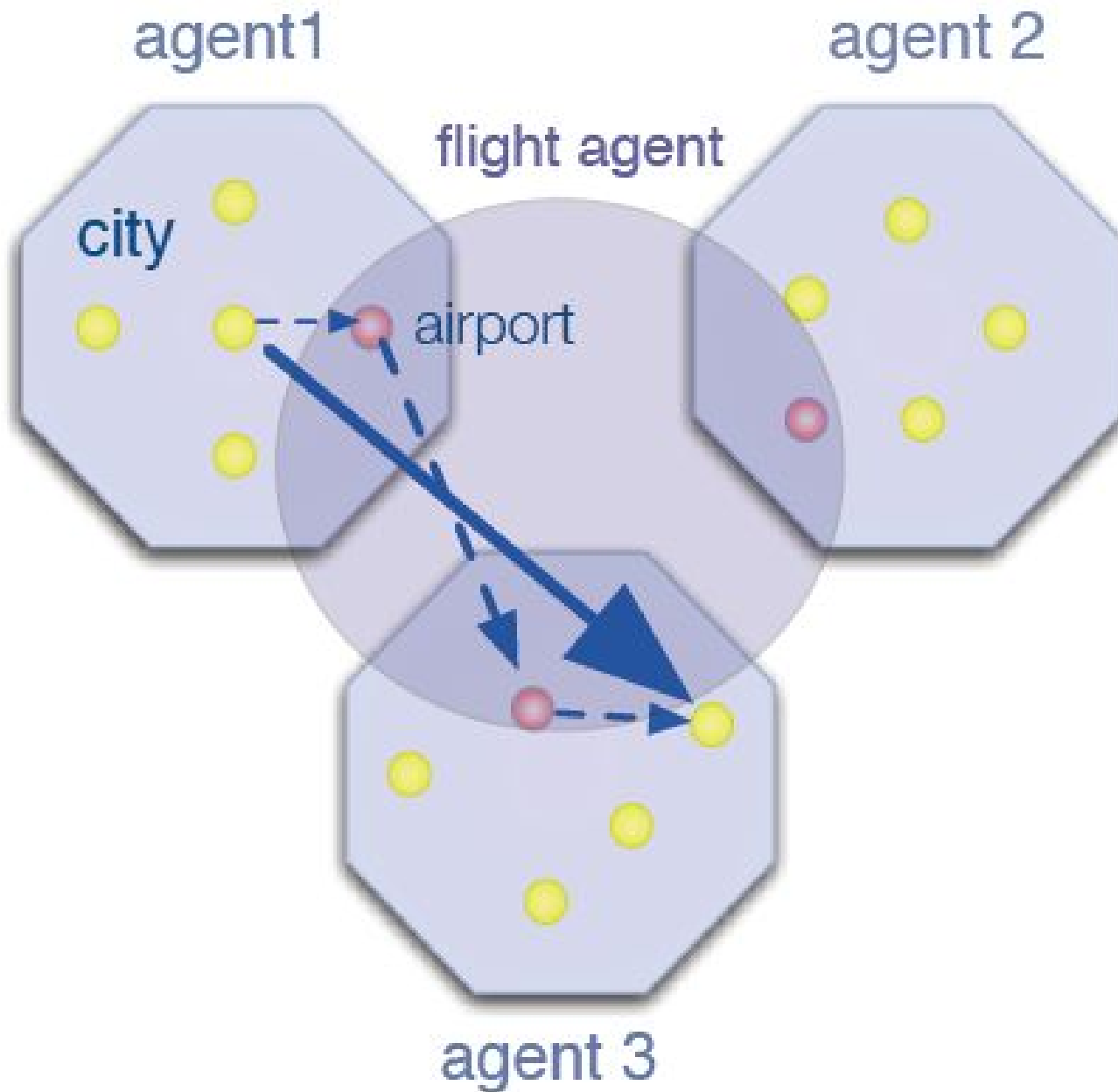


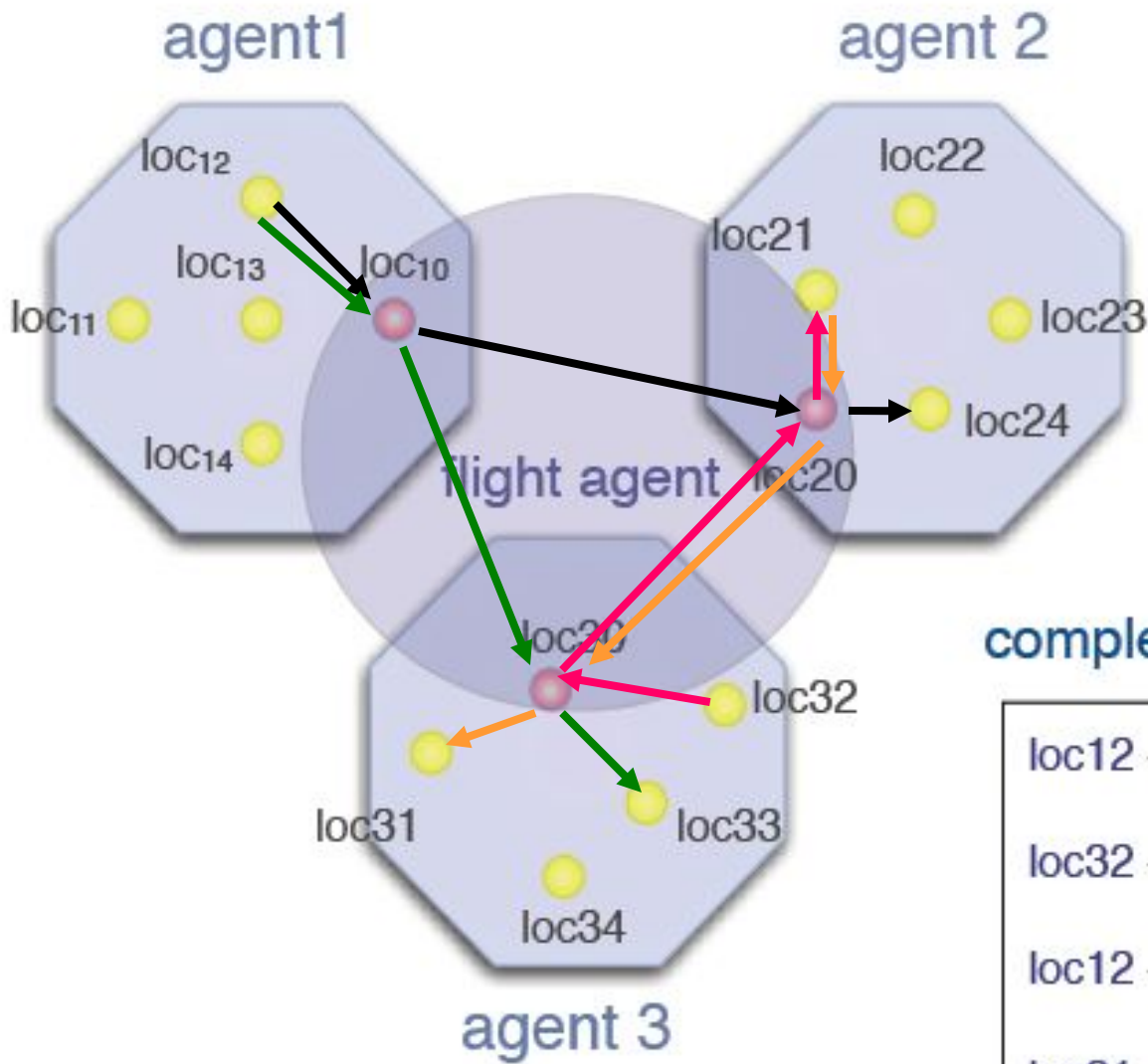




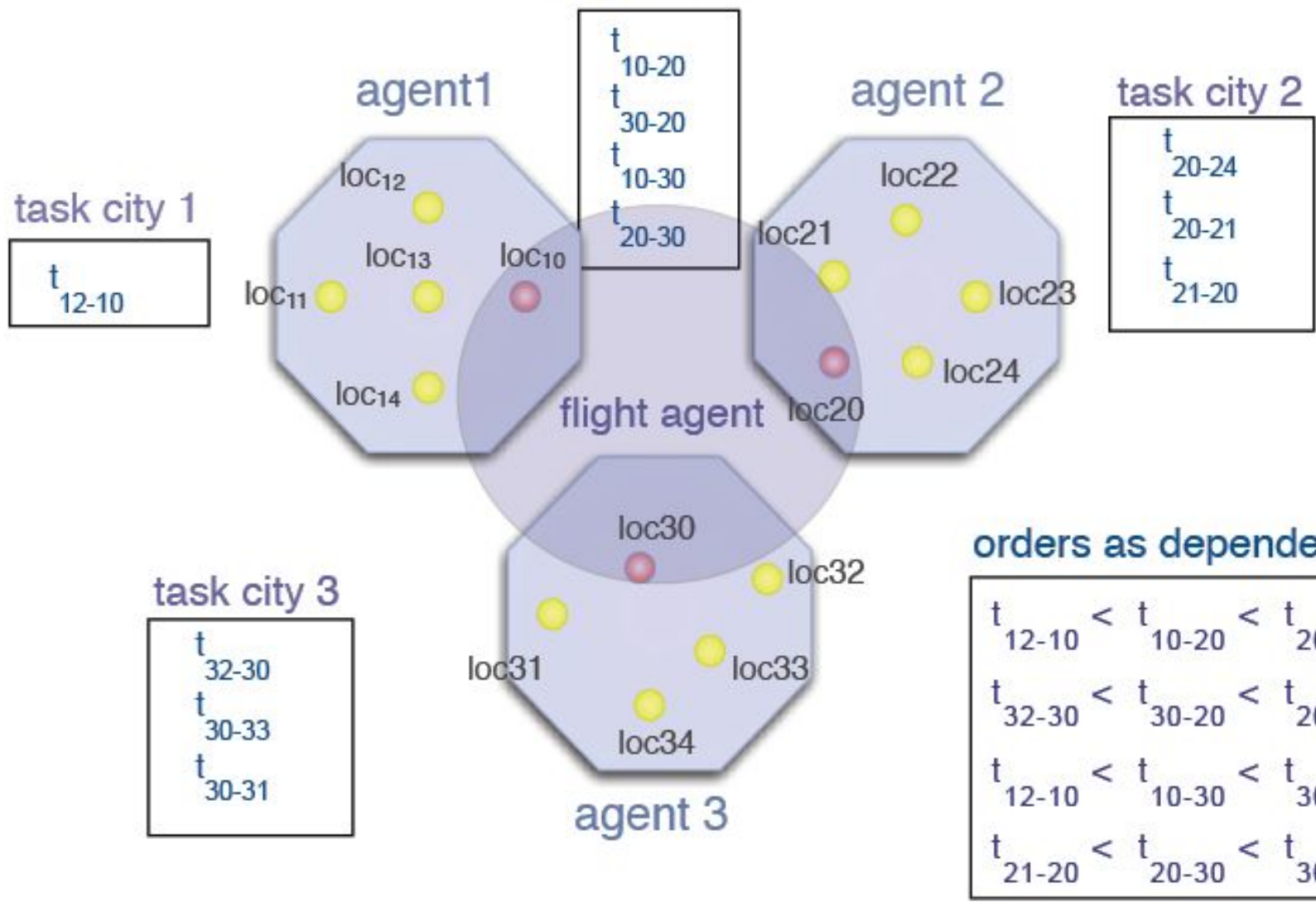








flight tasks



(define (domain taxi-repulo)

(:predicates (utas ?u) (taxi ?t) (hely ?h) (holu ?u ?h) (holt ?t ?h)
 (varos ?v) (varosban ?v ?h) (repulo ?r) (holr ?r ?h) (repter ?h))

(:action **taxi**

:parameters (?v ?u ?t ?honnann ?hova)

:precondition (and (utas ?u) (taxi ?t) (hely ?honnann) (hely ?hova) (holu ?u ?honnann)
 (holt ?t ?honnann) (varos ?v) (varosban ?v ?honnann) (varosban ?v ?hova))

:effect (and (holu ?u ?hova) (holt ?t ?hova)
 (not (holu ?u ?honnann)) (not (holt ?t ?honnann))))

(:action **taxi-uresjarat**

:parameters (?v ?t ?honnann ?hova)

:precondition (and (taxi ?t) (hely ?honnann) (hely ?hova)
 (holt ?t ?honnann) (varos ?v) (varosban ?v ?honnann) (varosban ?v ?hova))

:effect (and (holt ?t ?hova) (not (holt ?t ?honnann))))

(:action **repulo**

:parameters (?r ?u ?honnann ?hova)

:precondition (and (utas ?u) (repulo ?r) (repter ?honnann) (repter ?hova)
 (holu ?u ?honnann) (holr ?r ?honnann))

:effect (and (holu ?u ?hova) (holr ?r ?hova)
 (not (holu ?u ?honnann)) (not (holr ?r ?honnann))))

(:action **repulo-uresjarat**

:parameters (?r ?honnann ?hova)

:precondition (and (repulo ?r) (repter ?honnann) (repter ?hova) (holr ?r ?honnann))

:effect (and (holr ?r ?hova) (not (holr ?r ?honnann))))

```
(define (problem taxi-repulo)
  (:domain taxi-repulo)
  (:objects taxi1 taxi2 taxi3 airbus utas1 utas2 utas3 utas4 varos1 varos2 varos3
            hely12 hely13 hely10 hely11 hely14 hely20 hely21 hely22 hely23 hely24
            hely30 hely31 hely34 hely32 hely33)
  (:init (taxi taxi1) (taxi taxi2) (taxi taxi3) (repulo airbus)
         (utas utas1) (utas utas2) (utas utas3) (utas utas4)
         (varos varos1) (varos varos2) (varos varos3)
         (hely hely12) (hely hely13) (hely hely10) (hely hely11) (hely hely14) (hely hely20)
         (hely hely21) (hely hely22) (hely hely23) (hely hely24) (hely hely30) (hely hely31)
         (hely hely34) (hely hely32) (hely hely33)
         (repter hely10) (repter hely20) (repter hely30)
         (holu utas1 hely12) (holu utas2 hely32) (holu utas3 hely21) (holu utas4 hely12)
         (holt taxi1 hely12) (holt taxi2 hely21) (holt taxi3 hely32) (holr airbus hely10)
         (varosban varos1 hely12) (varosban varos1 hely13) (varosban varos1 hely10)
         (varosban varos1 hely11) (varosban varos1 hely14) (varosban varos2 hely21)
         (varosban varos2 hely20) (varosban varos2 hely22) (varosban varos2 hely23)
         (varosban varos2 hely24) (varosban varos3 hely31) (varosban varos3 hely30)
         (varosban varos3 hely32) (varosban varos3 hely33) (varosban varos3 hely34))
  (:goal (and (holu utas1 hely24) (holu utas2 hely21) (holu utas3 hely31) (holu utas4 hely33)))) )
```

(TAXI UTAS4 TAXI1 HELY12 HELY10)

(TAXI-URESJARAT TAXI3 HELY32 HELY30)

(TAXI UTAS3 TAXI2 HELY21 HELY20)

(REPULO AIRBUS UTAS4 HELY10 HELY30)

(TAXI-URESJARAT TAXI1 HELY10 HELY12)

(TAXI UTAS1 TAXI1 HELY12 HELY10)

(TAXI UTAS4 TAXI3 HELY30 HELY33)

(REPULO-URESJARAT AIRBUS HELY30 HELY10)

(REPULO AIRBUS UTAS1 HELY10 HELY20)

(TAXI-URESJARAT TAXI3 HELY33 HELY32)

(REPULO AIRBUS UTAS3 HELY20 HELY30)

(TAXI UTAS2 TAXI3 HELY32 HELY30)

(TAXI UTAS1 TAXI2 HELY20 HELY24)

(REPULO AIRBUS UTAS2 HELY30 HELY20)

(TAXI-URESJARAT TAXI2 HELY24 HELY20)

(TAXI UTAS3 TAXI3 HELY30 HELY31)

(TAXI UTAS2 TAXI2 HELY20 HELY21)

(TAXI UTAS4 TAXI1 HELY12 HELY10)

(TAXI-URESJARAT TAXI3 HELY32 HELY30)

(TAXI UTAS3 TAXI2 HELY21 HELY20)

(REPULO AIRBUS UTAS4 HELY10 HELY30)

(TAXI-URESJARAT TAXI1 HELY10 HELY12)

(TAXI UTAS1 TAXI1 HELY12 HELY10)

(TAXI UTAS4 TAXI3 HELY30 HELY33)

(REPULO-URESJARAT AIRBUS HELY30 HELY10)

(REPULO AIRBUS UTAS1 HELY10 HELY20)

(TAXI-URESJARAT TAXI3 HELY33 HELY32)

(REPULO AIRBUS UTAS3 HELY20 HELY30)

(TAXI UTAS2 TAXI3 HELY32 HELY30)

(TAXI UTAS1 TAXI2 HELY20 HELY24)

(REPULO AIRBUS UTAS2 HELY30 HELY20)

(TAXI-URESJARAT TAXI2 HELY24 HELY20)

(TAXI UTAS3 TAXI3 HELY30 HELY31)

(TAXI UTAS2 TAXI2 HELY20 HELY21)

(TAXI UTAS4 TAXI1 HELY12 HELY10)

(TAXI-URESJARAT TAXI3 HELY32 HELY30)

(TAXI UTAS3 TAXI2 HELY21 HELY20)

(REPULO AIRBUS UTAS4 HELY10 HELY30)

(TAXI-URESJARAT TAXI1 HELY10 HELY12)

(TAXI UTAS1 TAXI1 HELY12 HELY10)

(TAXI UTAS4 TAXI3 HELY30 HELY33)

(REPULO-URESJARAT AIRBUS HELY30 HELY10)

(REPULO AIRBUS UTAS1 HELY10 HELY20)

(TAXI-URESJARAT TAXI3 HELY33 HELY32)

(REPULO AIRBUS UTAS3 HELY20 HELY30)

(TAXI UTAS2 TAXI3 HELY32 HELY30)

(TAXI UTAS1 TAXI2 HELY20 HELY24)

(REPULO AIRBUS UTAS2 HELY30 HELY20)

(TAXI-URESJARAT TAXI2 HELY24 HELY20)

(TAXI UTAS3 TAXI3 HELY30 HELY31)

(TAXI UTAS2 TAXI2 HELY20 HELY21)

(TAXI UTAS4 TAXI1 HELY12 HELY10)

(TAXI-URESJARAT TAXI3 HELY32 HELY30)

(TAXI UTAS3 TAXI2 HELY21 HELY20)

(REPULO AIRBUS UTAS4 HELY10 HELY30)

(TAXI-URESJARAT TAXI1 HELY10 HELY12)

(TAXI UTAS1 TAXI1 HELY12 HELY10)

(TAXI UTAS4 TAXI3 HELY30 HELY33)

(REPULO-URESJARAT AIRBUS HELY30 HELY10)

(REPULO AIRBUS UTAS1 HELY10 HELY20)

(TAXI-URESJARAT TAXI3 HELY33 HELY32)

(REPULO AIRBUS UTAS3 HELY20 HELY30)

(TAXI UTAS2 TAXI3 HELY32 HELY30)

(TAXI UTAS1 TAXI2 HELY20 HELY24)

(REPULO AIRBUS UTAS2 HELY30 HELY20)

(TAXI-URESJARAT TAXI2 HELY24 HELY20)

(TAXI UTAS3 TAXI3 HELY30 HELY31)

(TAXI UTAS2 TAXI2 HELY20 HELY21)

(TAXI UTAS4 TAXI1 HELY12 HELY10)
 (TAXI-URESJARAT TAXI3 HELY32 HELY30)
 (TAXI UTAS3 TAXI2 HELY21 HELY20)
 (REPULO AIRBUS UTAS4 HELY10 HELY30)
(TAXI-URESJARAT TAXI1 HELY10 HELY12)
(TAXI UTAS1 TAXI1 HELY12 HELY10)
 (TAXI UTAS4 TAXI3 HELY30 HELY33)
 (REPULO-URESJARAT AIRBUS HELY30 HELY10)
 (REPULO AIRBUS UTAS1 HELY10 HELY20)
 (TAXI-URESJARAT TAXI3 HELY33 HELY32)
 (REPULO AIRBUS UTAS3 HELY20 HELY30)
 (TAXI UTAS2 TAXI3 HELY32 HELY30)
(TAXI UTAS1 TAXI2 HELY20 HELY24)
 (REPULO AIRBUS UTAS2 HELY30 HELY20)
(TAXI-URESJARAT TAXI2 HELY24 HELY20)
 (TAXI UTAS3 TAXI3 HELY30 HELY31)
(TAXI UTAS2 TAXI2 HELY20 HELY21)

Több ágenses tervek kereséssel – igazi MAS A*

Modell: MA-STRIPS

set of agents $\Phi = \{\varphi_i\}_{i=1}^k$

$\Pi = \langle P, \{A_i\}_{i=1}^k, I, G \rangle$ $I \subseteq P$ $G \subseteq P$

action $a = \langle \text{pre}(a), \text{eff}(a), \text{cost}(a) \rangle$

1. Privát/publikus információ

privát tény adott ágens által igényelt és értékben befolyásolt
cselekvés pre, eff csakis privát tények

publikus (megosztott) cselekvés
minden más cselekvés
(pre, eff elvben más ágensek hatása alatt is)

2. A közös célok elérését előíró cselekvések mind publikusak

3. „Privacy-preserving MA-STRIPS planning”

„weak privacy preserving”: nem szükséges a privát információkat közölni

„strong privacy preserving”: privát információ nem kikövetkeztethető

MAFS algoritmus

- open/closed listák (A^*)
- $f = h + g$ kritérium
- egy állapot kifejtésénél egy ágens csakis a saját cselekvéseit használja
- üzenet, ha egy **A** ágens által kifejtett **S** állapot releváns pl. **B** ágens számára:
 - **A** ágens az **S**-hez vezető cselekvés publikus
 - **B**-nek van egy olyan publikus cselekvése, melynek előfeltételei **S**-ben igazak

Algorithm 1 MAFS for agent φ_i

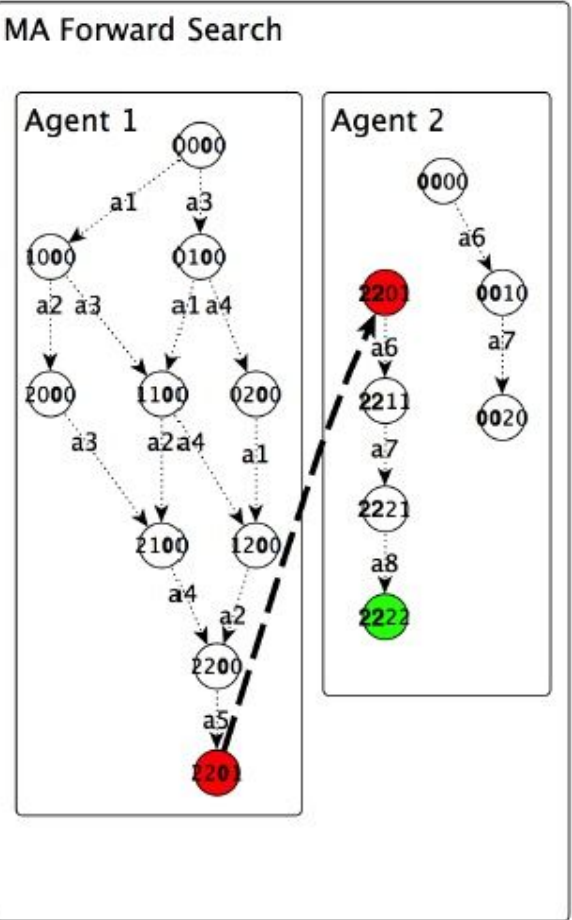
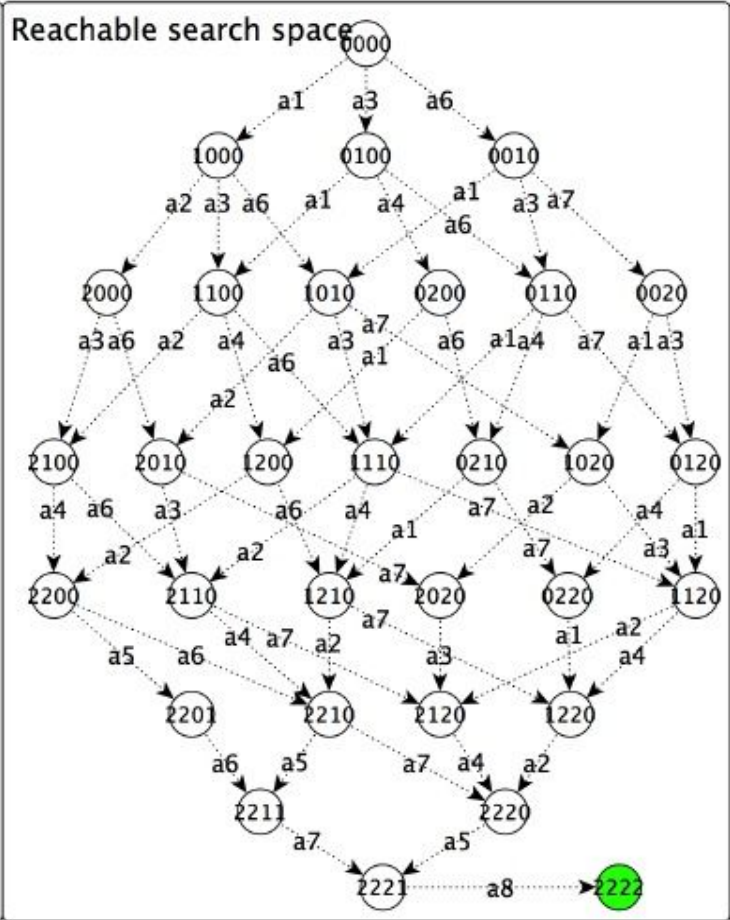
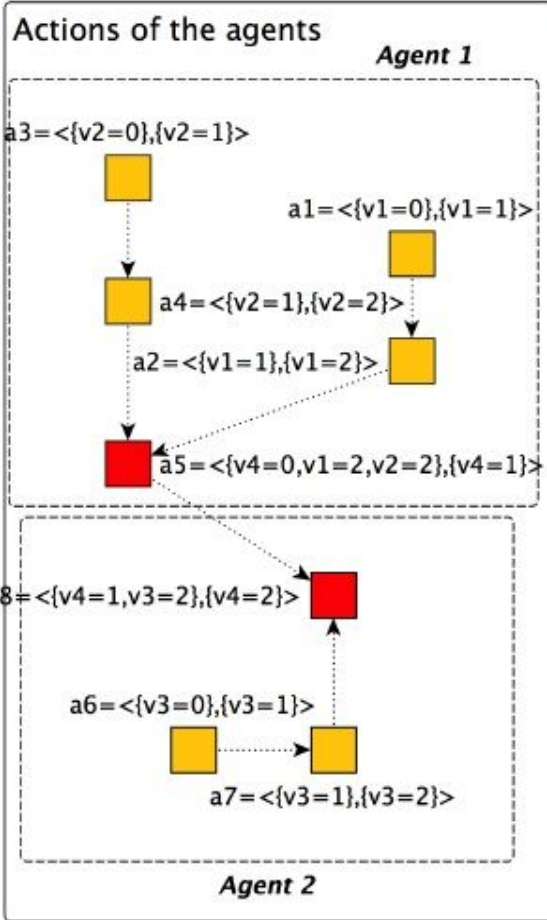
```
1: while TRUE do
2:   for all messages  $m$  in message queue do
3:     process-message( $m$ )
4:    $s \leftarrow \text{extract-min}(\text{open list})$ 
5:   expand( $s$ )
```

Algorithm 2 process-message($m = \langle s, g_{\varphi_j}(s), h_{\varphi_j}(s) \rangle$)

```
1: if  $s$  is not in open or closed list or  $g_{\varphi_i}(s) > g_{\varphi_j}(s)$  then
2:   add  $s$  to open list and calculate  $h_{\varphi_i}(s)$ 
3:    $g_{\varphi_i}(s) \leftarrow g_{\varphi_j}(s)$ 
4:    $h_{\varphi_i}(s) \leftarrow \max(h_{\varphi_i}(s), h_{\varphi_j}(s))$ 
```

Algorithm 3 expand(s)

```
1: move  $s$  to closed list
2: if  $s$  is a goal state then
3:   broadcast  $s$  to all agents
4:   if  $s$  has been broadcasted by all agents then
5:     return  $s$  as the solution
6: for all agents  $\varphi_j \in \Phi$  do
7:   if the last action leading to  $s$  was public and  $\varphi_j$  has a public action for which all
   public preconditions hold in  $s$  then
8:     send  $s$  to  $\varphi_j$ 
9: apply  $\varphi_i$ 's successor operator to  $s$ 
10: for all successors  $s'$  do
11:   update  $g_{\varphi_i}(s')$  and calculate  $h_{\varphi_i}(s')$ 
12:   if  $s'$  is not in closed list or  $f_{\varphi_i}(s')$  is now smaller than it was when  $s'$  was moved to
   closed list then
13:     move  $s'$  to open list
```



1. ágens: privát változói v1, v2, privát cselekvései sárgák

2. ágens: privát változói v3

v4 publikus (piros cselekvések publikusak)

Keresés menete: 0000 - xxx2

0000 kezdeti állapotból 2. ágens elakad, 1. ágens elér egy releváns állapotot, amit üzenettel 2. ágensnek adja át, aki a tervet most már befejezi

Collaborative Inductive Logic Programming – C-ILP

Útkeresés

ILP: háttértudás **B**
egy fogalom példái **E**
hipotézis **H**

$$B \wedge \mathbf{H} \models E$$

C- ILP

ágenshalmaz A , háttértudás $B_i, i \in A$,
pozitív és negatív példák halmaza E^+, E^- .
 $B = \cup B_i, E^+ = \cup E^+_i, E^- = \cup E^-_i$.

Együttműködő hipotézisalakítás (közös terv kialakítása):

$$B \wedge H \wedge E^- \not\models \emptyset$$

$$B \not\models E^+$$

$$B \wedge H \models E^+, \text{ and}$$

$$\neg \exists i \in A, B_i \wedge H \models E^+$$

Collaborative Inductive Logic Programming – C-ILP

Útkeresés

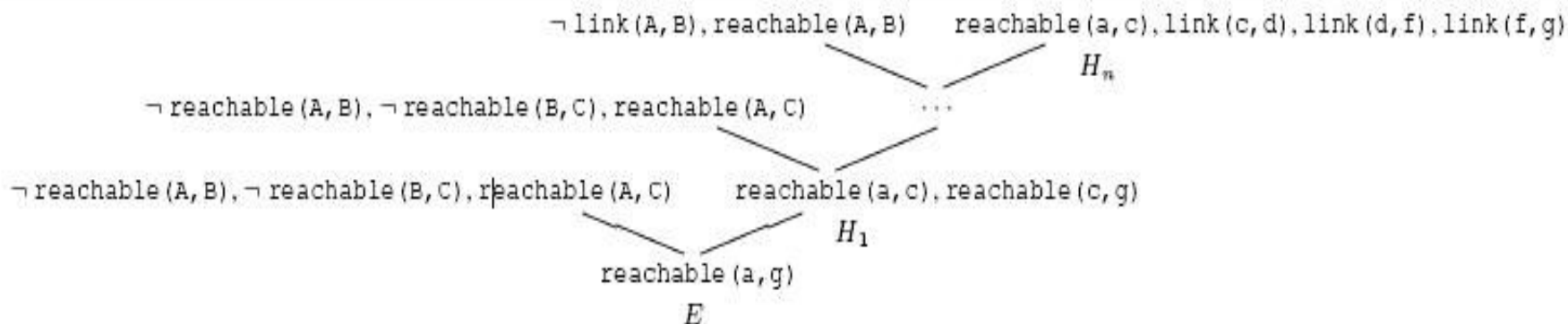
Minden ágens háttértudása:

1. $\text{link}(A,B) \rightarrow \text{reachable}(A,B)$
2. $\text{reachable}(A,B) \wedge \text{reachable}(B,C) \rightarrow \text{reachable}(A,C)$

Query: $\text{reachable}(A,B)$

Agens_k: magyarázó hipotézis $\mathbf{H} \wedge B_k \models \text{reachable}(A,B)$

Hipotézis generálás – **inverz rezolúció**



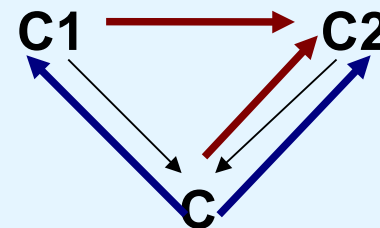
Collaborative Inductive Logic Programming – C-ILP

Útkeresés

Hipotézis generálás – **inverz rezolúció**

C (rezolvens) = $C1$ (klóz 1), $C2$ (klóz 2)

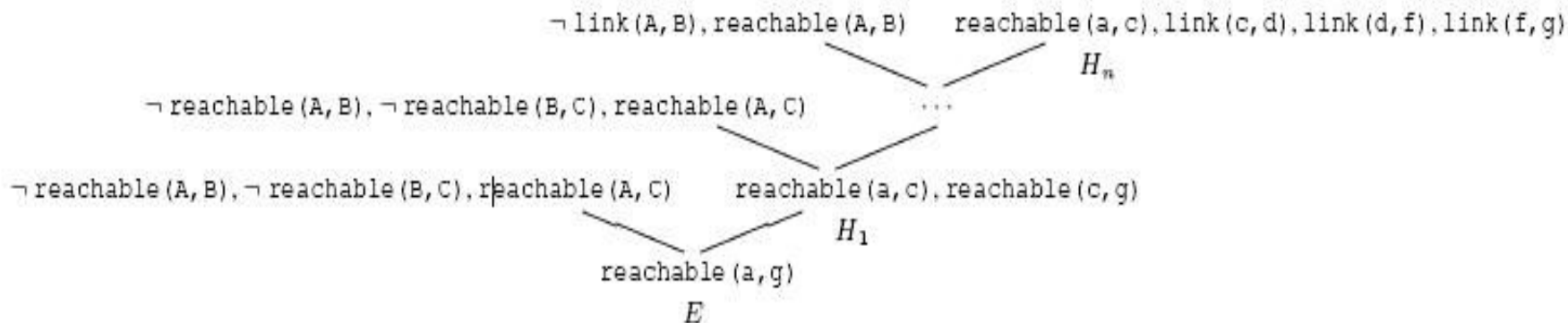
C (rezolvens), $C1$ (klóz 1) = $C2$ (klóz 2)



Illeszkedő klózok megválasztása – **keresés**, indeterminisztikus döntések

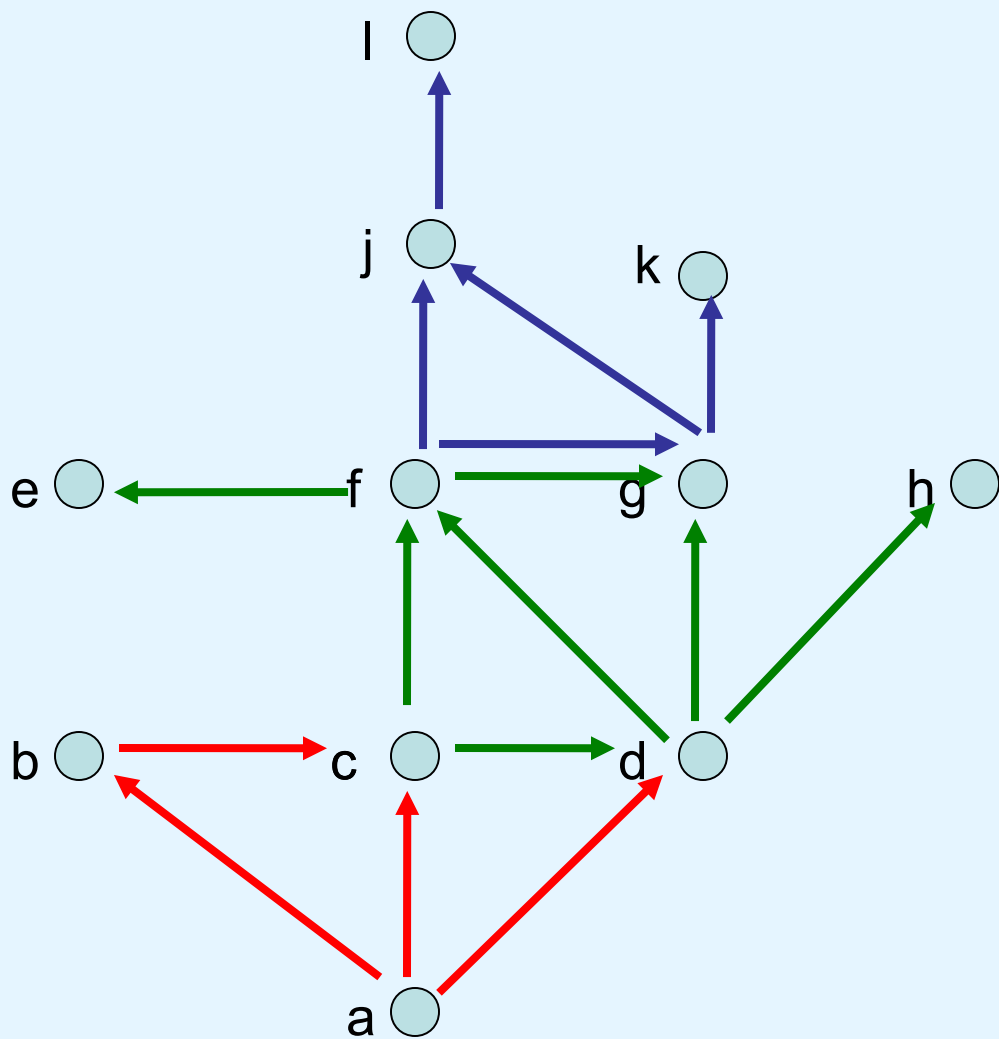
Hipotézis $H_n = \text{reachable}(a,c) \wedge \text{link}(c,d) \wedge \text{link}(d,f) \wedge \text{link}(f,g)$:

$\text{reachable}(a,g)$ (query) igaz, ha igaz a $\text{reachable}(a,c)$, ismert $\text{link}(c,d)$, $\text{link}(d,f)$ és $\text{link}(f,g)$ mellett.



Collaborative Inductive Logic Programming – C-ILP

Útkeresés



A : $E = \text{reachable}(a, l)$

A ASKS (B,C) : $E = \text{reachable}(a, l)$

C INDUCES: $H = \text{reachable}(a, g), \text{link}(g, j), \text{link}(j, l)$

C REPLIES: (if) $H = \text{reachable}(a, g)$

A DEDUCES: $K1 = K_a(K_c(\text{reachable}(g, l)))$

$K2 = K_a(\exists i K_i(\text{reachable}(a, g)) \rightarrow K_a(\text{reachable}(a, l)))$

A : $E = \text{reachable}(a, g)$

A ASKS (B,C) : $E = \text{reachable}(a, g)$

B INDUCES: $H = \text{reachable}(a, c), \text{link}(c, d), \text{link}(d, g)$

B REPLIES: (if) $H = \text{reachable}(a, c)$

A DEDUCES: $K3 = K_a(K_b(\text{reachable}(c, g)))$

$K4 = K_a(\exists i K_i(\text{reachable}(a, c)) \rightarrow K_a(\text{reachable}(a, g)))$

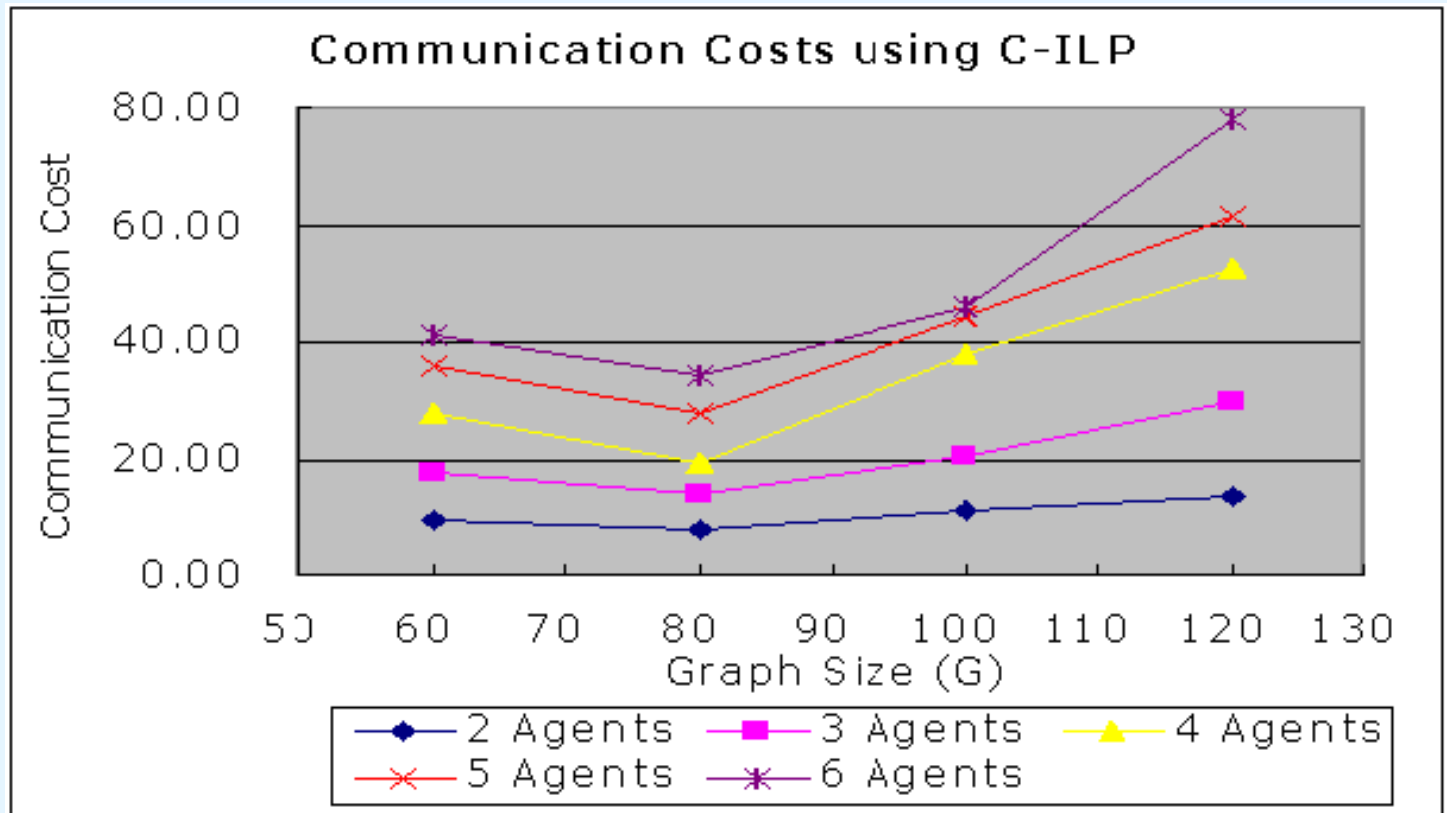
A : $E = \text{reachable}(a, c)$

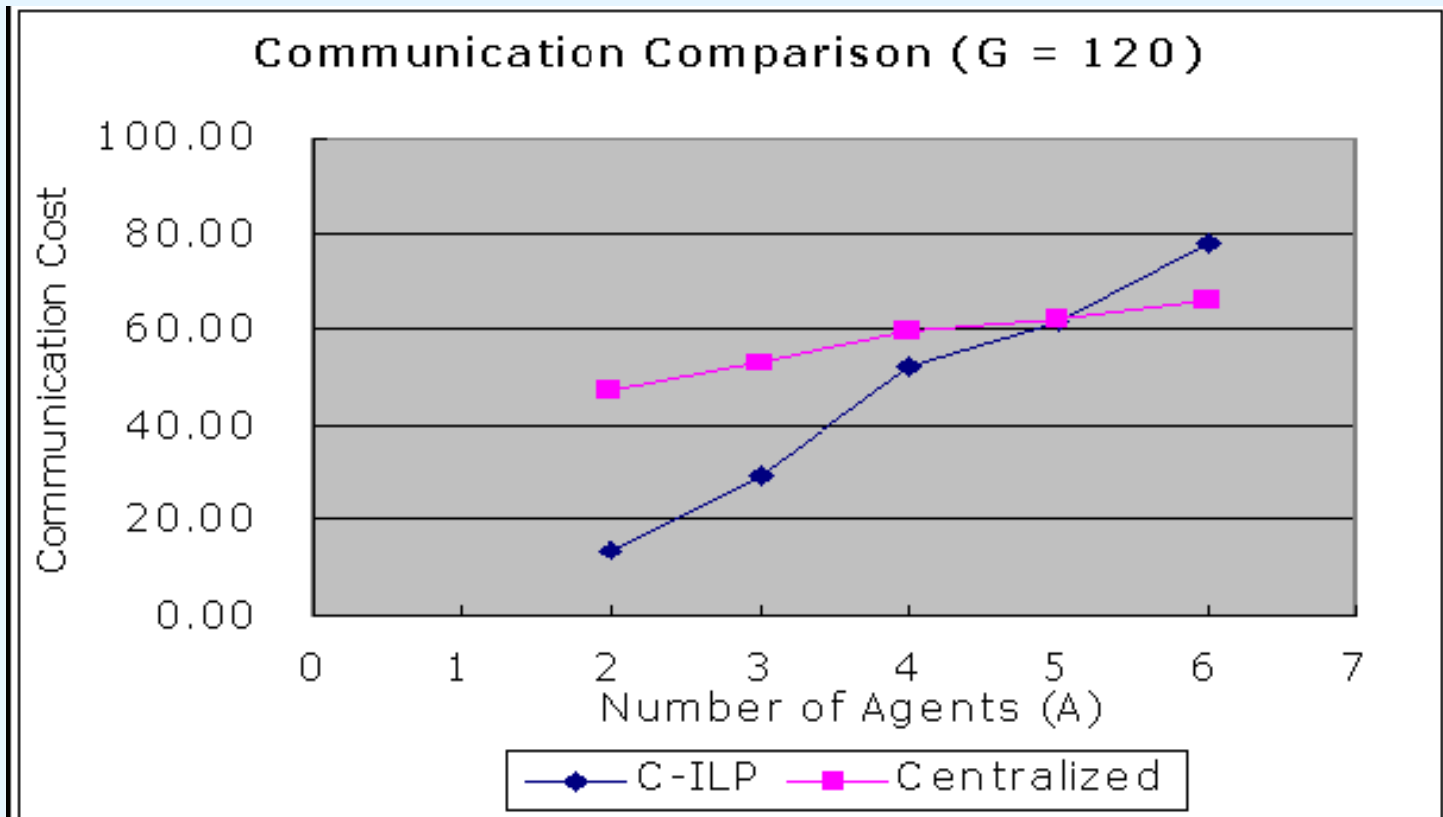
A INDUCES: $H = \text{link}(a, c)$

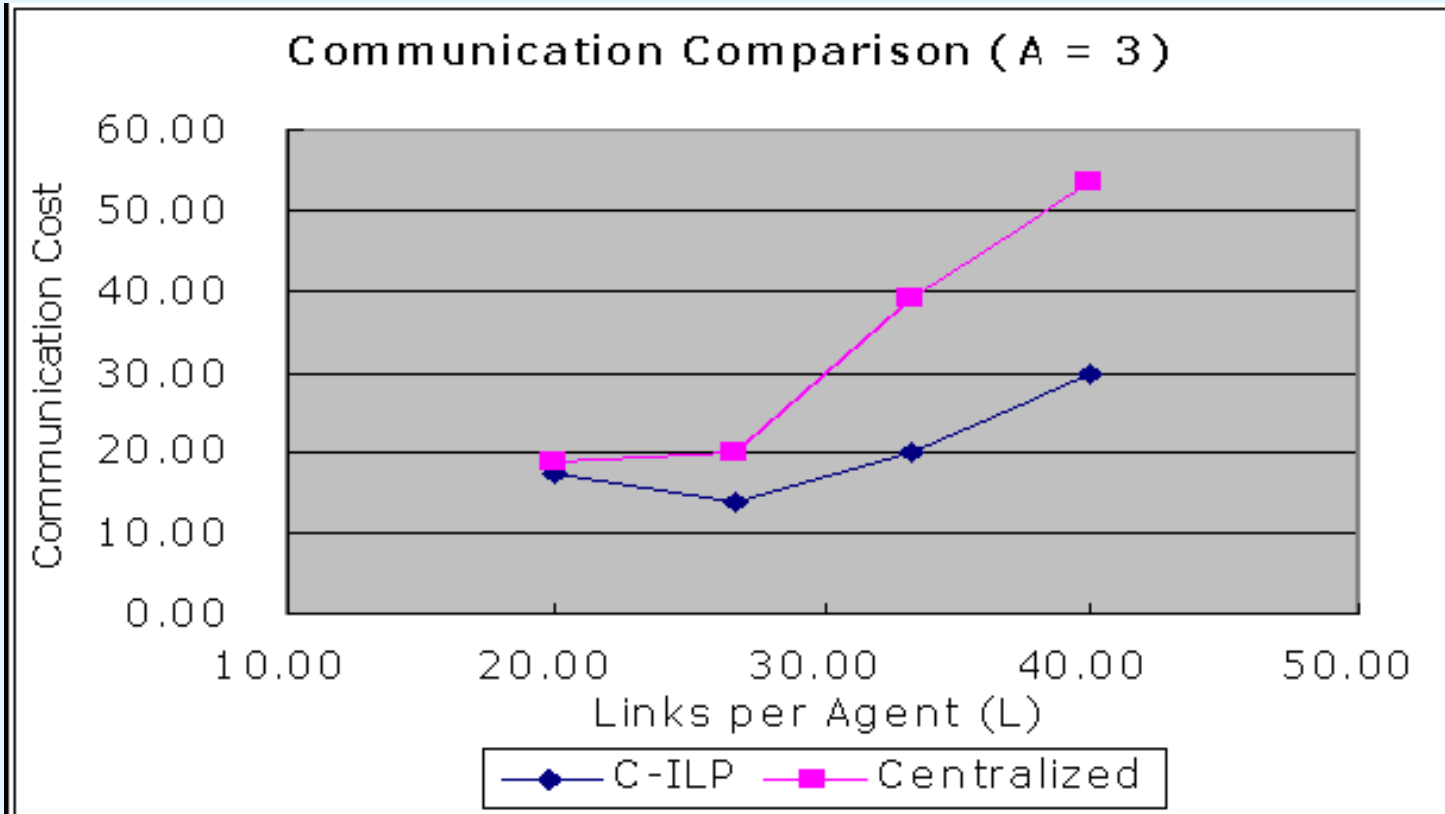
Kommunikációs nyereség:

- 1) Link információ közvetítése csak a pálya kiszámítása után.
- 2) Ha egy további D gk. a csomópontok ezreit tudná, csak azokat tudatja másokkal, amelyek az I csomóponthoz vezetnek.

További hipotézis diszkriminálás, az összes hipotézis elküldése helyett.







MAP problémák

- erősen csatolt ↔ független
terhek közös emelése, gépkocsi szerelés, robot foci, kórház, PhD kutatás, ...
- kooperatív ↔ önérdekű
- konfliktusfeloldó ↔ hatékonyságra törekvő
- kommunikáció nincs ↔ megbízható kommunikáció
 - SAR robotok, bolygó feltárás, katonai műveletek
- eset/kölcsönhatás gyakorisága

Technikák

- mikor készülnek a tervek?
- centralizált ↔ elosztott
- MAP **koordinálás** - mikor?
 - **tervkészítés előtt** ↔ **közben** ↔ **utána**
 - hogyan koordináljuk a terveket?
- függőség ellenőrzés?
- taszk/ erőforrás hozzárendelés?

Miért nehéz a koordinálás?

- Nincs globális rálátás
- Dinamikusan változó helyzetek
- Bizonytalanság a cselekvések kimenetelében
- Leképzési problémák számítási komplexitása
(kiválasztás + ütemezés)
- Skálázás #ágens-ben és #taszk-ban
- Határidők/ temporális nyomás
- Ágensek egyedi önérdeke
- **Megosztott hasznosság hiánya**

Koordinálás mechanizmusai

Statikus koordináció

- fejlesztési időben kialakított
- Pl.: a KRESZ, LRA

Dinamikus koordináció

- ágensek által futási időben „megtervezett”
- parametrizált statikus mechanizmus
- statikus alternatívák közti választás

Mikor koordinálni?

- Tervkészítés után?
- Tervkészítés közben?
- Tervkészítés előtt?
- Végrehajtás előtt?

Mit koordinálni?

- célok
- tervek
- cselekvések
- kényszerek
- időzítések
- állapotok
- erőforrások
- (semmi)

és kivel?

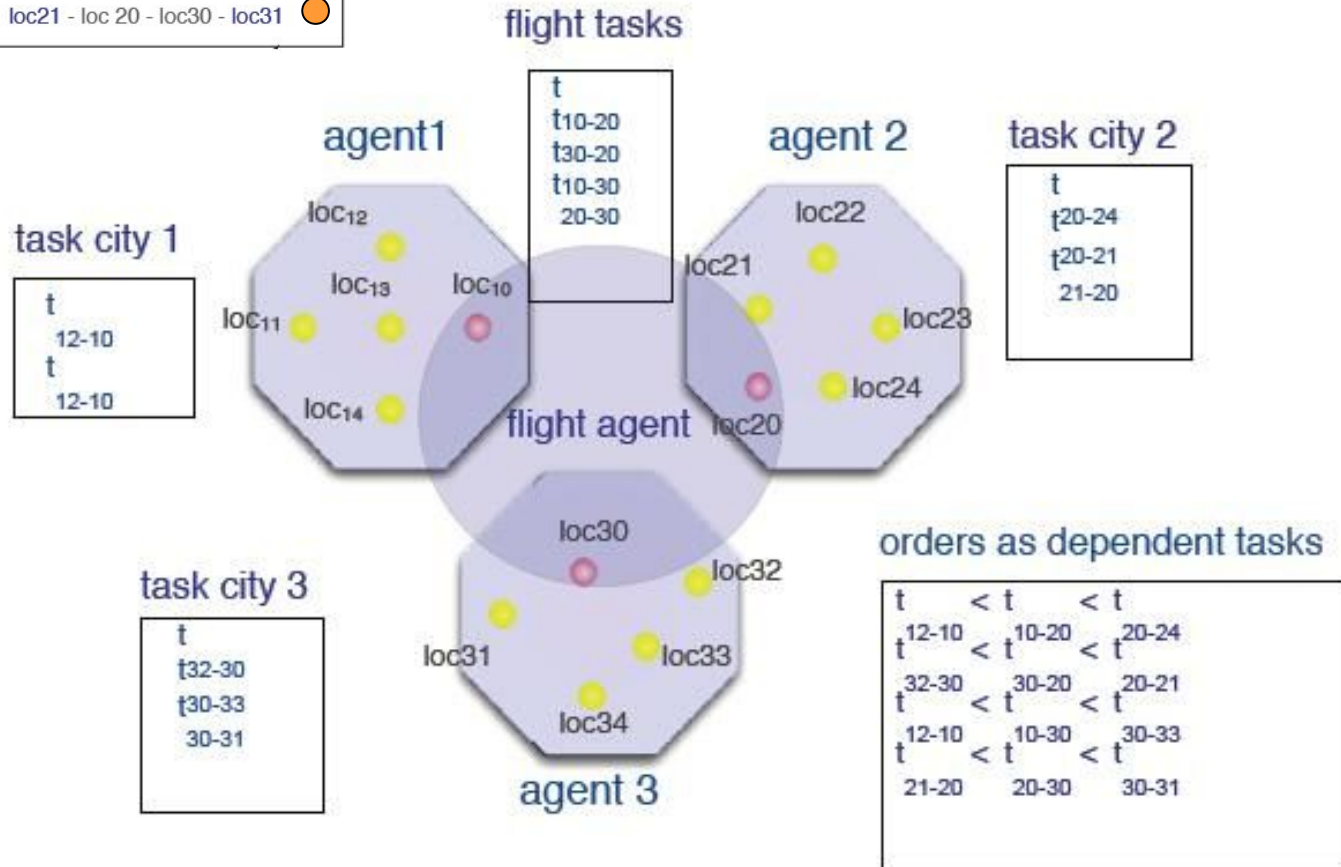
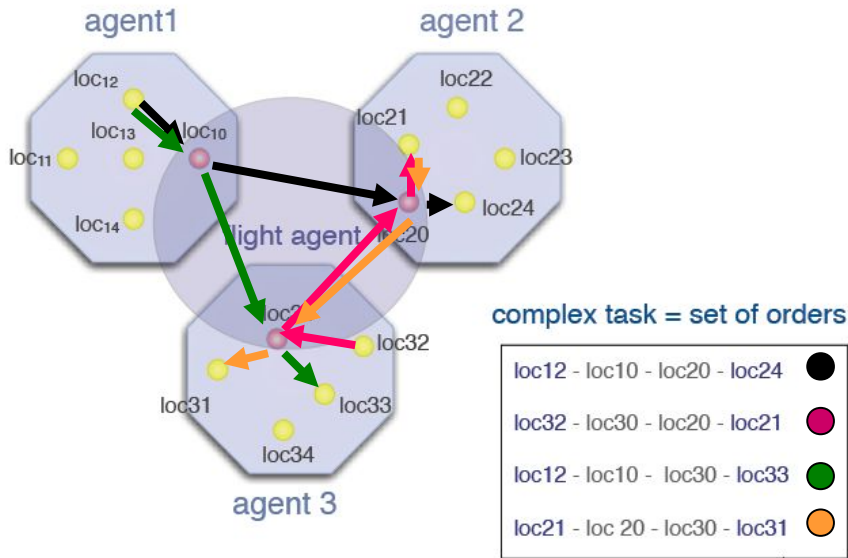
Koordinálás mechanizmusai

- **Implicit** koordinálási mechanizmusok
 - környezet definiálása/megváltoztatása, hogy a koordinálási probléma „magától” megoldódjon
 - pl. szociális konvenciók/ törvények
 - pl. szervezetek
 - pl. ágensek modellezése
 - pl. szabad piaci gazdaság („a láthatatlan kéz”)
- **Explicit** koordinálási mechanizmusok
 - ágensek explicit módon érvelnek/tárgyalnak, ki, mit, mikor?
 - pl. elkötelezettségek reprezentálása és csere
 - pl. elosztott tervekészítés
 - pl. elosztott ütemezés
- Valóság: hibrid esetek

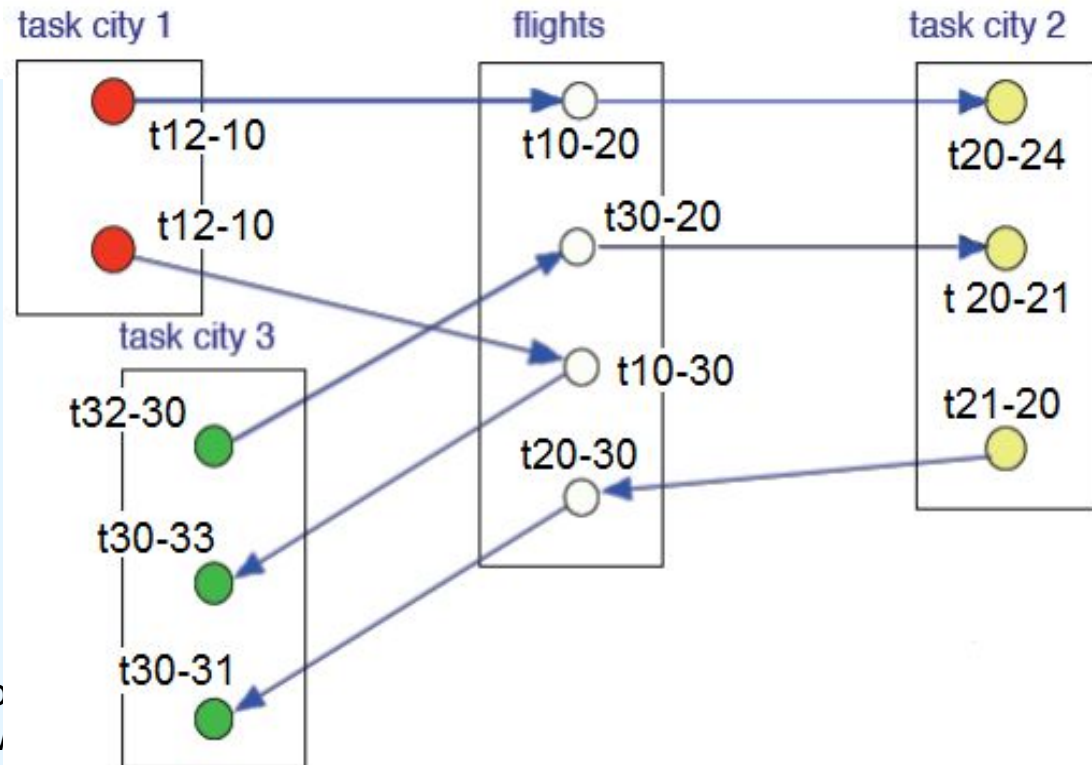
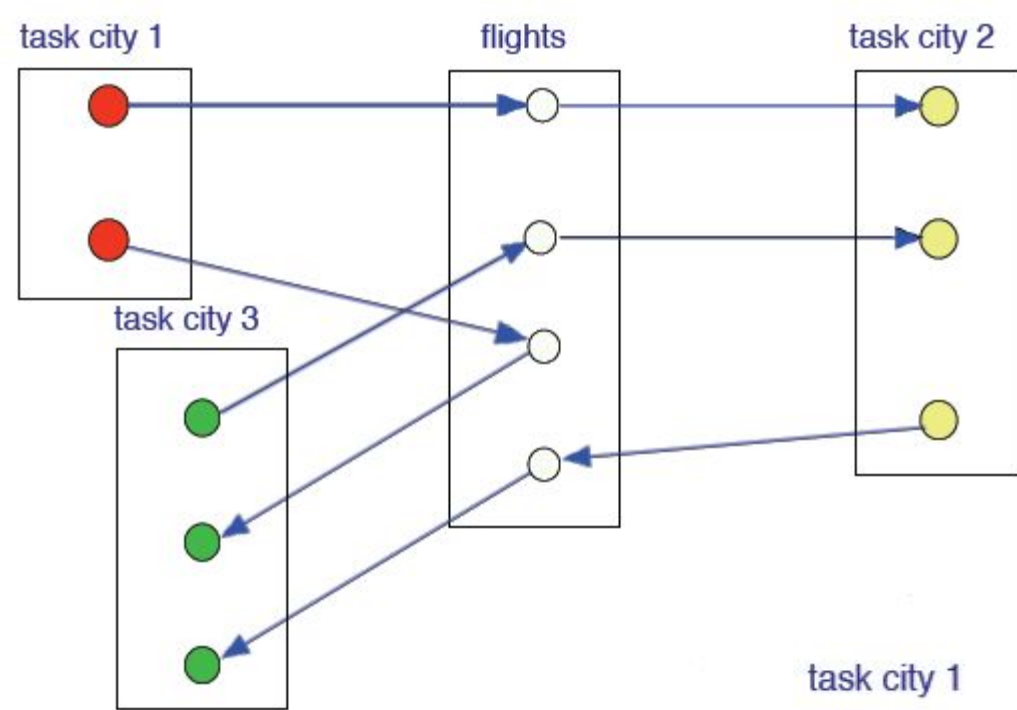
Több ágenses tervekészítés kritériumai

- Számítási költségek
- Kommunikációs költségek
 - üzenetek száma
 - adatmennyiség (sávszélesség)
 - megengedett/előírt késedelmek
- Rugalmasság (elkötelezettség)
 - mennyi szabadságot biztosítanak egymásnak az ágensek
 - idő
 - erőforrások
 - cselekvések megválasztása
- Robusztusság – a környezet befolyása, befolyásolása
- Terv kvalitása
- Skálázhatóság
 - ágensek száma?
 - probléma nagysága
 - kölcsönhatások

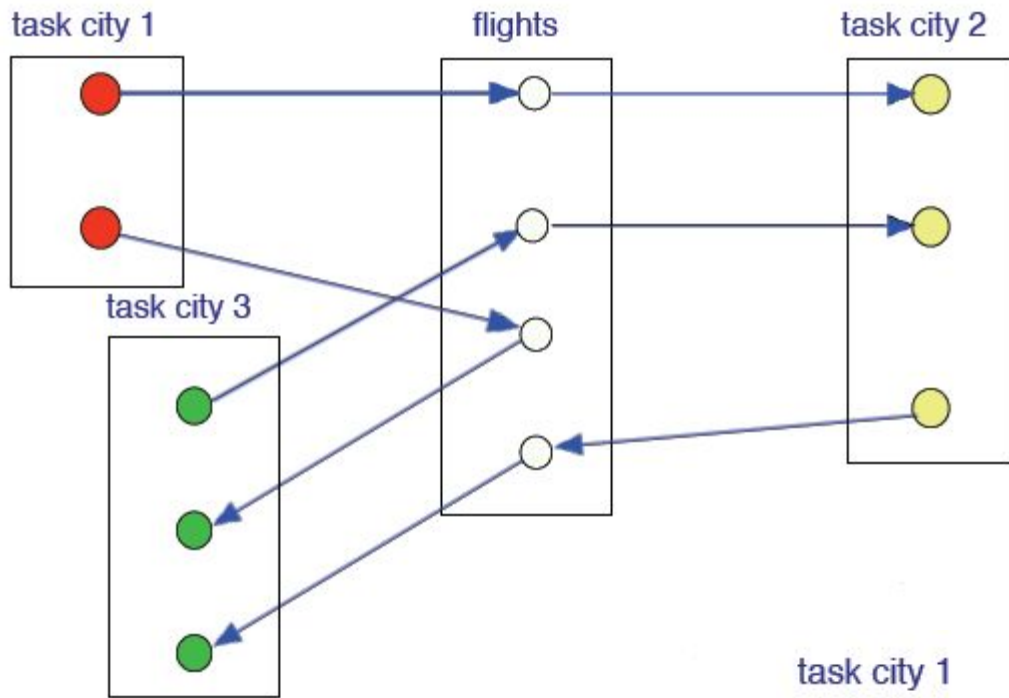
Koordinálás tervekészítés előtt



Gráf reprezentáció

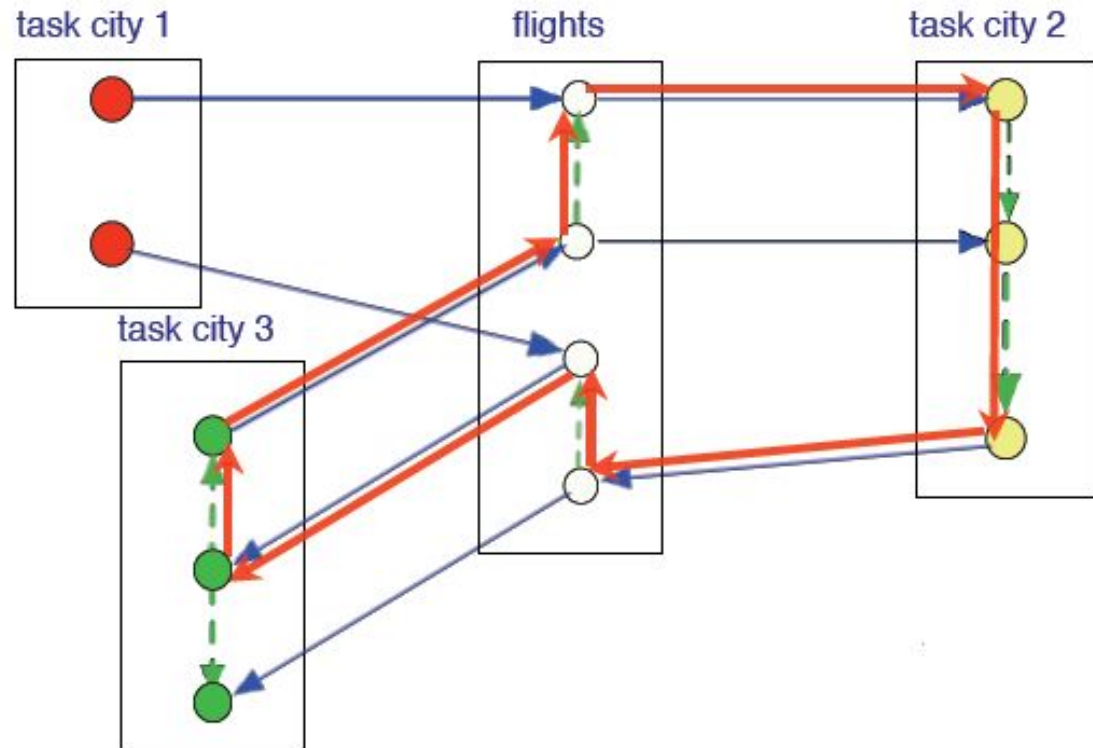


Gráf reprezentáció



Egyedi tervek nem koordinált megválasztása

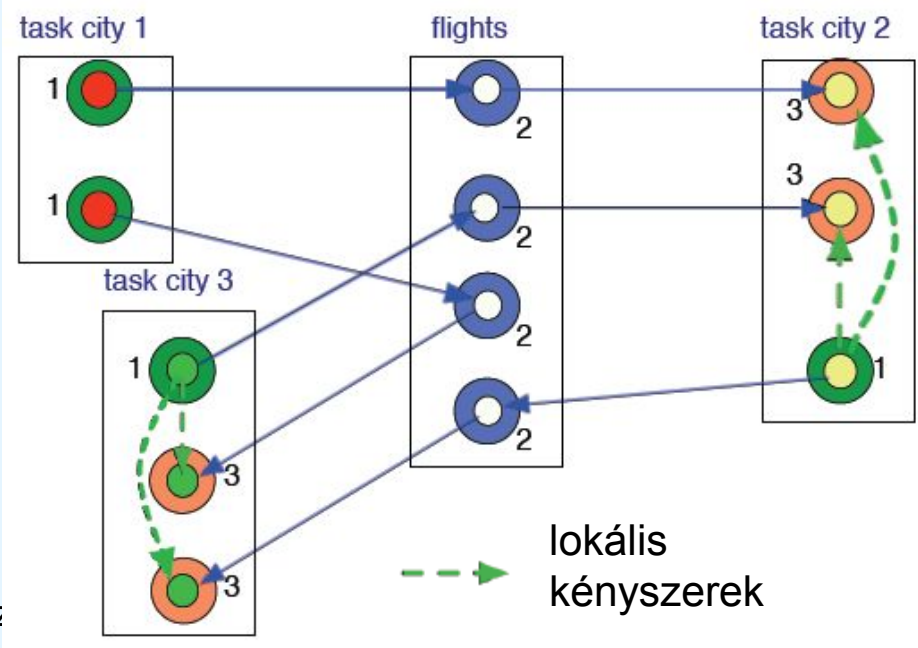
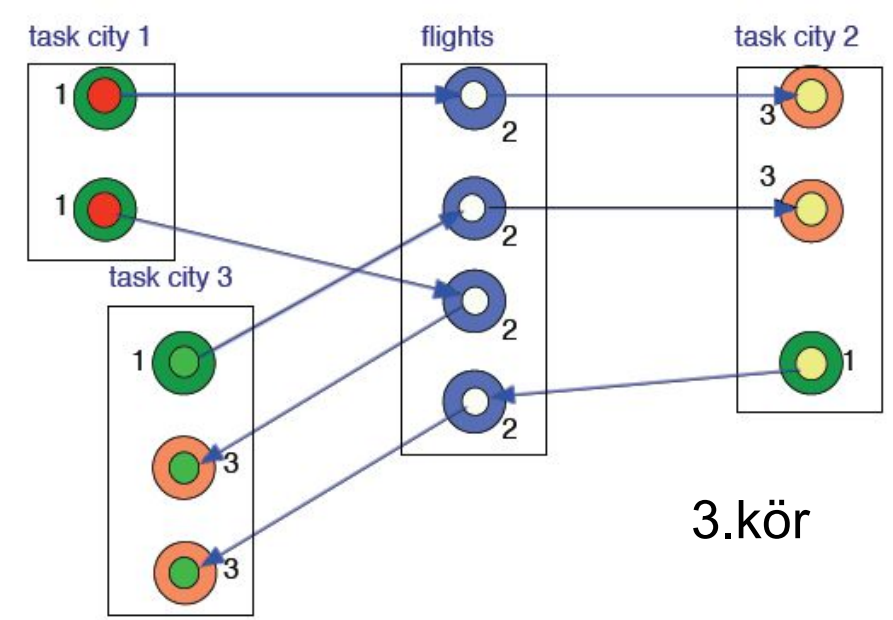
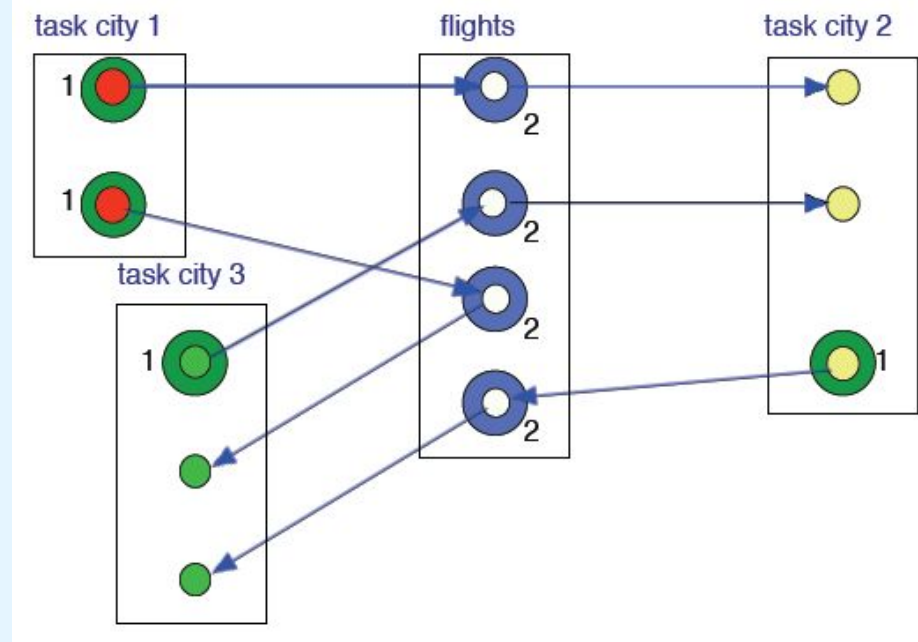
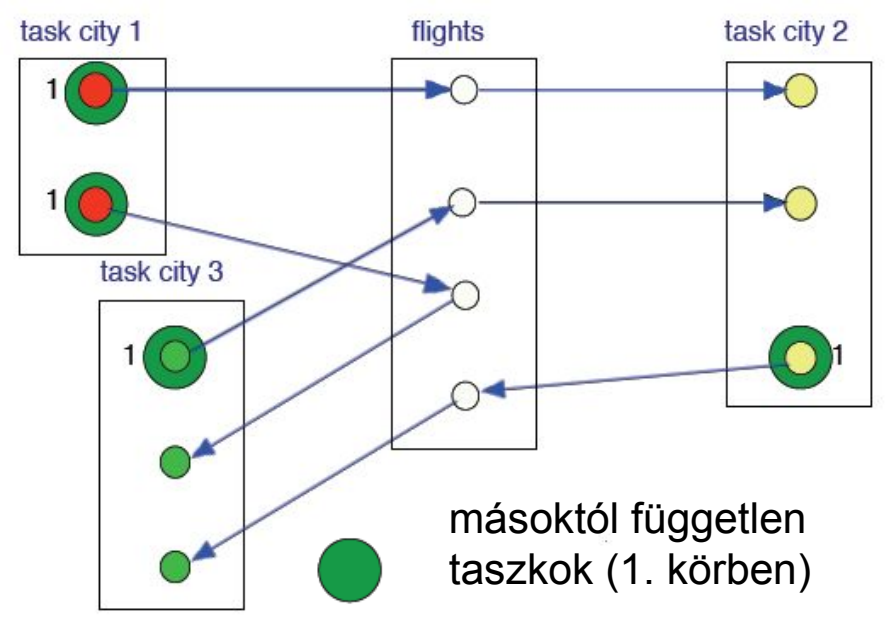
egyedi tervek rendben
együttes terv hm-hm?



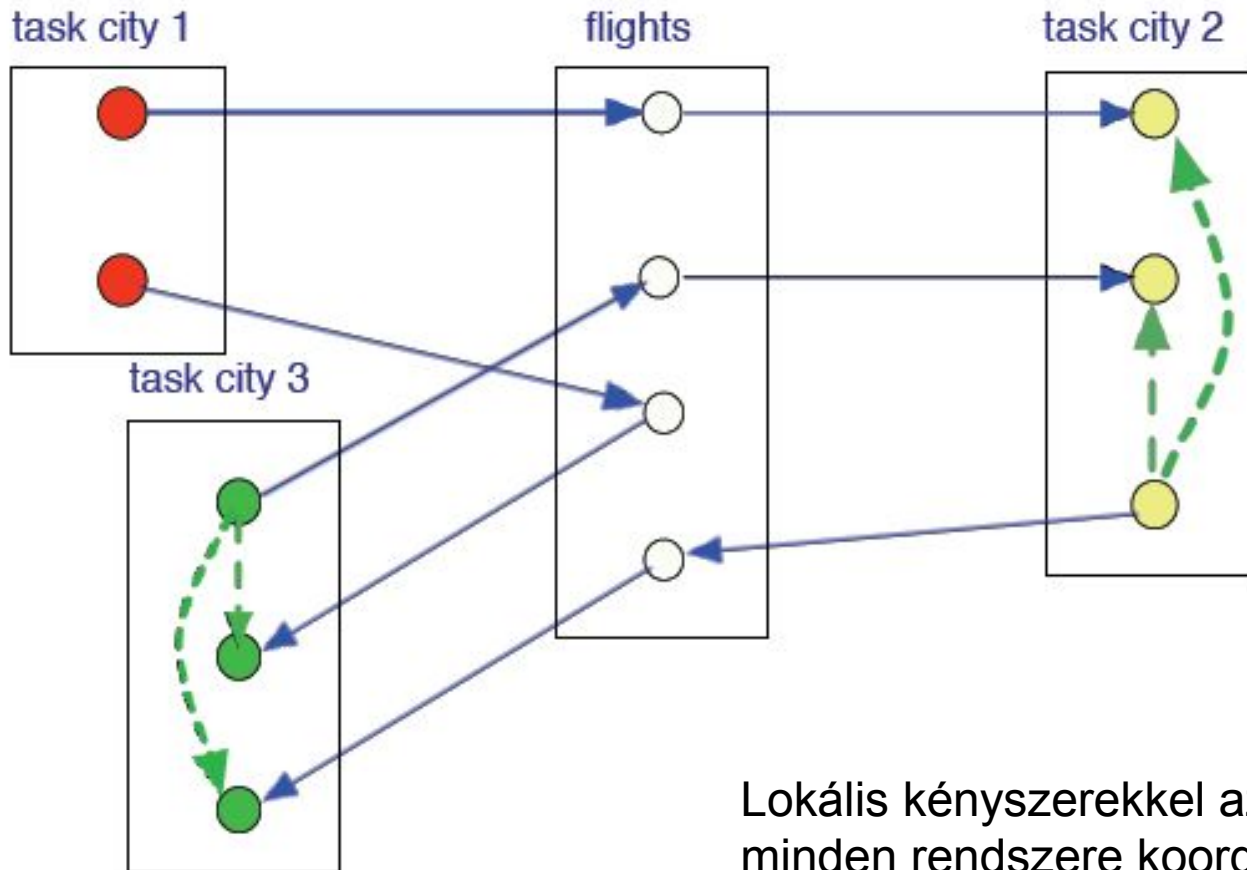
Elosztott particionálás

1.kör

2.kör



és
lész



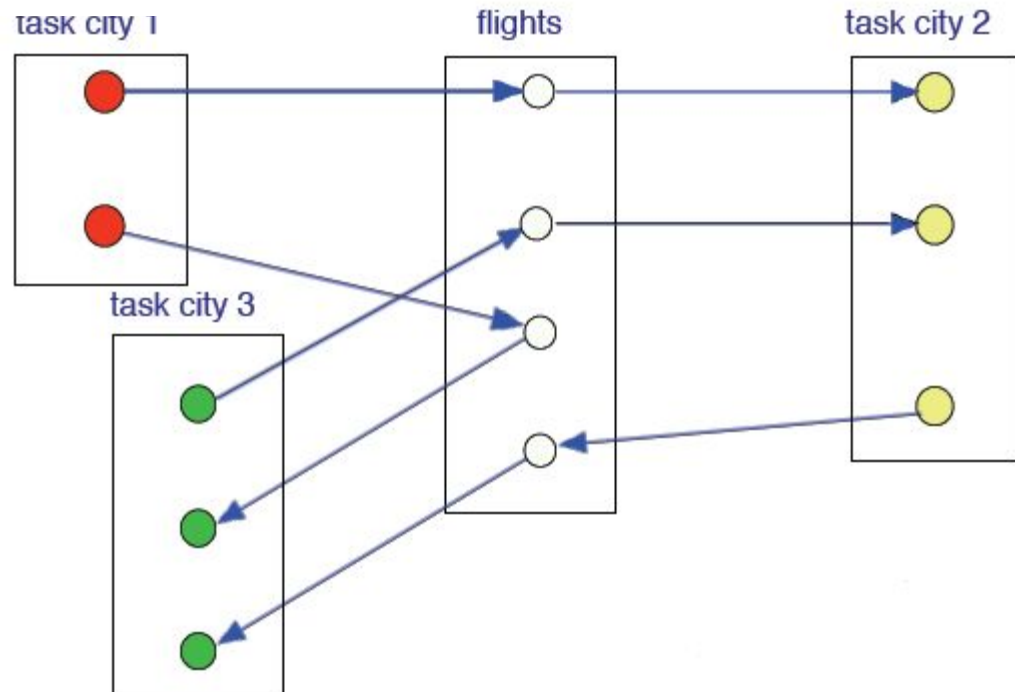
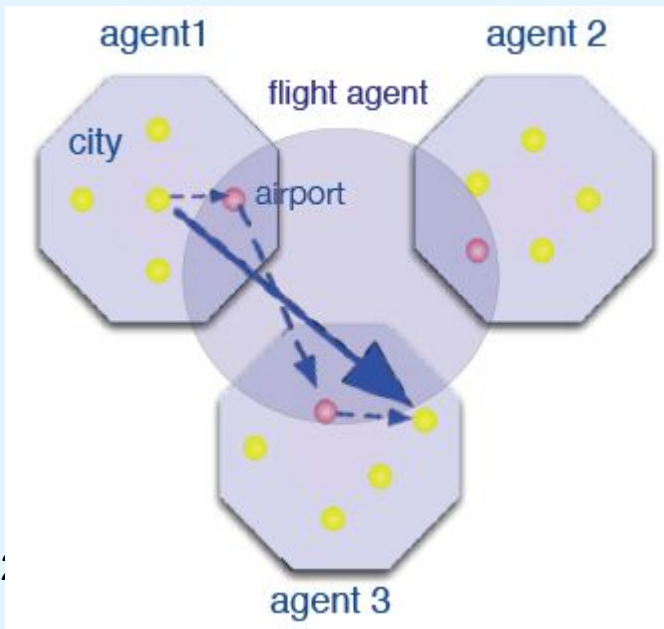
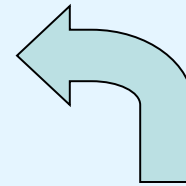
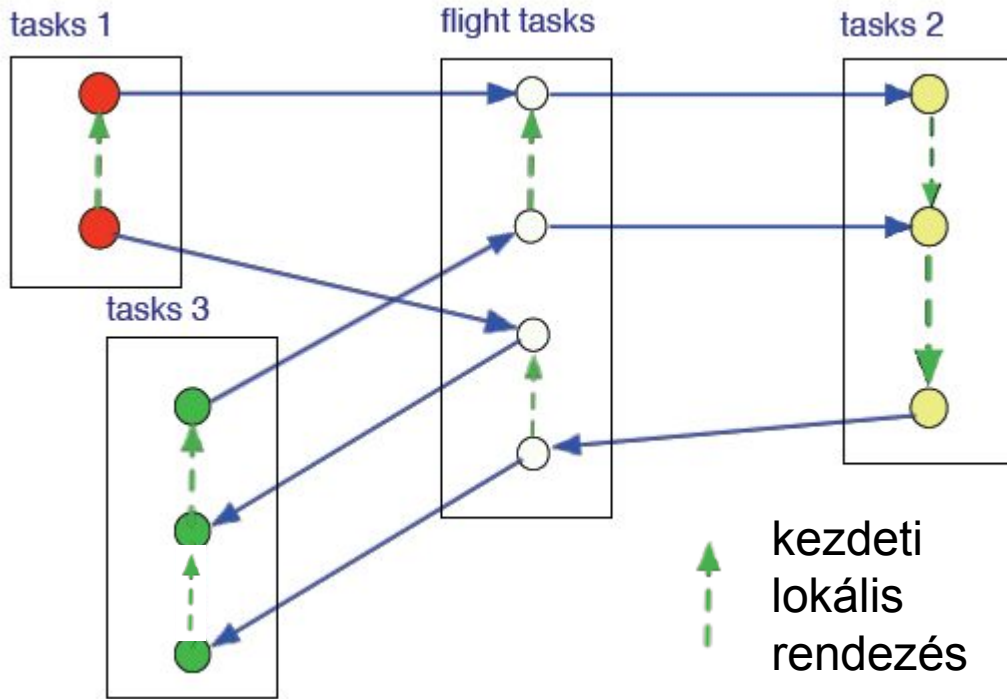
Lokális kényszerekkel az egyedi tervek minden rendszere koordinált közös terv!

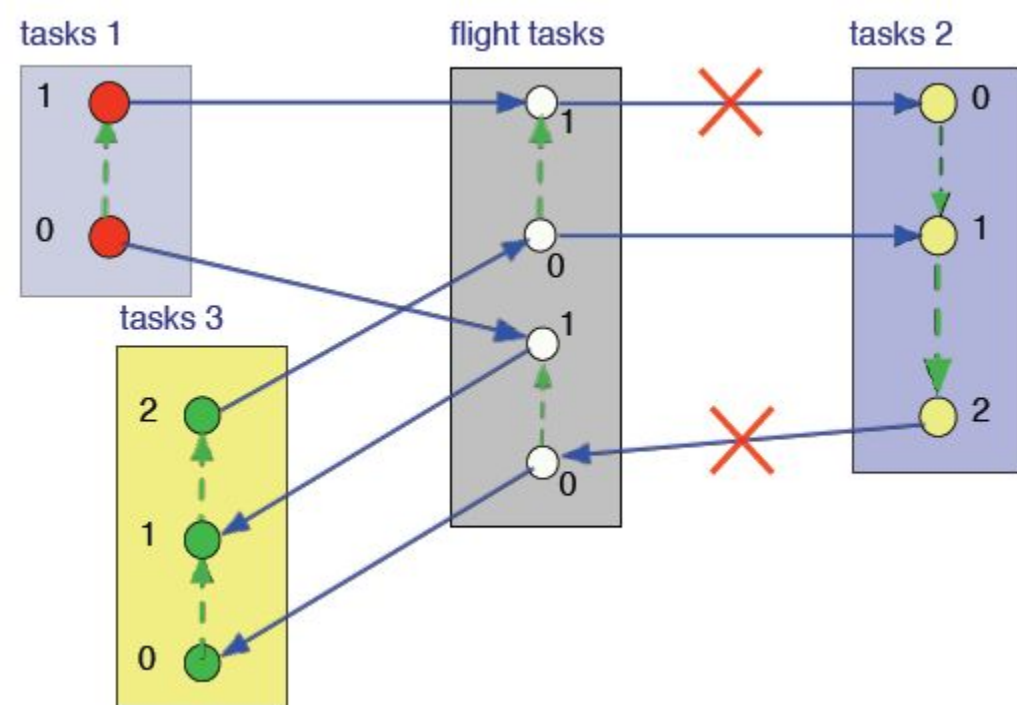
Egyszerű koordinációs protokoll:

Taxi: minden pre-flight + lokális közlekedés a post-flight feladat előtt

Repülő: nincs megkötés

Koordinálás tervkészítés közben

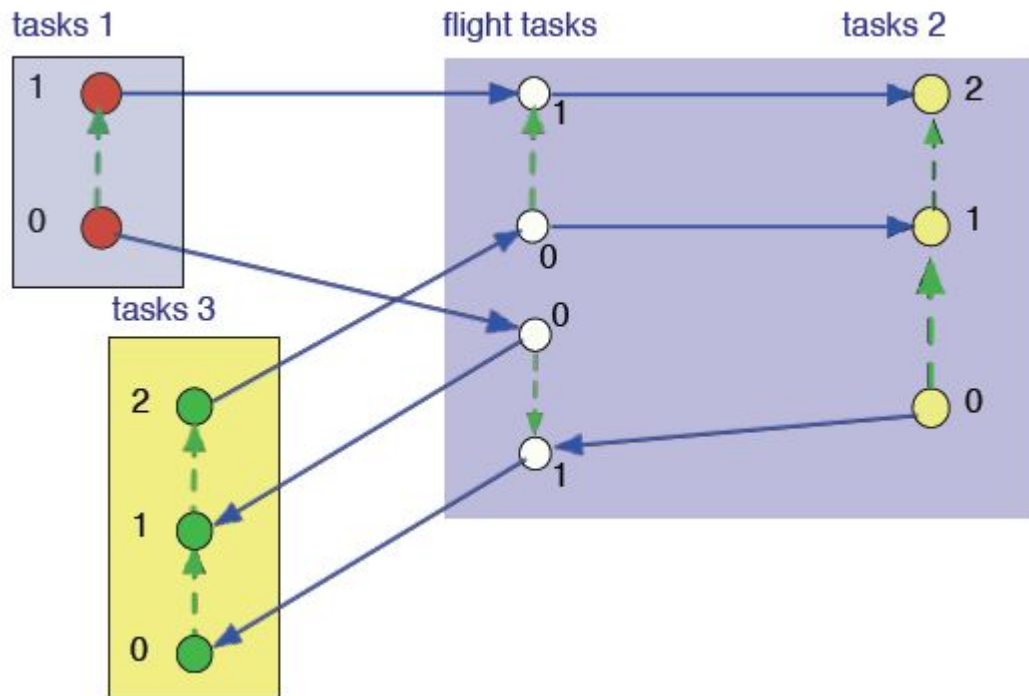


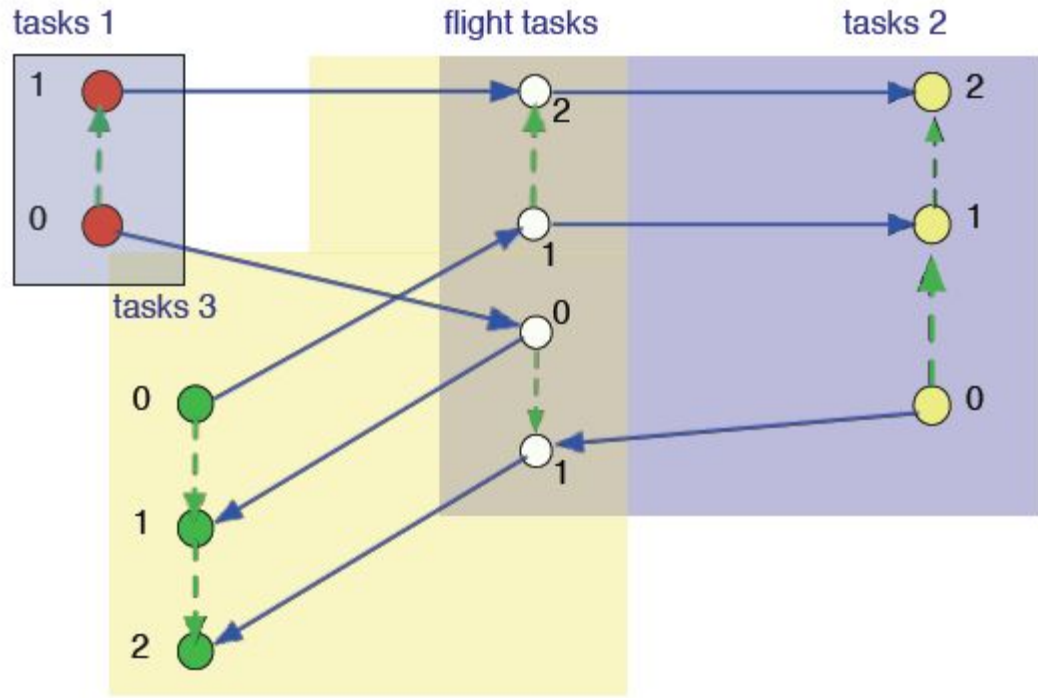
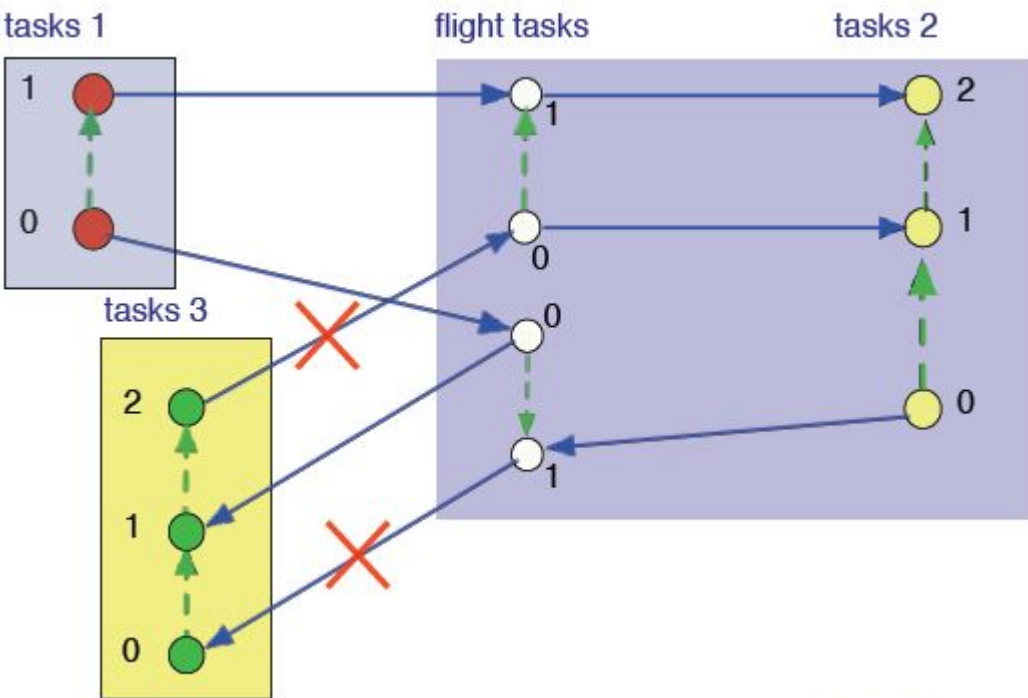


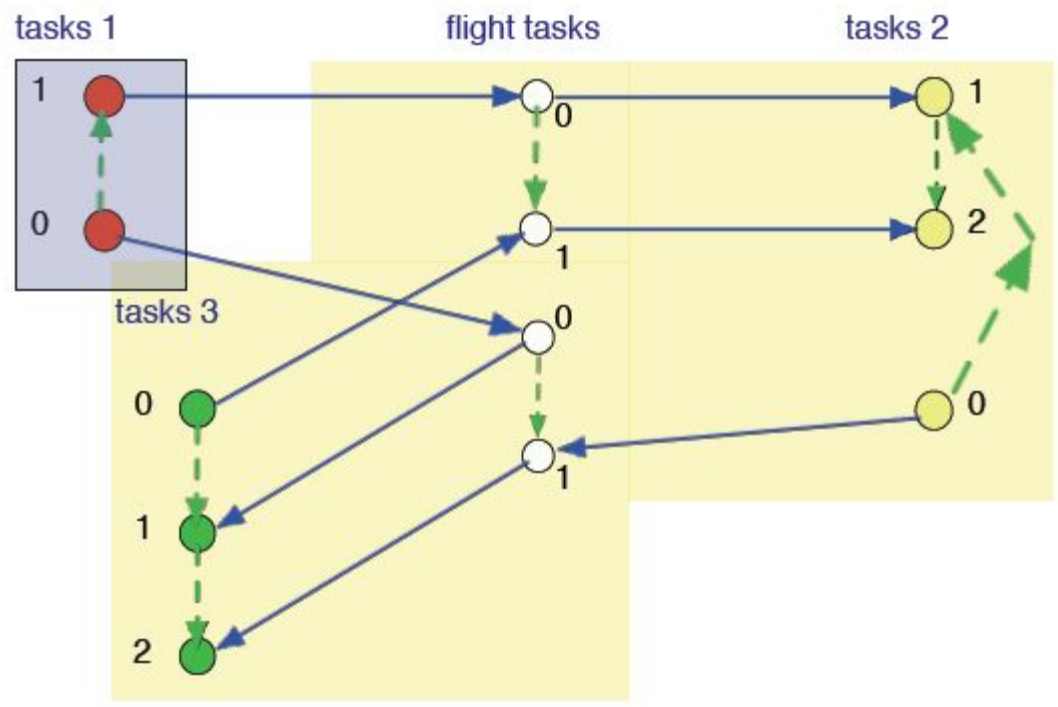
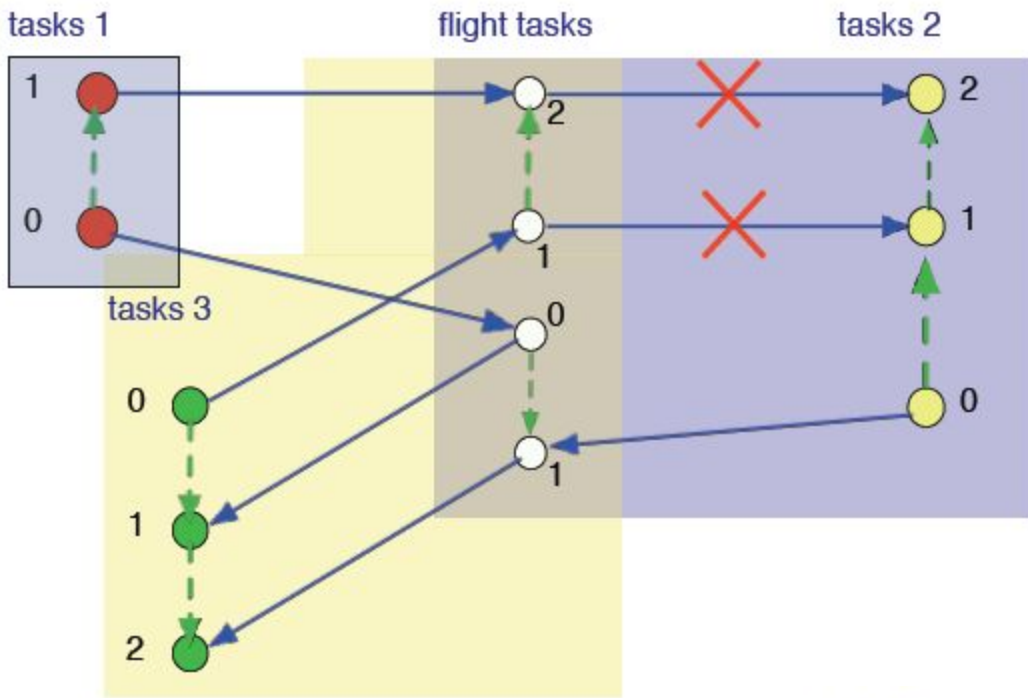
kezdeti ütemezések

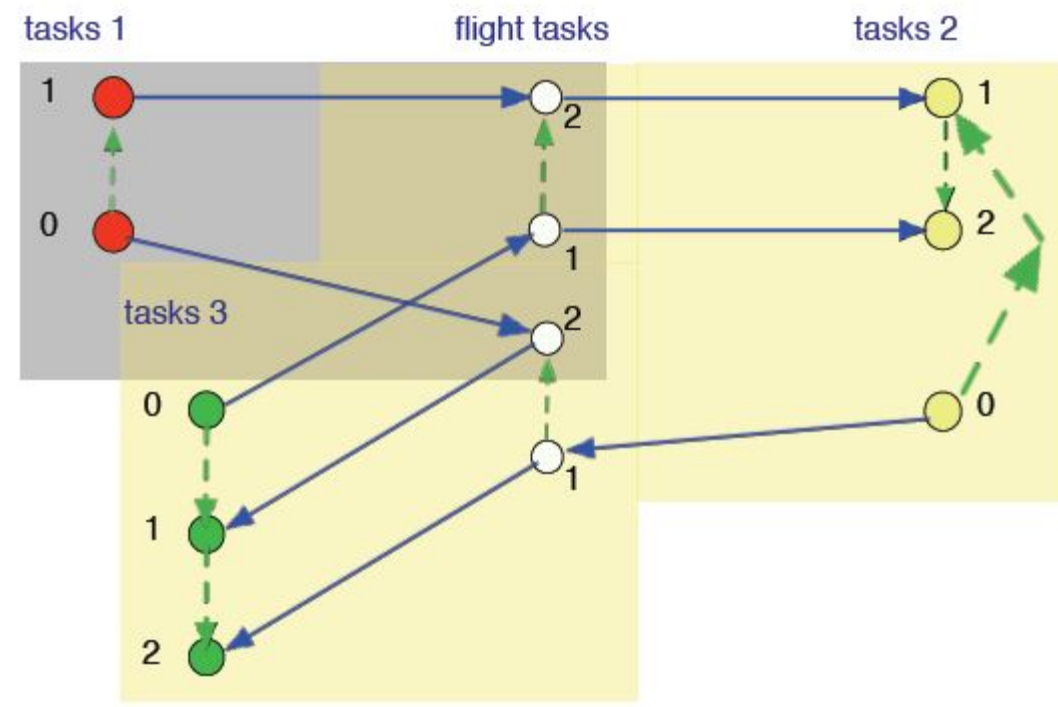
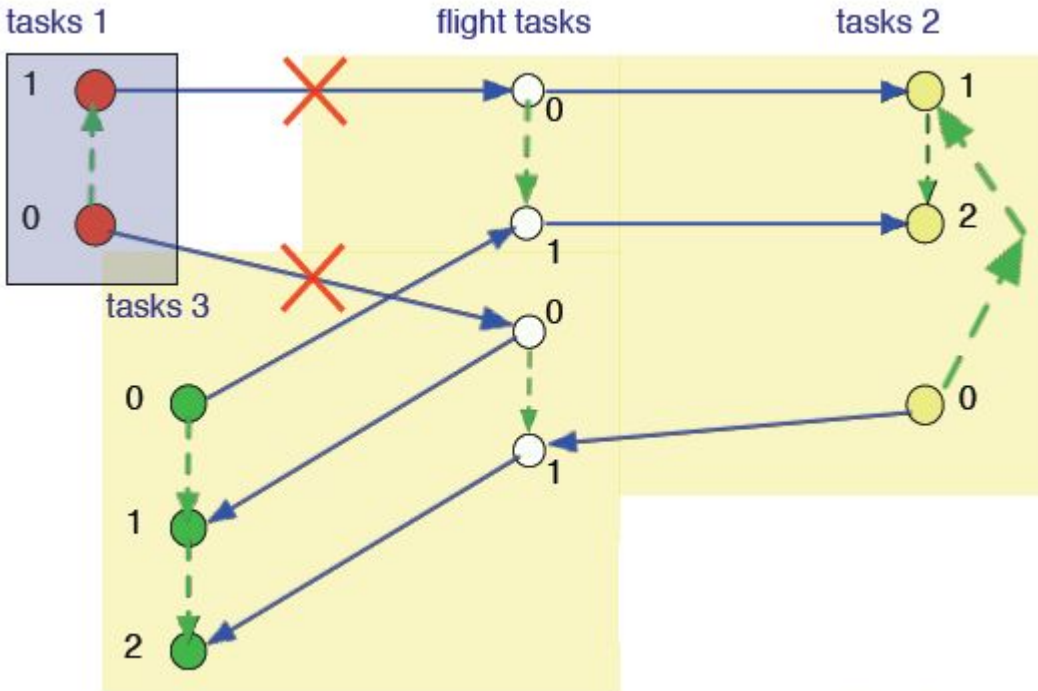
konfliktus:
repülő – ágens 2

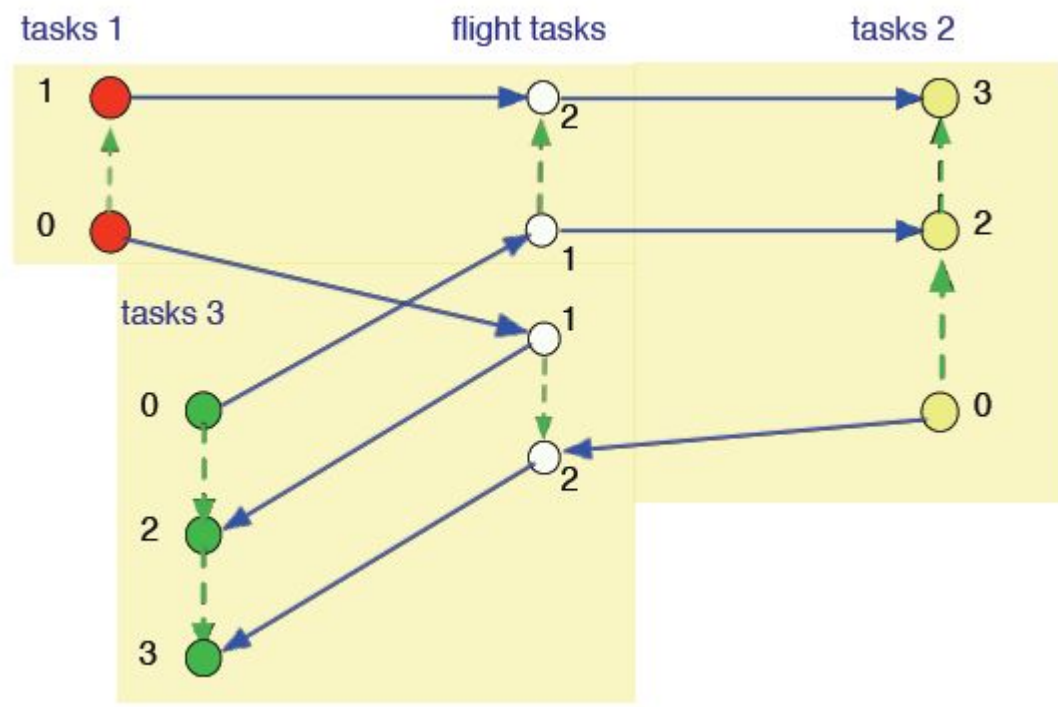
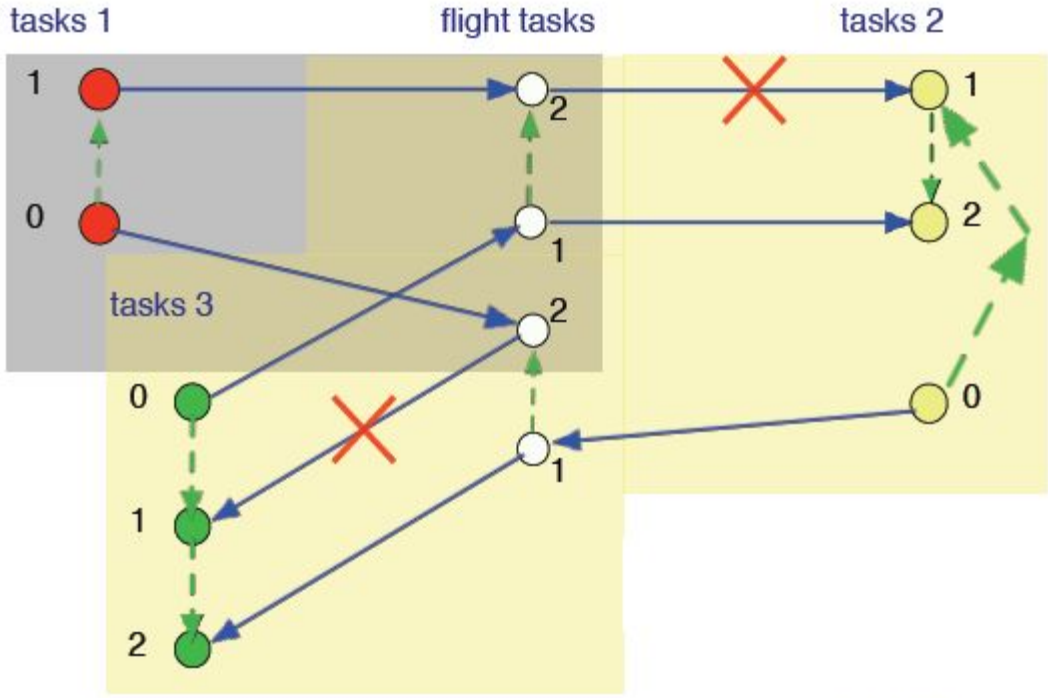
Konfliktus feloldás:
repülő – ágens 2
lokális kényszerek cseréje
memorizálás











....
 nincs több konfliktus
 minden terv lokálisan
 és globálisan kivitelezhető

Heurisztika tervekészítéshez: tervekészítési gráfok - Graphplan

Lehetséges cselekvések: Kezdeti állapot $\Rightarrow \Rightarrow \Rightarrow$ Kiegyenlítés

Kölcsönös kizárások – **mutual exclusion** - **mutex**

Cél predikátumok \notin Kiegyenlített állapot - **megoldás nincs**

Cél predikátumok \in Kiegyenlített állapot, cél predikátumok \neq mutex - **megoldás lehet**

A gráfból ki kell nyerni, ehhez gráf jó heurisztikát ad.

Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

Előfeltétel: Van(Süti)

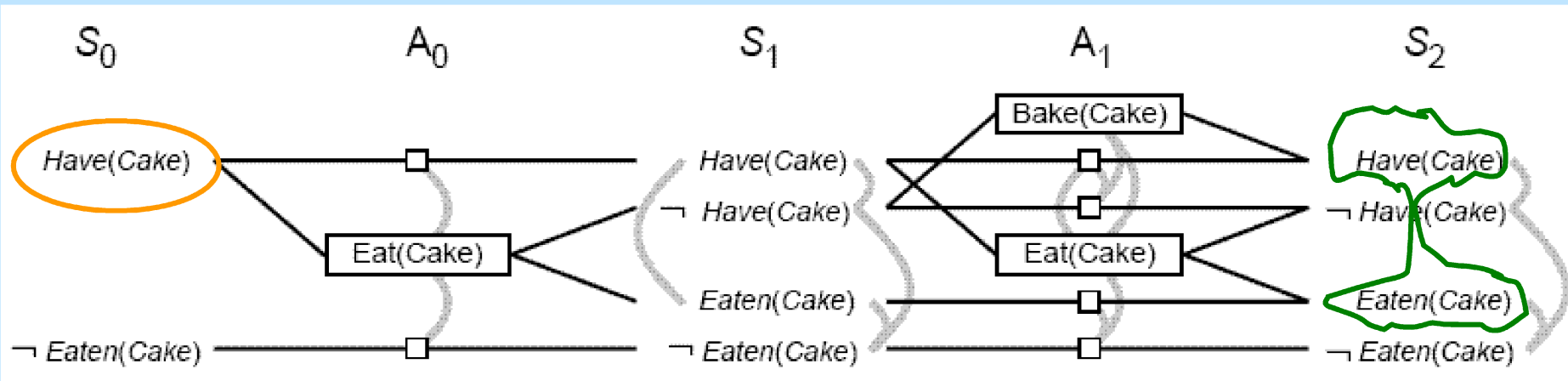
Követk: \neg Van(Süti) \wedge

Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

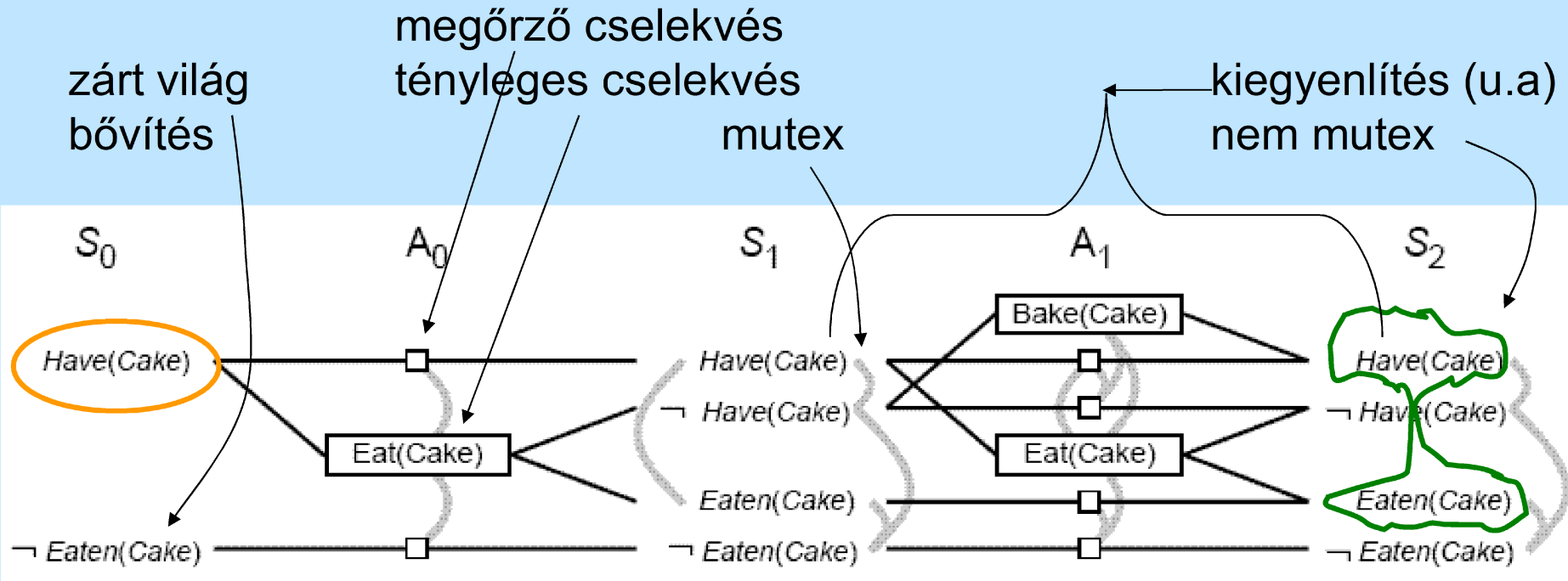
Előfeltétel: Van(Süti)

Követk: \neg Van(Süti) \wedge
Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Mutex-ek

Inkonzisztens hatások

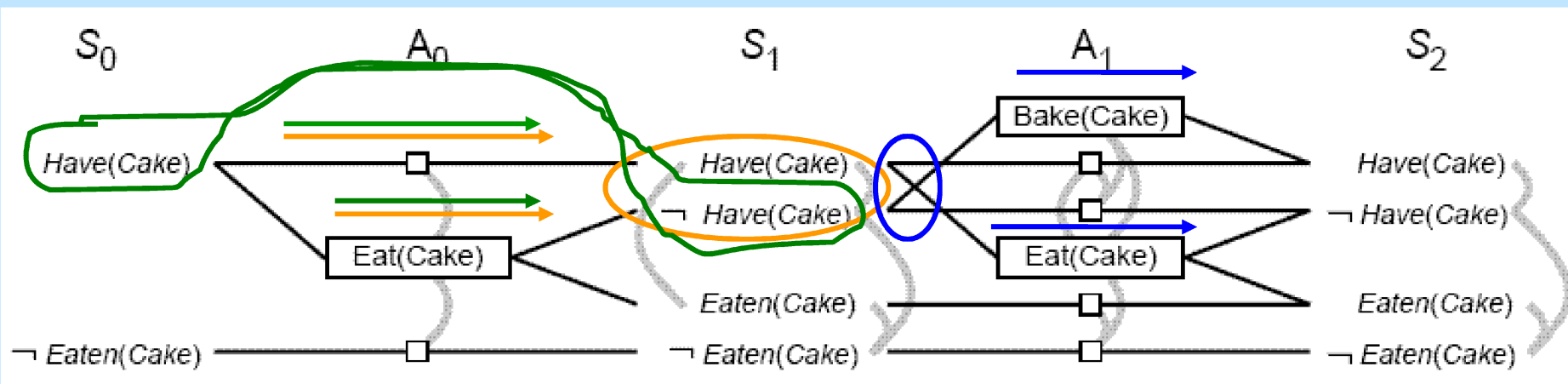
$$\text{Követk}(Cs1) = \neg \text{Követk}(Cs2) \quad \blacklozenge$$

Interferencia

$$\text{Követk}(Cs1) = \neg \text{Előfelt}(Cs2) \quad \blacklozenge$$

Versenyhelyzet

$$\text{Előfelt}(Cs1) = \neg \text{Előfelt}(Cs2) \quad \blacklozenge$$



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

Előfeltétel: Van(Süti)

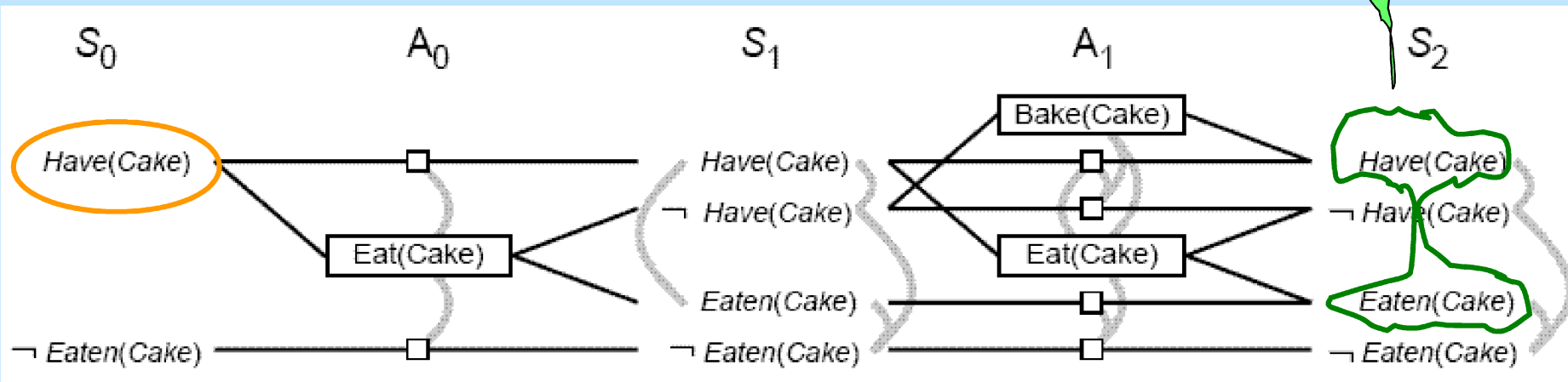
Követk: \neg Van(Süti) \wedge
Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)

Kiegyenlített állapotban **igaz** a cél és **nem mutex**
= a terv létezik, csak valahogy ki kell nyerni!



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

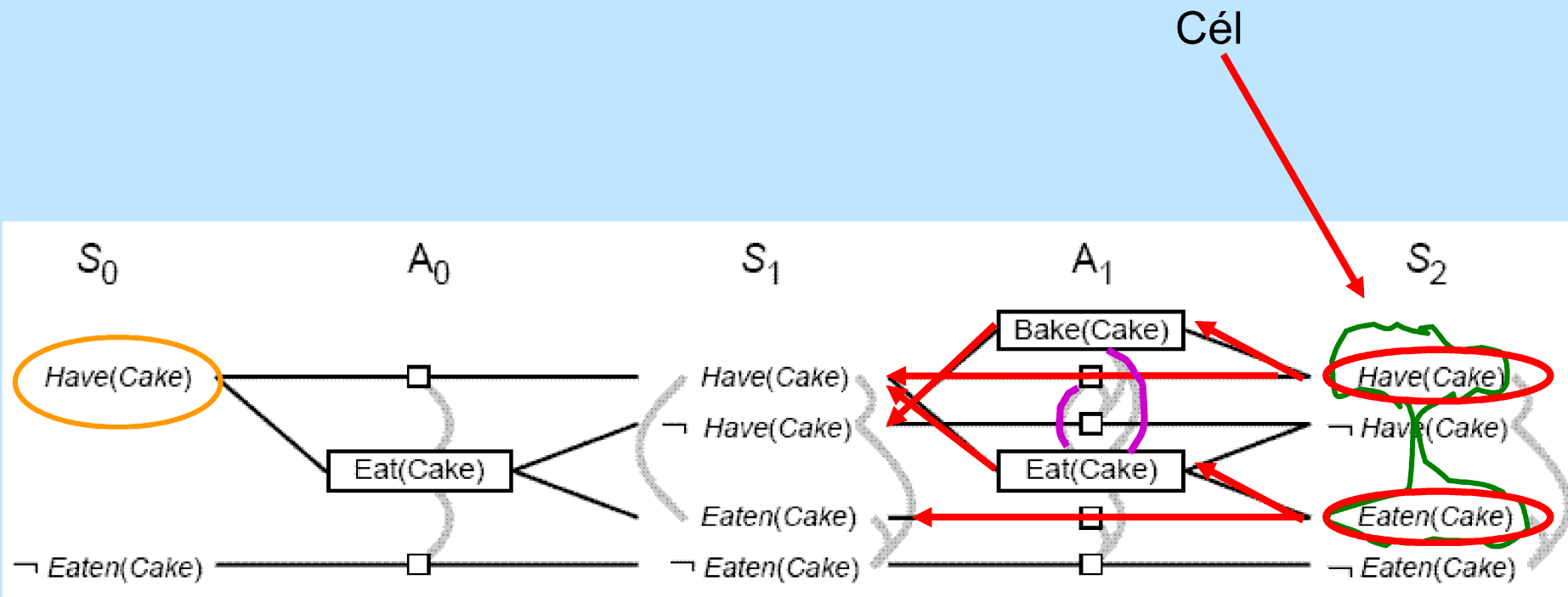
Előfeltétel: Van(Süti)

Követk: \neg Van(Süti) \wedge
Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

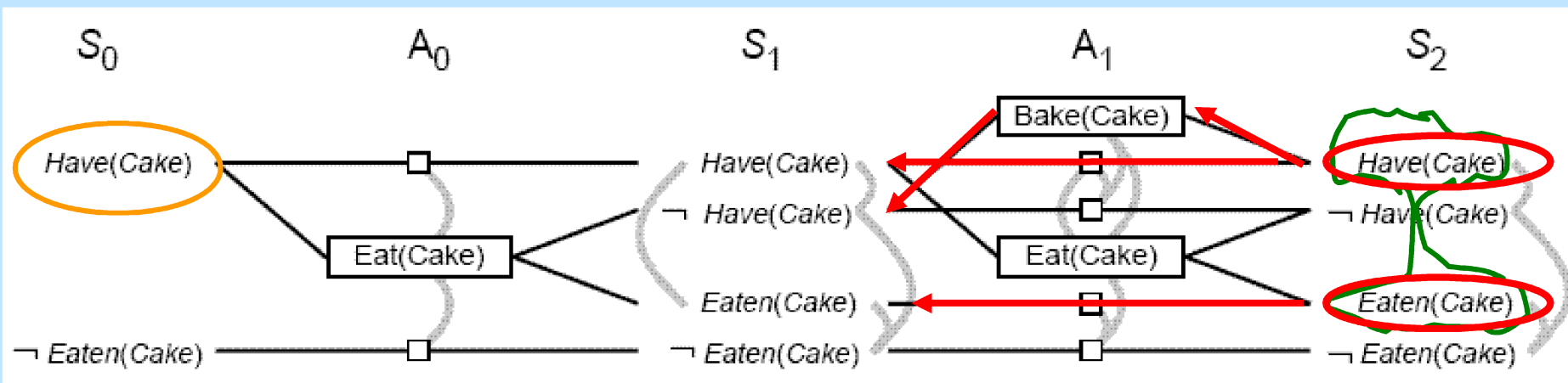
Előfeltétel: Van(Süti)

Követk: \neg Van(Süti) \wedge
Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

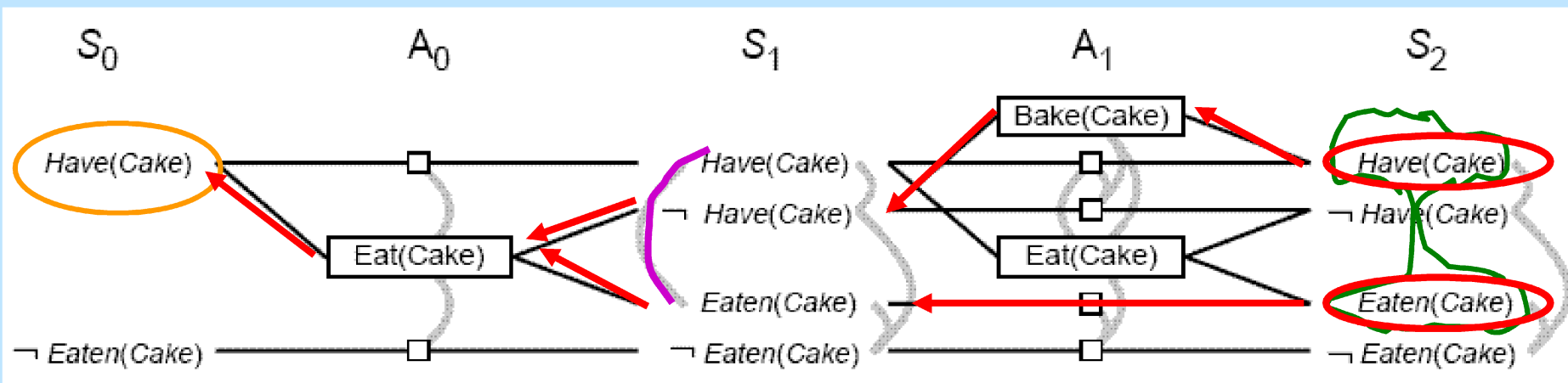
Előfeltétel: Van(Süti)

Követk: \neg Van(Süti) \wedge
Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

Előfeltétel: Van(Süti)

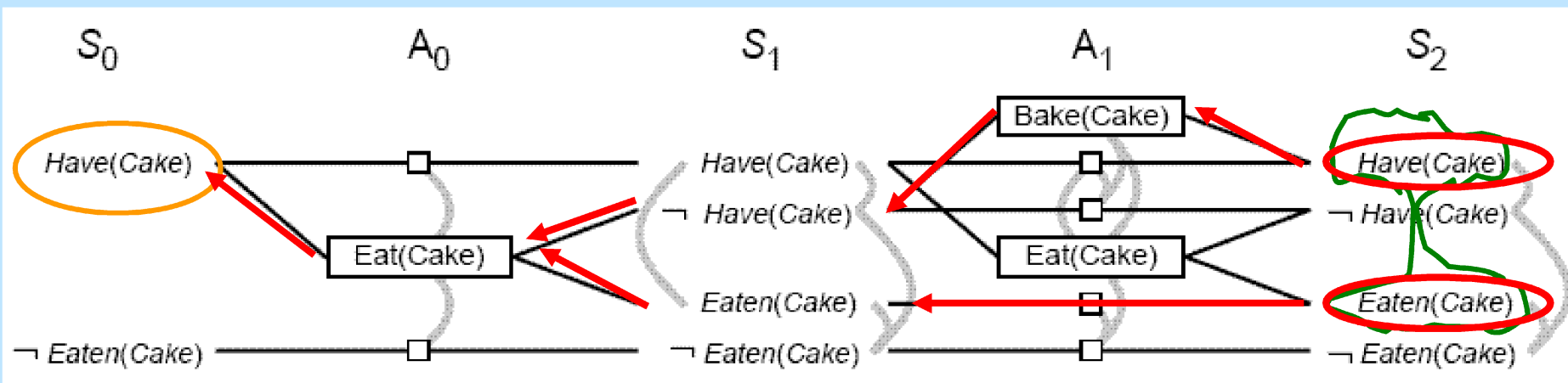
Követk: \neg Van(Süti) \wedge

Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

Előfeltétel: Van(Süti)

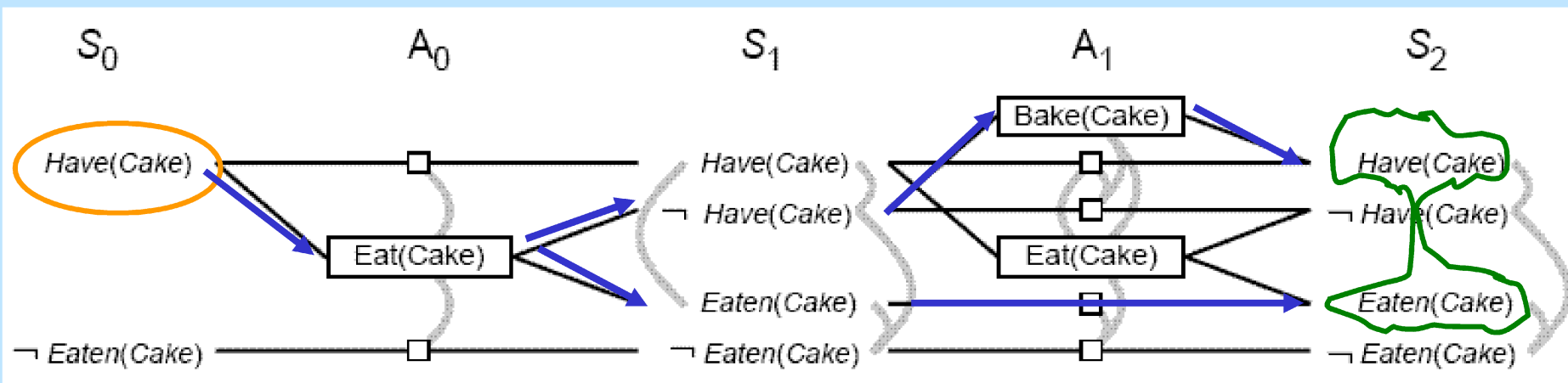
Követk: \neg Van(Süti) \wedge

Megevett(Süti)

Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Op(Cselekvés: **Start**

Előfeltétel: -

Követk: **Van(Süti)**

Op(Cselekvés: **Cél**

Előfeltétel: **Van(Süti) \wedge
Megevett(Süti)**

Op(Cselekvés: **Eszik(Süti)**

Előfeltétel: Van(Süti)

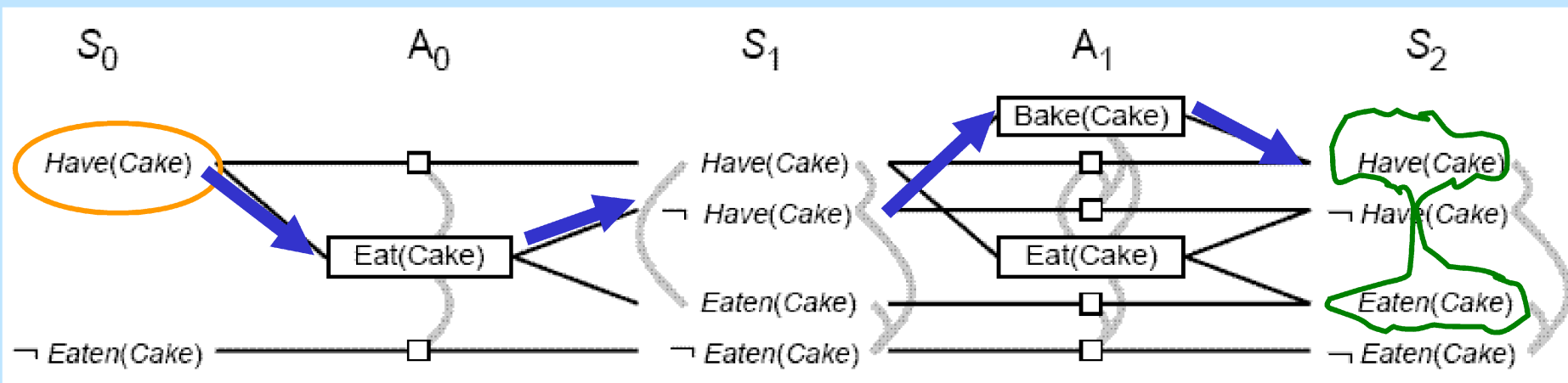
Követk: \neg Van(Süti) \wedge

Megevett(Süti)

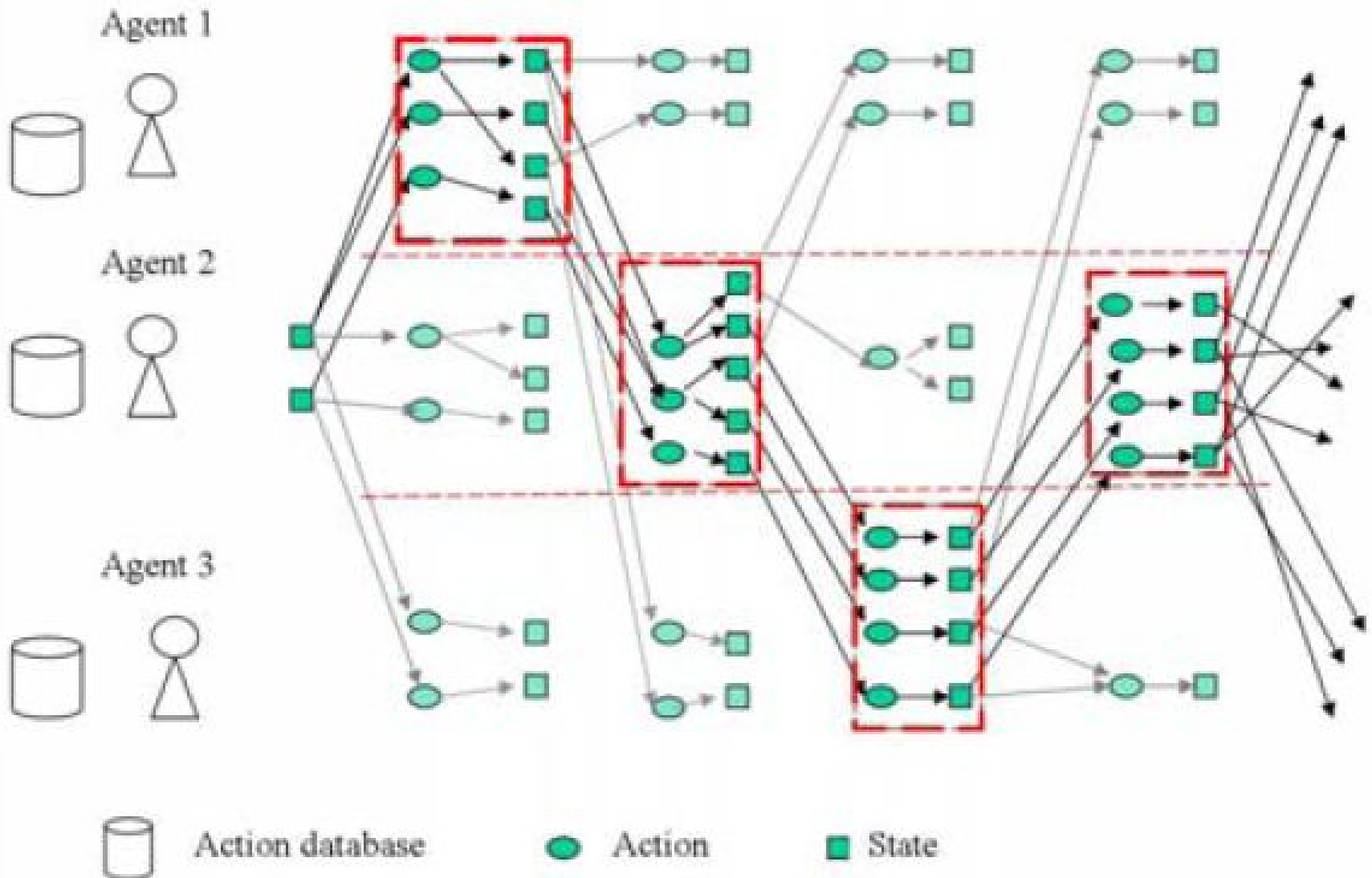
Op(Cselekvés: **Süt(Süti)**

Előfeltétel: \neg Van(Süti)

Követk: Van(Süti)



Elosztott Graphplan alapú tervekészítés



Elosztott Graphplan alapú tervekészítés

Algoritmus:

Gráfrészlet szerkesztése

1' szint szerkesztése kezdeti állapotokból

cselekvésszint szerkesztése az előbbi állapotszint alapján (X)

állapotszint szerkesztése az előbbi cselekvésszint alapján

aktuális állapotszint meghirdetése

üzenetek vétele után ellenőrzés: reach-all-goals?

if ezen a szinten minden célt sikerült elérni **then**

 search-for-valid-plan

else

 decide-next-state-level

 goto (X)

end if

Decide-next-state-level:

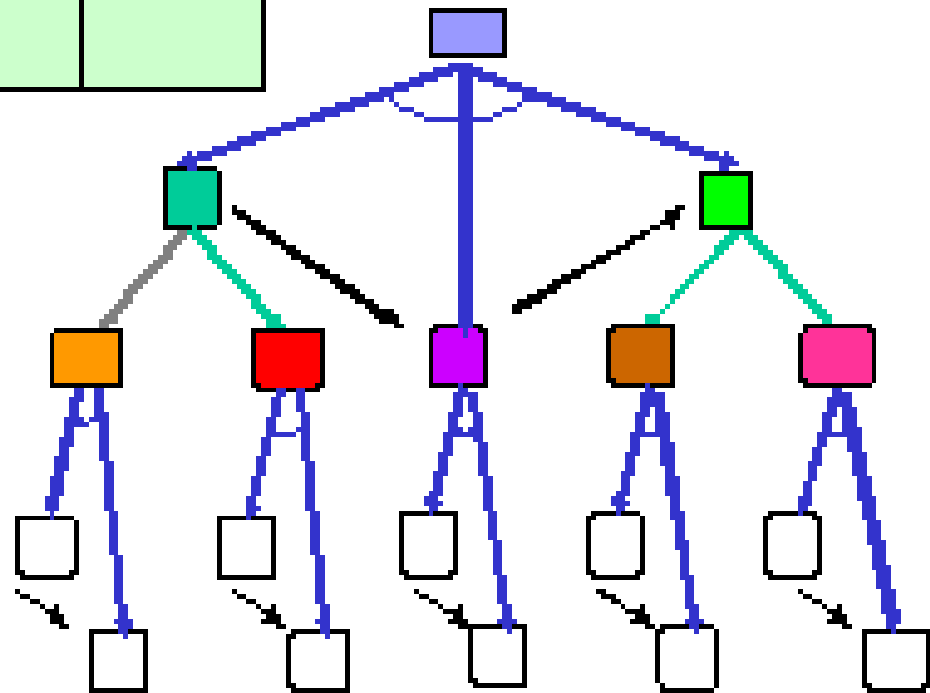
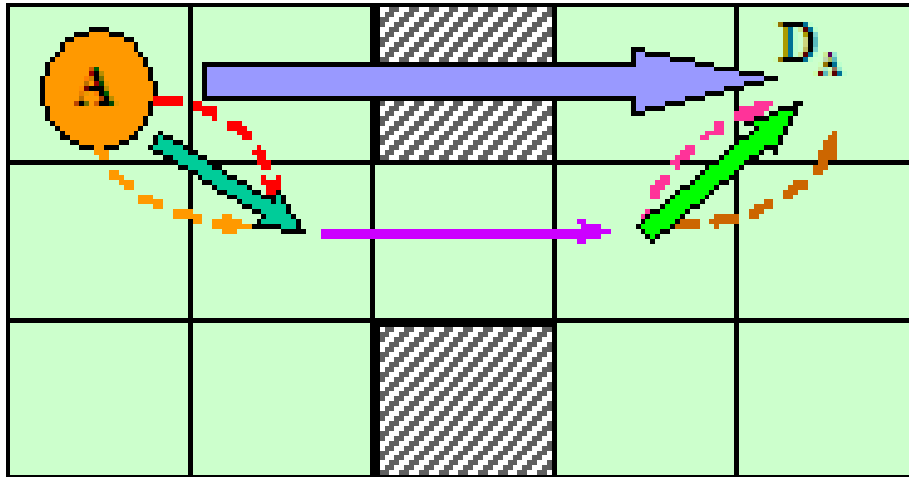
Max branching stratégia: állapotszint max. számú állapottal

Merge all stratégia: minden ágens utolsó állapota fuzionálva

 Max branching stratégia – kevesebb komm. overhead

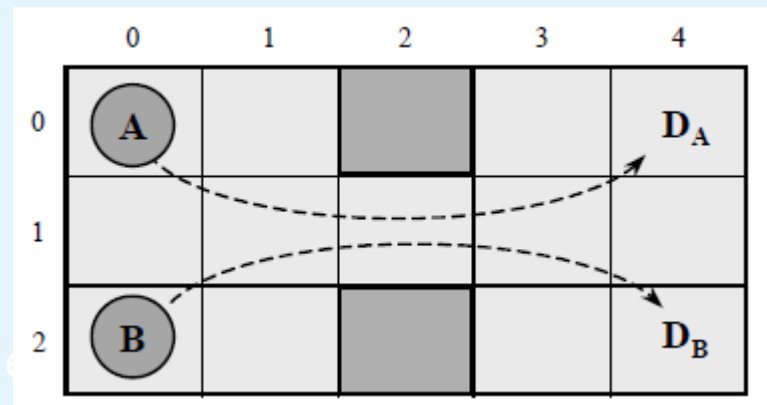
 Merge all stratégia – esély, hogy korábban éri el a tervet

Hierarchikus koordináció és tervekészítés

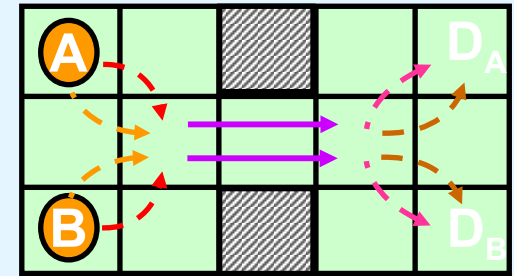
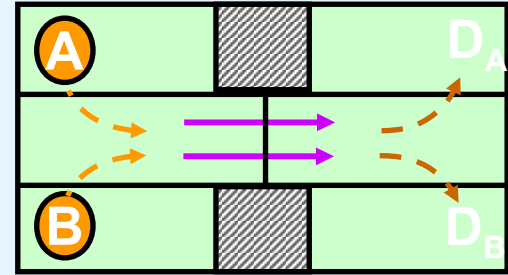
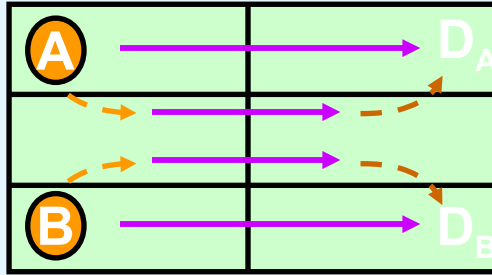
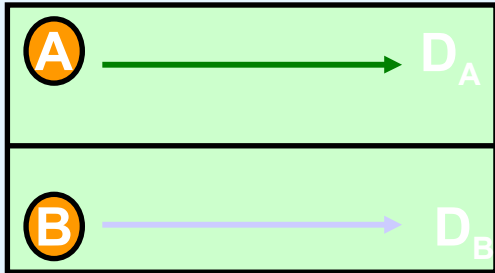


Hierarchikus koordináció és tervkészítés

1. Aktuális absztrakciós szint = a legabsztraktabb
2. Adott szinten ágensek kicserélik a terveiket és a céljaikat
3. Konfliktus nélküli tervek elhagyása. Ha nem marad semmi, készen vagyunk. Ha ezen a szinten foglalkozni kell konfliktusokkal, goto 5.
4. Szint állítása lejjebbre, tervek finomítása. Goto 2.
5. Feloldás:
 - (i) ágensek totális rendezése,
 - (ii) aktuális top ágens a többieknek elküldi a terveit,
 - (iii) alacsonyabb ágensek módosítják a terveiket konfliktus elkerülés érdekében,
 - (iv) a következő alacsonyabb ágens top ágens lesz.

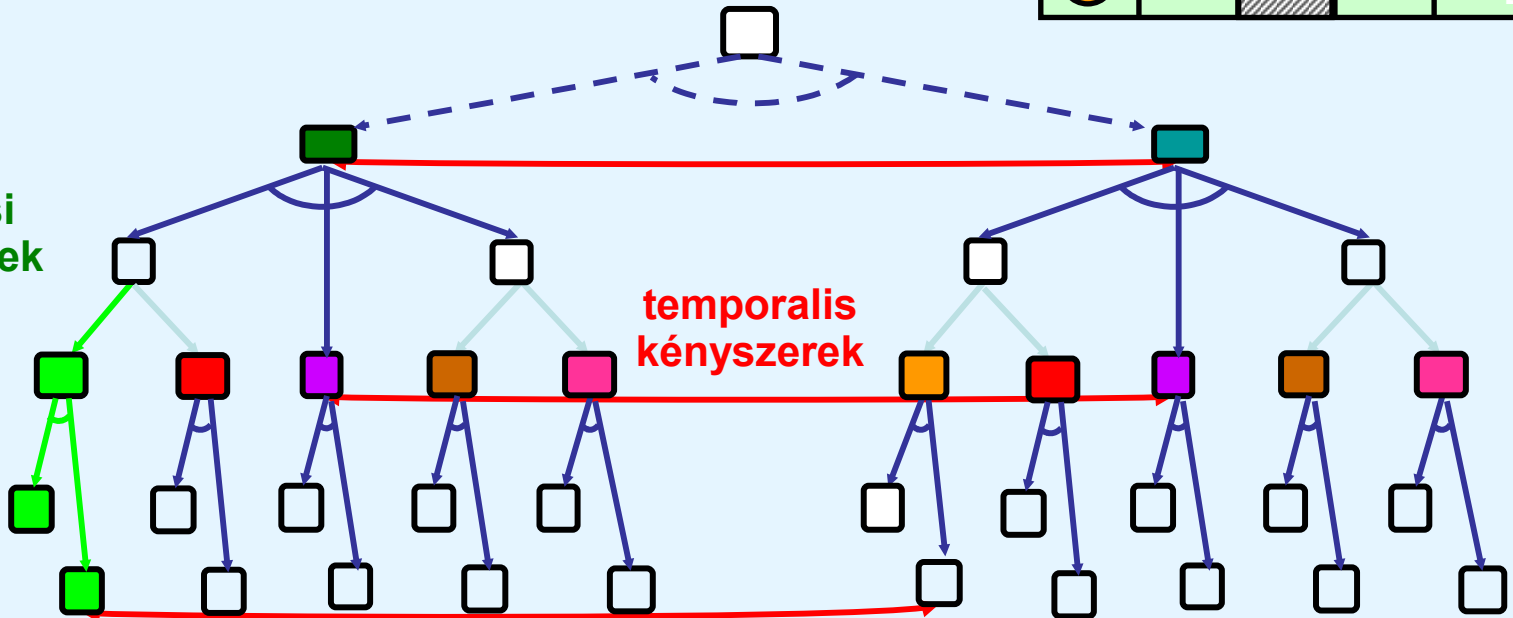


Hierarchikus koordináció és tervekészítés



választási
kényszerek

temporalis
kényszerek



Hierarchikus koordináció és tervekészítés

Ágens A – globális terve: p Ágens B- globális terve: q

p terv összesített (külső) előfeltételei (summary external **pre**conditions)

-//- -//- utófeltételei (summary external **post**conditions)

-//- -//- közbenső feltételei (-//- -//- **in**conditions)

→ p_{sum} ua. → q_{sum}

A → **üzenet** (p_{sum}) → **B**

CanAnyWay(időreláció, p_{sum} , q_{sum}), MightSomeWay(időreláció, p_{sum} , q_{sum})

CAW(before, p_{sum} , q_{sum}) = igaz? (könnyű a döntés)

MSW(overlaps, p_{sum} , q_{sum}) = hamis? (könnyű a döntés)

különben q egy szinttel lebontása ... **sub** q_{sum}

A ← **üzenet** (sub q_{sum}) ← **B**

MSW(overlaps, sub p_{sum} , sub q_{sum}) ??

Stb.

időreláció
előfeltétel

before, overlap, ...
must, may

always, sometimes, first, last

Joint-Intentions, mint a komplex és robusztus tervekészítés mechanizmusa

- Team célok/tervek explicit reprezentálása
- Team tagok: elkötelezettségek és kötelességek mások felé (team cselekvések szempontjából)
- Joint-intentions: pl. **joint persistent goal (JPG)**
(nincs, kell, mielőtt feladják, biz. hogy előállt, v. lehetetlen)
- **JPG feloldása: egyedi ágens logikai feltételei
kommunikációs kényszer (...)**

(PG – tartós cél: nincs, kell, mielőtt eldobjuk, bizonyosodjunk meg, hogy már meglett más módon, ill. nem valósítható meg)