Koordinálás és feladatkiosztás aukciókkal 2.rész

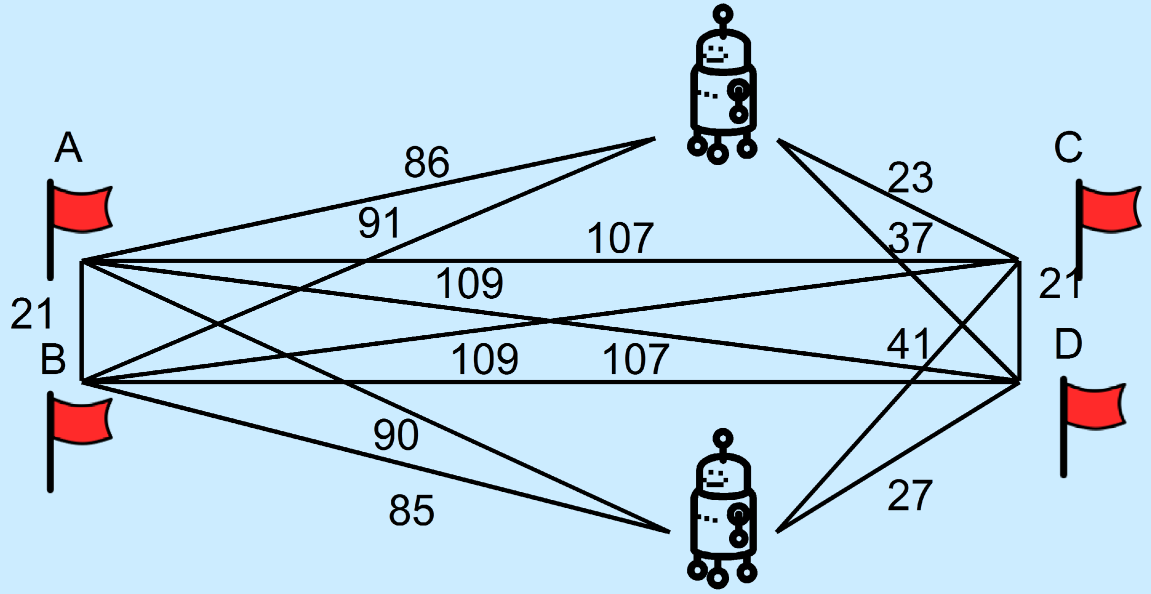
Kooperáció és intelligencia,
Dobrowiecki-Mészáros, BME-MIT

Koordinálás aukciókkal

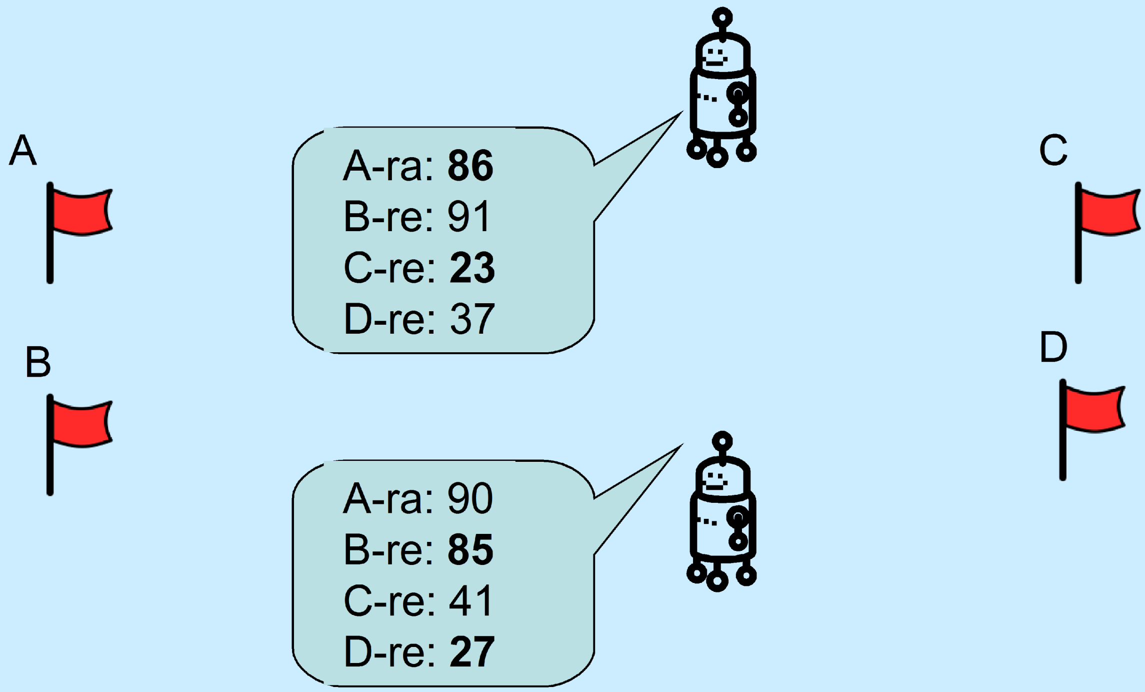
Parallel aukciók
Kombinatorikus aukciók
Szekvenciális aukciók

Parallel aukció:

Minden ágens licitje másoktól független és egyidejű.
A célpontra legkevesebbet licitáló ágens nyer.
Minden ágens meghatározza az elnyert célpontokat összekötő minimális költségű pályát és elkezdi követni.

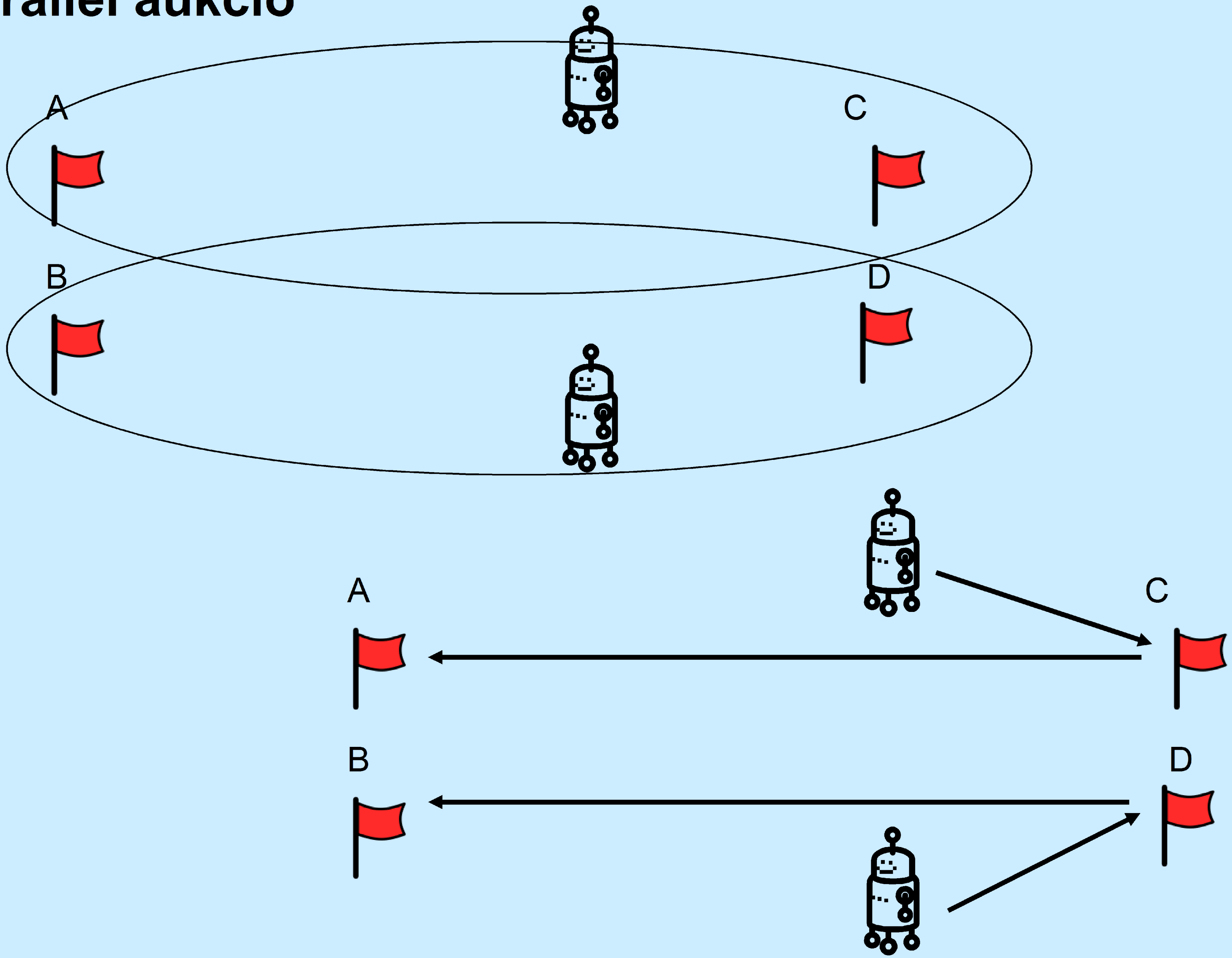


Parallel aukció

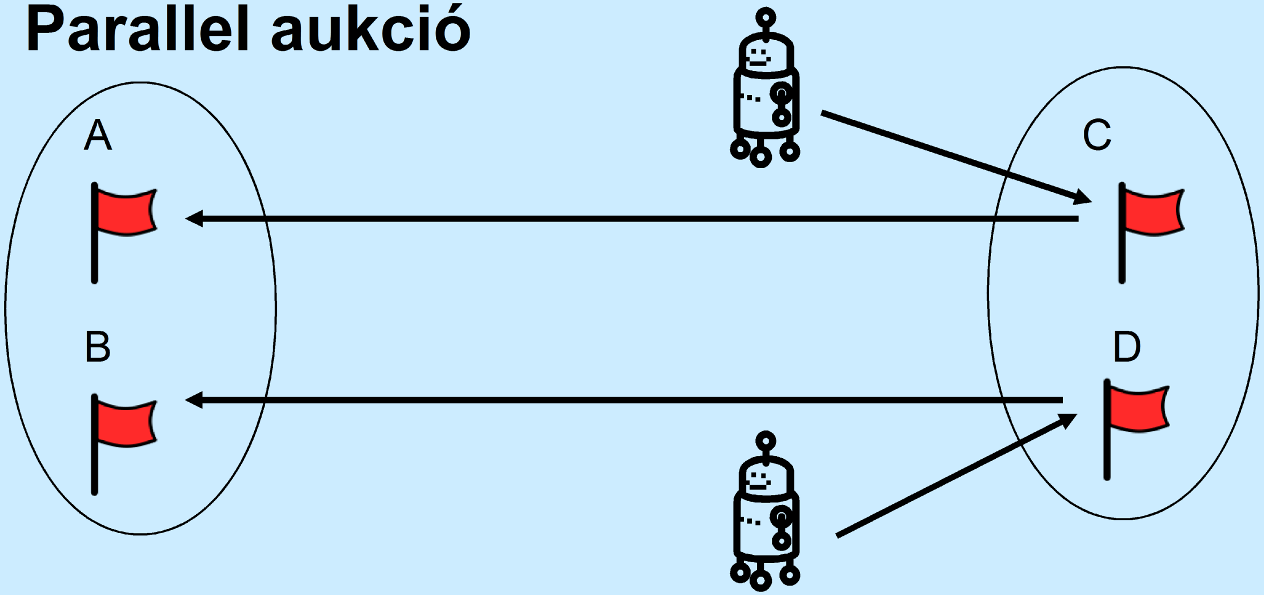


Minden ágens a célpont eléréséhez az adott helyzetéből kiindulva a minimális költségű pályát licitálja.

Parallel aukció

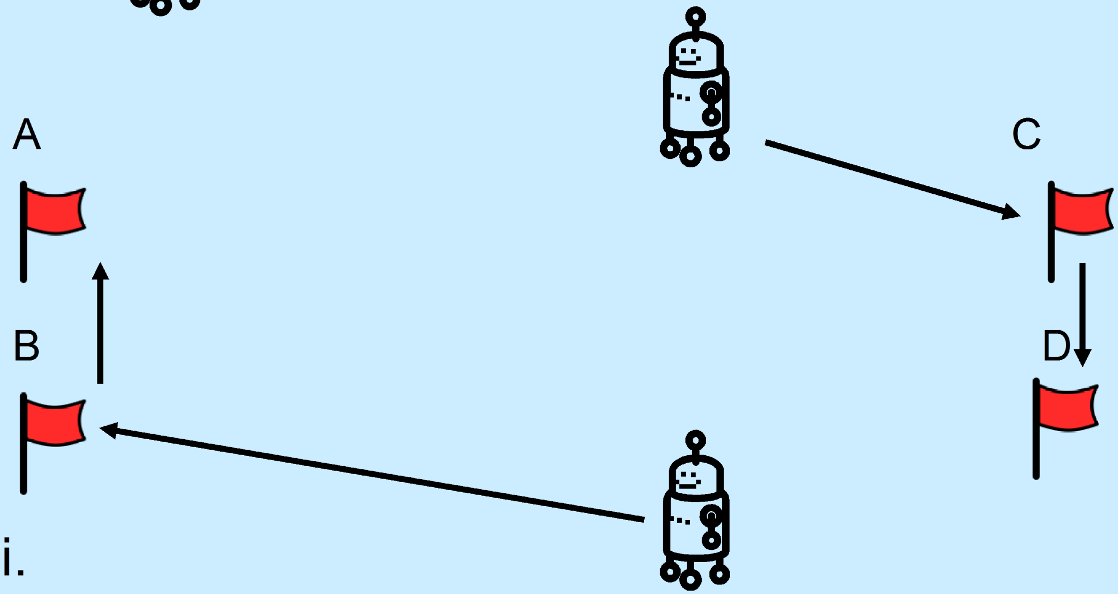


Parallel aukció

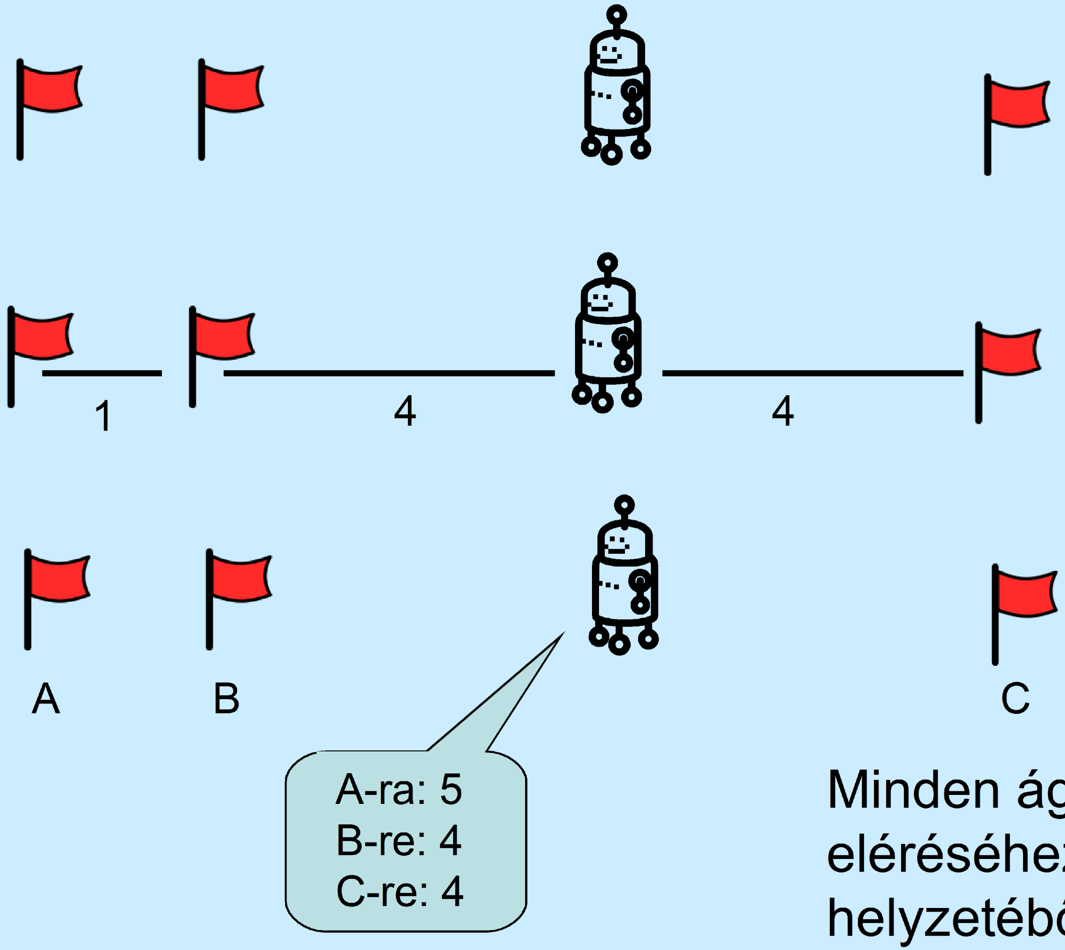


Sokszor felesleges különböző ágenseket ugyanahhoz a célpontklaszterhez küldeni.

A minimális team költséget nem sikerült elérni.
Parallel aukciókból származó **team költség magas**, mert a célpontok közötti **kölcsönhatást** figyelembe nem tudja venni.

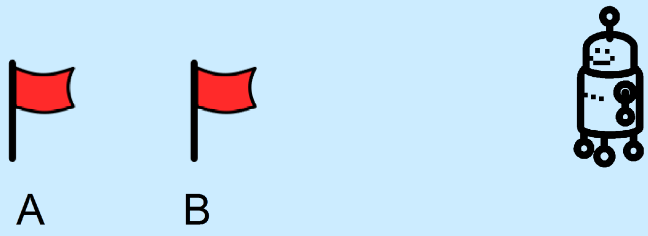


Kölcsönhatások



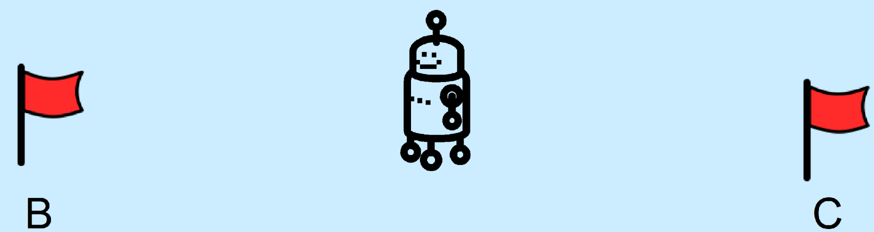
Minden ágens a célpont eléréséhez az adott helyzetéből kiindulva a minimális költségű pályát licitálja.

Kölcsönhatások



Min pályaköltség (A): 5
Min pályaköltség (B): 4 **Min pályaköltség (A és B): 5**

Min pályaköltség (A és B) < Min pályaköltség (A) + Min pályaköltség (B)



Min pályaköltség (B): 4
Min pályaköltség (C): 4 **Min pályaköltség (B és C):12**

Min pályaköltség (B és C) > Min pályaköltség (B) + Min pályaköltség (C)

Parallel aukció:

Implementáció: egyszerű

Decentralizálás: egyszerű

Licitgenerálás: olcsó

Licitkommunikáció: olcsó

Aukció lebonyolítás: olcsó

Team hatékonyság: **gyenge**

kölcsönhatások nincsenek figyelembe véve

Ideális kombinatorikus aukció:

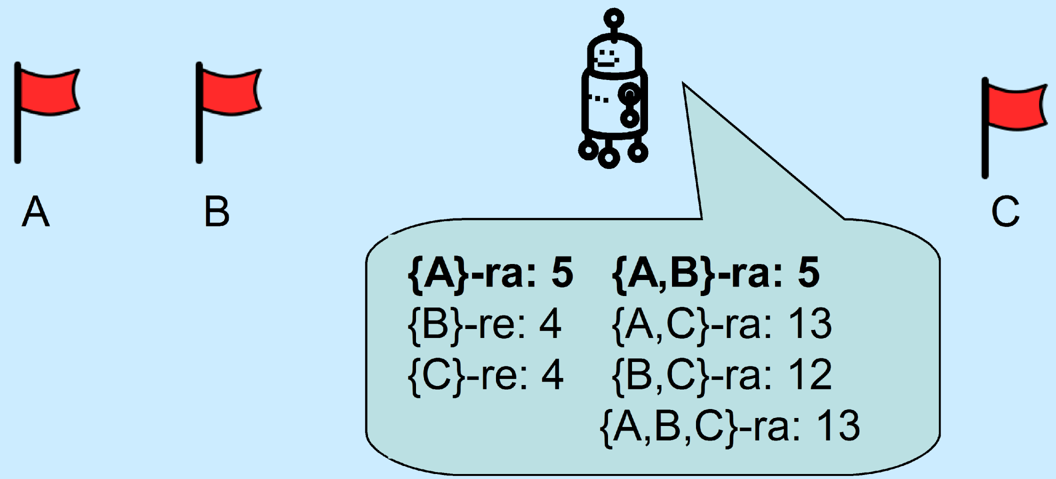
Minden ágens **minden** célkötegre (= részhalmazra) licitál.

Minden ágens legfeljebb egy köteget (de több feladatot) nyer.

Minden feladat kap végrehajtót.

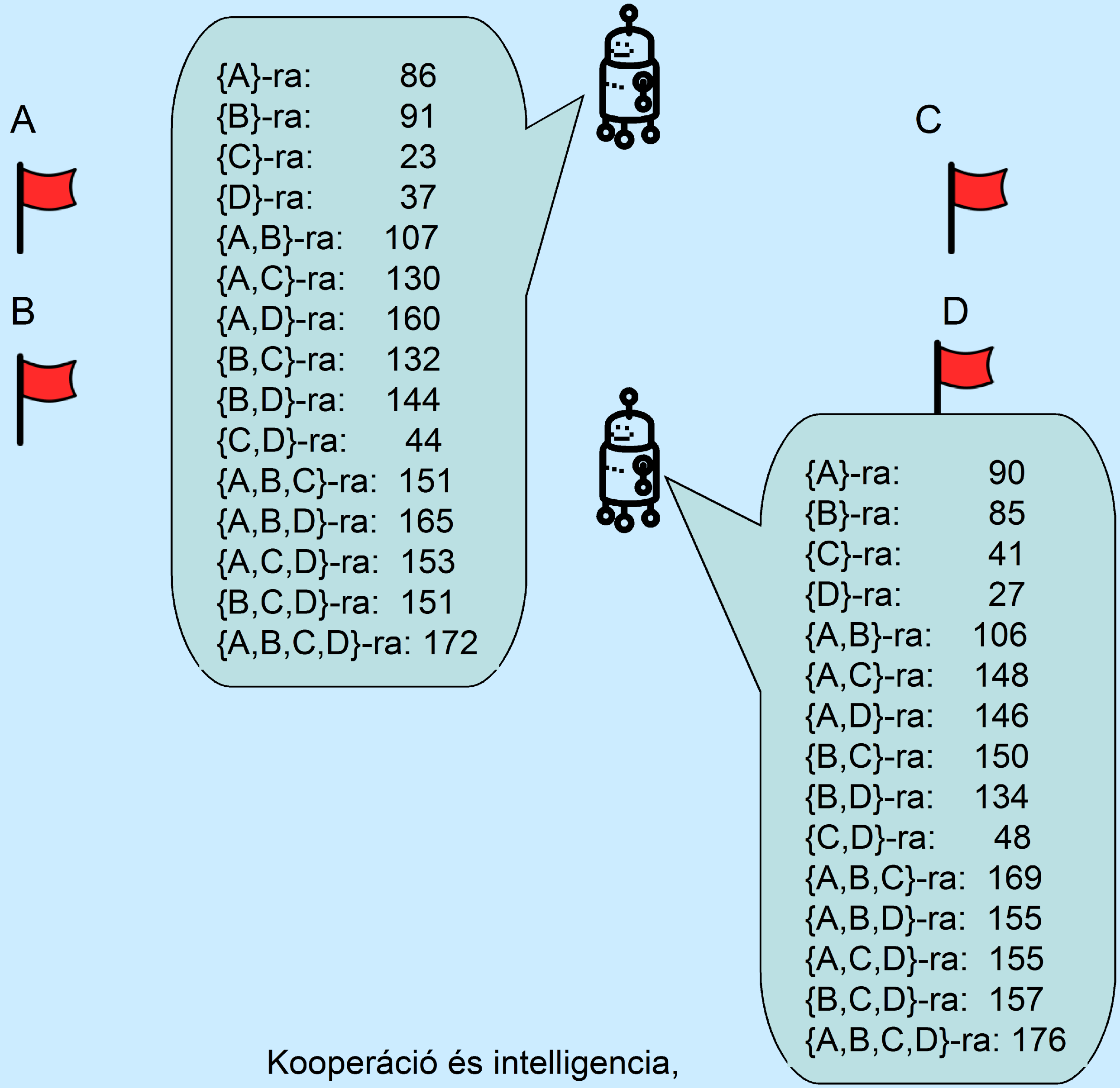
Minden ágens a elnyert célpontokhoz kiszámítja a költségminimális pályát és elkezdi követni.

Ideális kombinatorikus aukció: Kölcsönhatás



Minden ágens a kötegben foglalt minden célpont eléréséhez az adott helyzetéből kiindulva a minimális költségű pályát licitálja.

Ideális kombinatorikus aukció: Példa



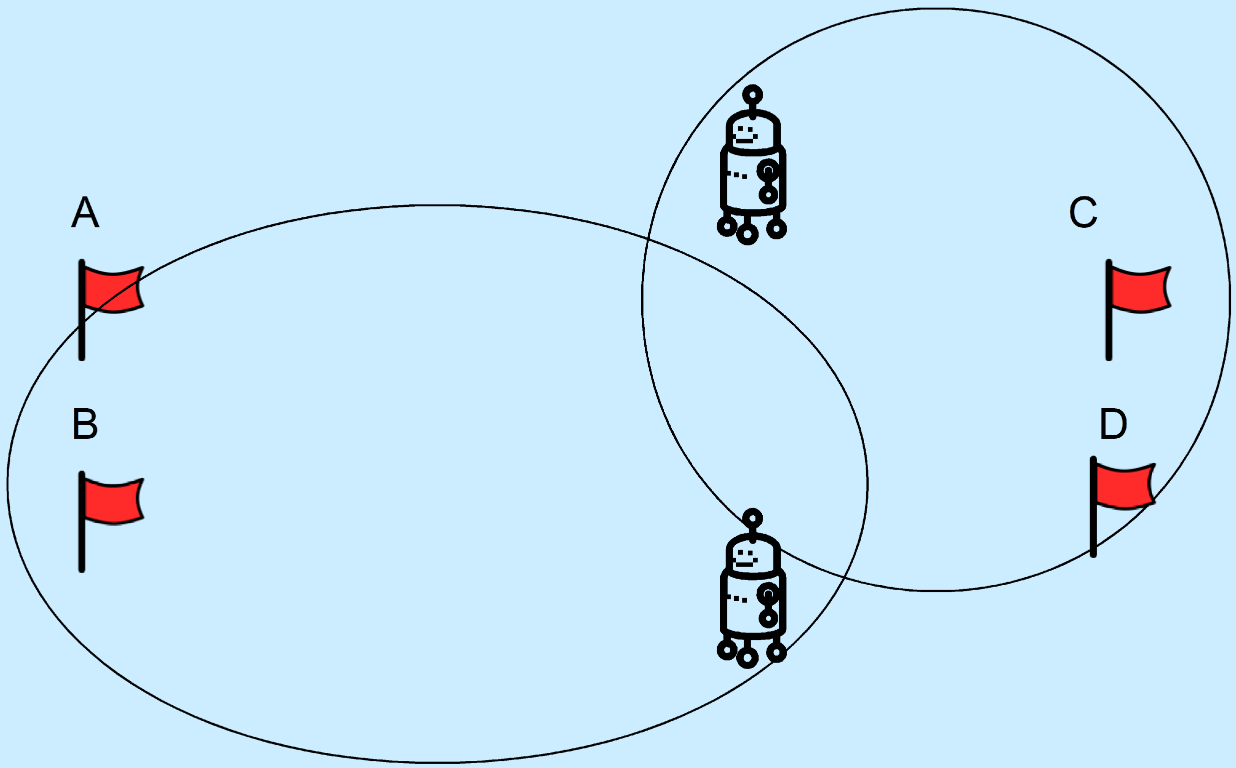
Ideális kombinatorikus aukció: Példa

{A}-ra:	**86**
{B}-ra:	91
{C}-ra:	**23**
{D}-ra:	37
{A,B}-ra:	107
{A,C}-ra:	**130**
{A,D}-ra:	160
{B,C}-ra:	**132**
{B,D}-ra:	144
{C,D}-ra:	**44**
{A,B,C}-ra:	**151**
{A,B,D}-ra:	165
{A,C,D}-ra:	**153**
{B,C,D}-ra:	**151**
{A,B,C,D}-ra:	**172**

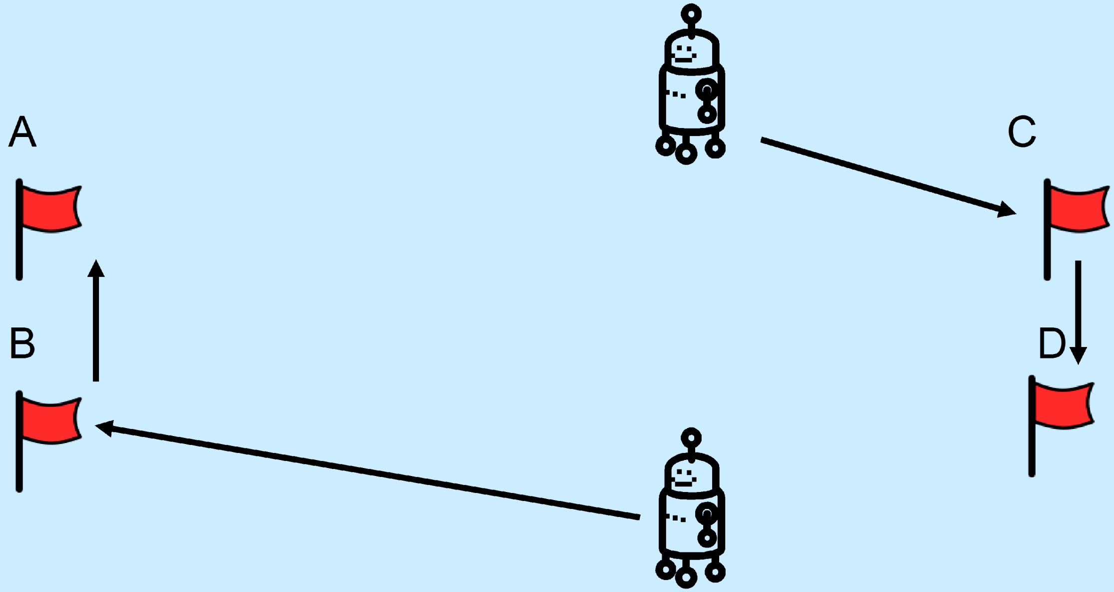
{A}-ra:	90
{B}-ra:	**85**
{C}-ra:	41
{D}-ra:	**27**
{A,B}-ra:	**106**
{A,C}-ra:	148
{A,D}-ra:	**146**
{B,C}-ra:	150
{B,D}-ra:	**134**
{C,D}-ra:	48
{A,B,C}-ra:	169
{A,B,D}-ra:	**155**
{A,C,D}-ra:	155
{B,C,D}-ra:	157
{A,B,C,D}-ra:	176

-	{A,B,C,D}	176
{A}	{B,C,D}	243
{B}	{A,C,D}	246
{C}	{A,B,D}	178
{D}	{A,B,C}	206
{A,B}	**{C,D}**	**155**
{A,C}	{B,D}	264
{A,D}	{B,C}	310
{B,C}	{A,D}	278
{B,D}	{A,C}	288
{C,D}	**{A,B}**	**150**
{A,B,C}	{D}	178
{A,B,D}	{C}	206
{A,C,D}	{B}	238
{B,C,D}	{A}	241
{A,B,C,D}	-	172

Ideális kombinatorikus aukció: Példa



Ideális kombinatorikus aukció: Példa



Ideális kombinatorikus aukcióból adódó **team költség minimális**, mert minden kölcsönhatást vesz figyelembe a célpontok között, ami egy **NP-nehéz** probléma megoldása.
A licitek száma **exponenciális** a célpontok számában.
Licitgenerálás, -kommunikáció és a győztes kiszámítása költséges.

Kombinatorikus aukció/2

Minden ágens **néhány** célkötegre (= halmazra) licitál.

Minden ágens legfeljebb egy köteget nyer.

Minden ágens a elnyert célpontokhoz kiszámítja a költség minimális pályát és elkezdi követni.

Ilyen kombinatorikus aukcióból adódó **team költség alacsony**, de **szuboptimális** lehet.

Licitgenerálás, -kommunikáció és a győztes kiszámítása még mindig (viszonylag) költséges.

Kombinatorikus aukció/2

Licit stratégiák
Melyik kötegre licitálni nagyban feltáratlan, mert egy jó kötegggenerálási stratégia feladatfüggő.

Jó kötegggenerálási stratégiák:

Kis számú köteget generálni

A megoldásteret **lefedő** kötegeket generálni

Jövedelmező kötegeket generálni

Kötegeket **hatékonyan** generálni

…

Kombinatorikus aukció/2

Dómén-független köteggenerálás

Buta köteggenerálás esetén minden kötegre licitálunk.

HÁROM-KOMBINÁCIÓ

3 vagy kevesebb célpontot tartalmazó kötegre licitálni

Megjegyzés: Lehet, hogy az összes célpont hozzárendelése nem lesz lehetséges.

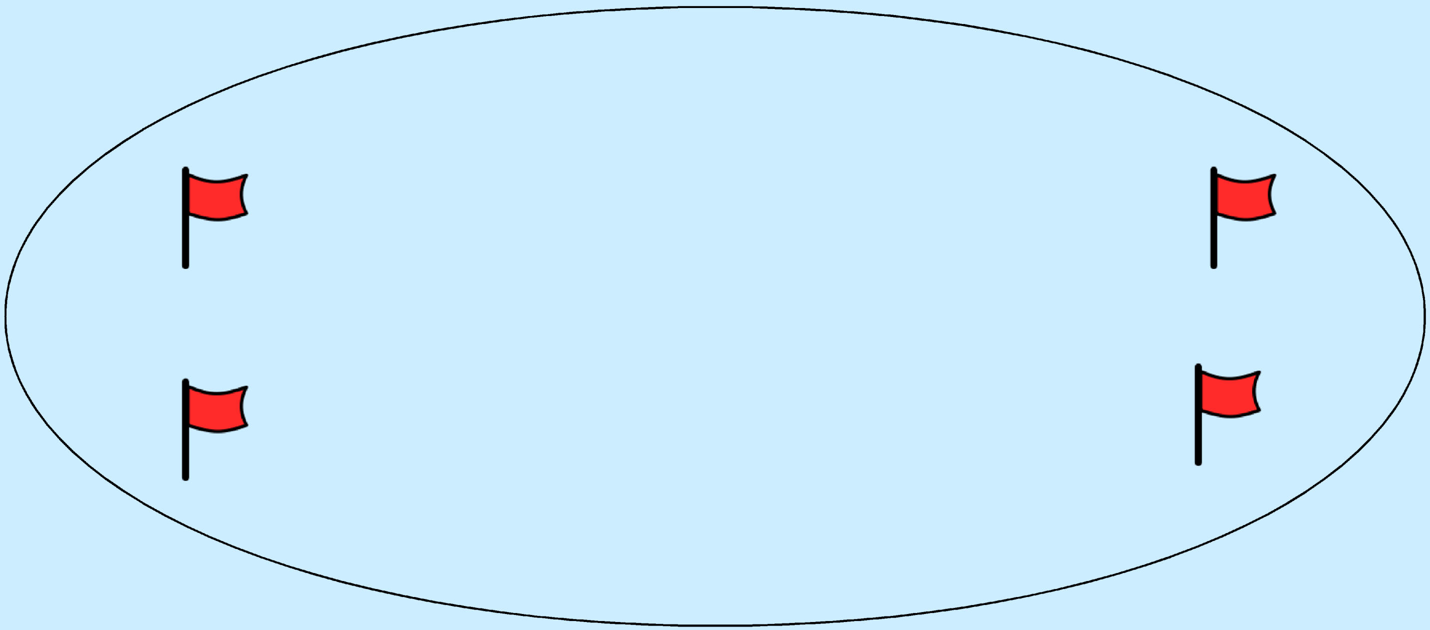
Dómén-függő köteggenerálás

Okos köteggenerálás célpontok klasztereire licitál.

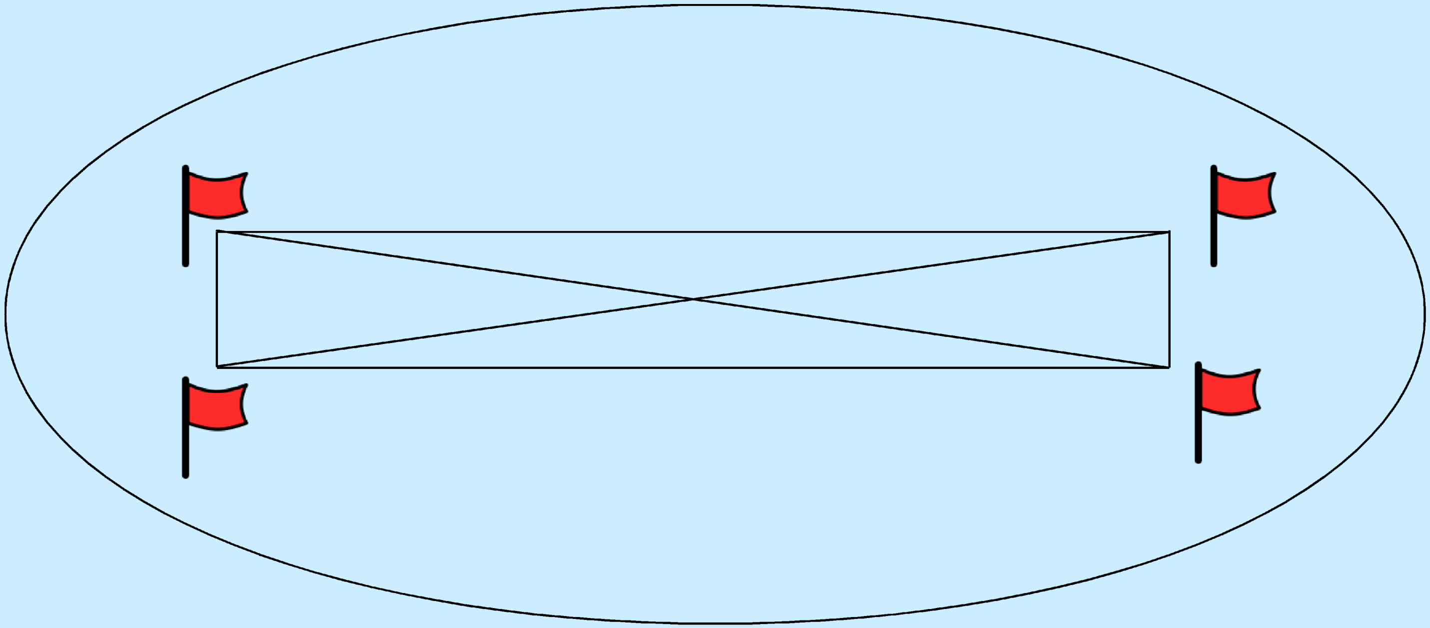
GRAPH-CUT

Kezdjük az összes célpontot tartalmazó köteggel. Építsünk egy teljes gráfot, melynek csúcsai a kötegbeli célpontok és élek költsége a csúcsok közötti pályaköltség. Vágjuk a gráfot ketté a maximális vágás (közelítése) mentén. Ismételjük az eljárást (kétszer) mindkét részgráf célpontjai esetén.

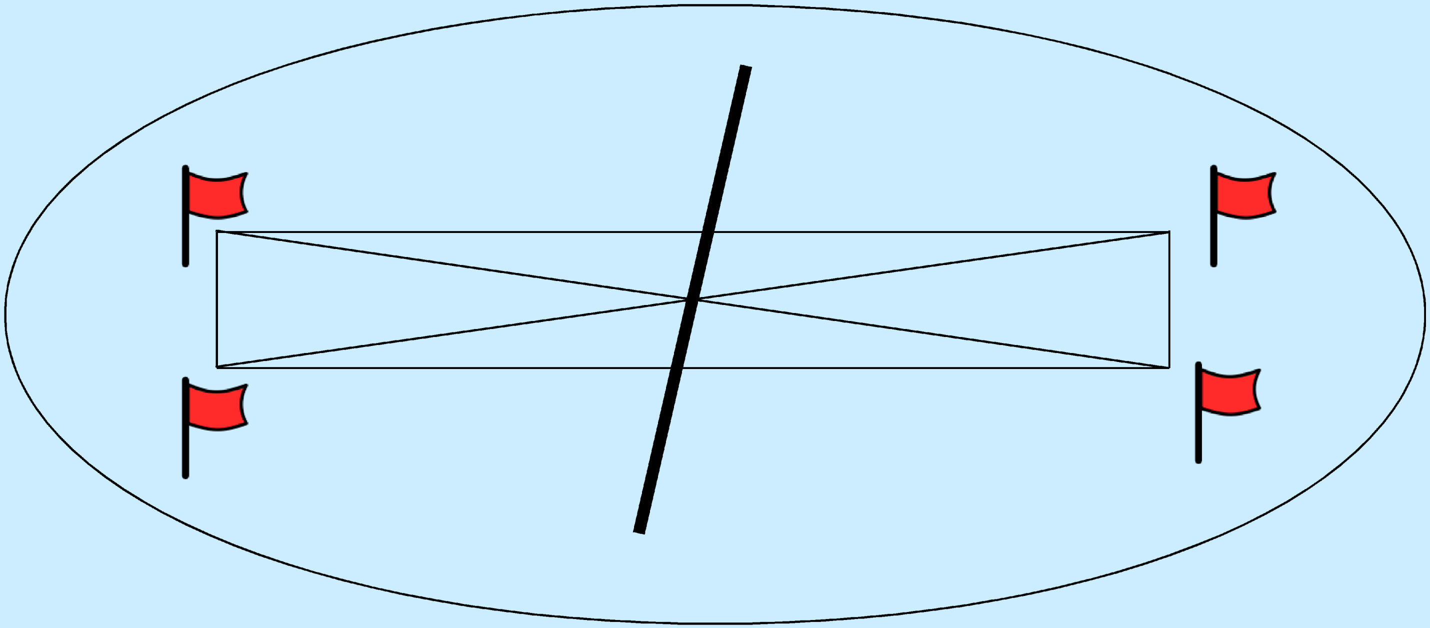
Kombinatorikus aukció/2 - Dómén-függő köteggenerálás



Kombinatorikus aukció/2 - Dómén-függő köteggenerálás

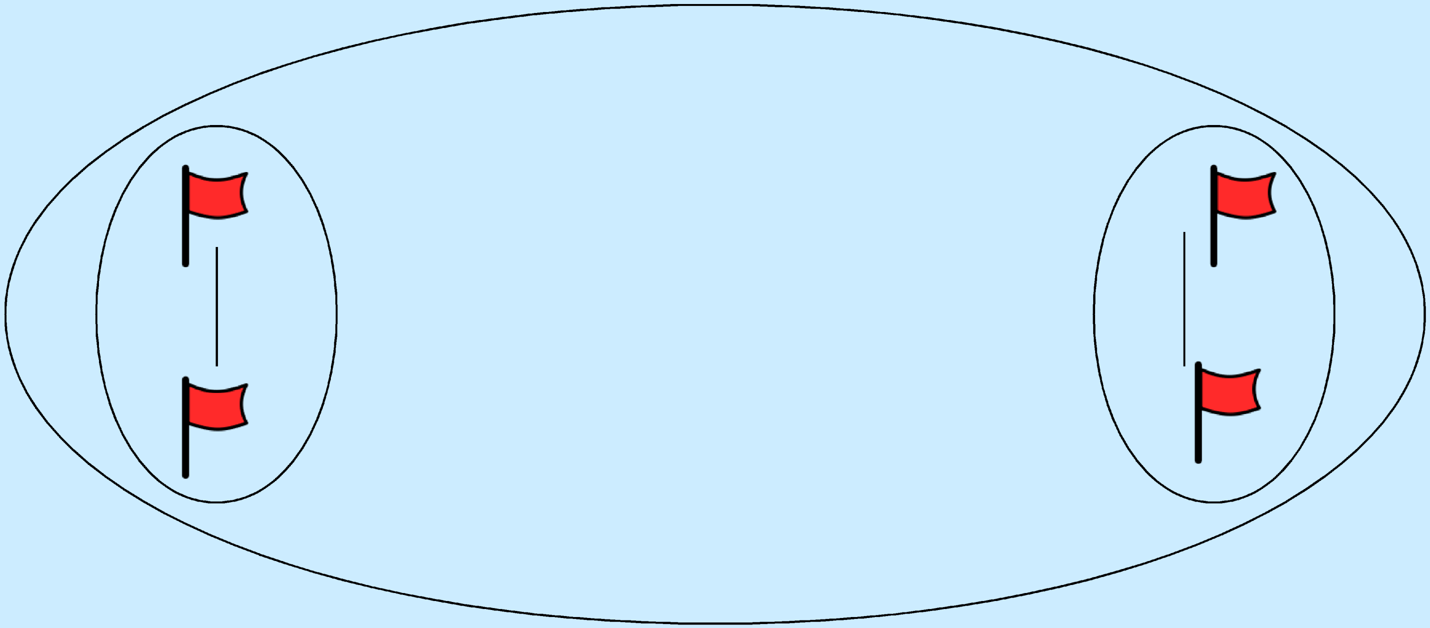


Kombinatorikus aukció/2 - Dómén-függő köteggenerálás

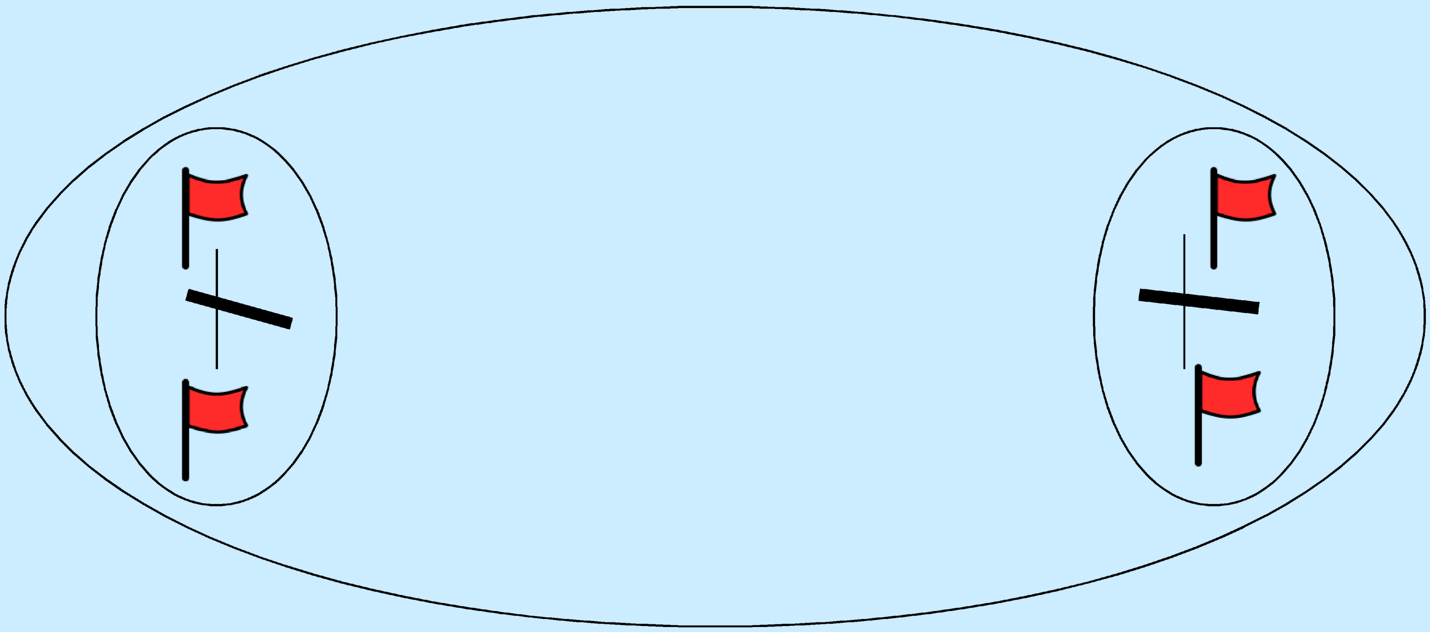


Vágás = két halmazzá, ami a gráf csúcsait particionálja.
Maximális vágás = maxcut = a két csúcshalmazt összekötő élek költségeit maximáló vágás.
Maximális vágás megkeresése NP-nehéz, de approximálni kell.

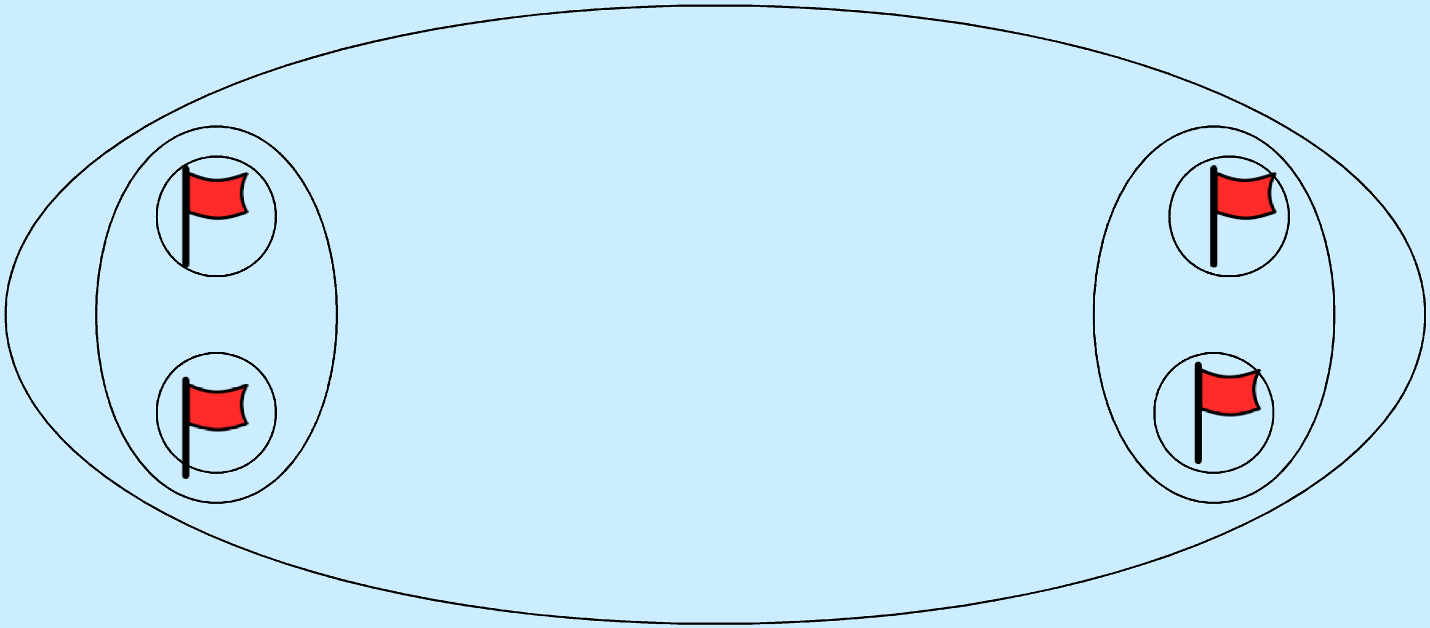
Kombinatorikus aukció/2 - Dómén-függő köteggenerálás



Kombinatorikus aukció/2 - Dómén-függő köteggenerálás



Kombinatorikus aukció/2 - Dómén-függő köteggenerálás



Az alábbi kötegekre kell licitálni
(hierarchikus licit, ld. előbb)

{A}, {B}, {C}, {D}
{A,B}, {C,D}
{A,B,C,D}

Kombinatorikus aukció/2

Kísérletek ismert terepen
3 ágens, ismert terep, 5 db. célpont klaszter á 4 célpont

	Team SUM
Parallel aukció	427
Kombinátorikus akció	
Három Kombináció heur.	248
Graph-Cut heur.	184
Optimális (ideális, IP)	**184**

Kombinatorikus aukció

Implementáció: nehéz

Decentrálizálás: tisztázatlan

Köteggenerálás: drága (lehet NP-nehéz)

Licitgenerálás/ köteg: ok (NP-nehéz)

Licitkommunikáció: drága

Aukció lebonyolítás: drága (NP-nehéz)

Team hatékonyság: **nagyon jó** (optimális)
sok (minden) összhang figyelembe vétele

Használjuk ügyes köteggenerálást.

Különböző NP-nehéz problémák approximációja.

Szekvenciális aukció

Parallel Aukció

Implementáció: egyszerű
Decentralizálás: egyszerű
Licitgenerálás: olcsó
Licitkommunikáció: olcsó
Aukció lebonyolítás: olcsó
Team hatékonyság: gyenge

Kombinatorikus Aukció

Implementáció: nehéz
Decentralizálás: tisztázatlan
Licitgenerálás: drága
Licitkommunikáció: drága
Aukció lebonyolítás: drága
Team hatékonyság: “optimális”

Szekvenciális aukció jó középút a parallel és a kombinatorikus aukció között.

Szekvenciális aukció

Néhány licitkör elteltével ágensek minden célpontot elnyernek.

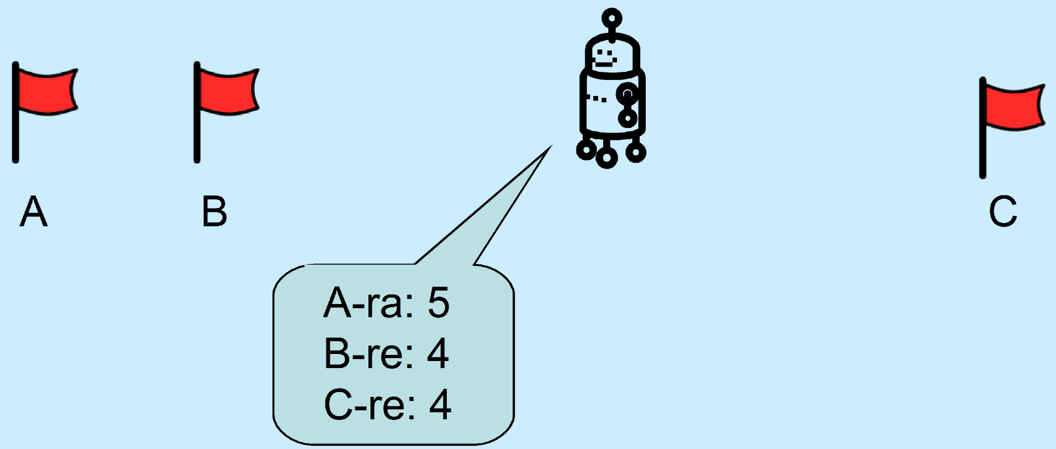
Körönként csak egy célpont kell el.

Egy licitkörben minden ágens az összes, mások által nem elnyert célpontra licitál.

A győztes az ágensekre és a célpontokra nézve minimális licit.
(A licitáló ágens elnyeri az adott célpontot)

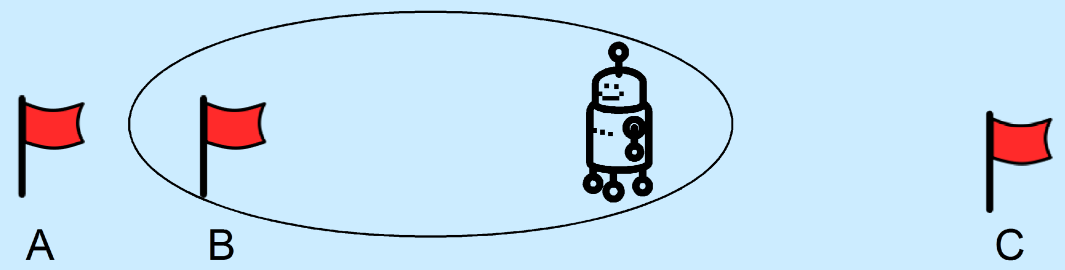
Minden ágens a elnyert célpontokhoz kiszámítja a költség minimál pályát és elkezdi követni.

Szekvenciális aukció – Kölcsönhatás



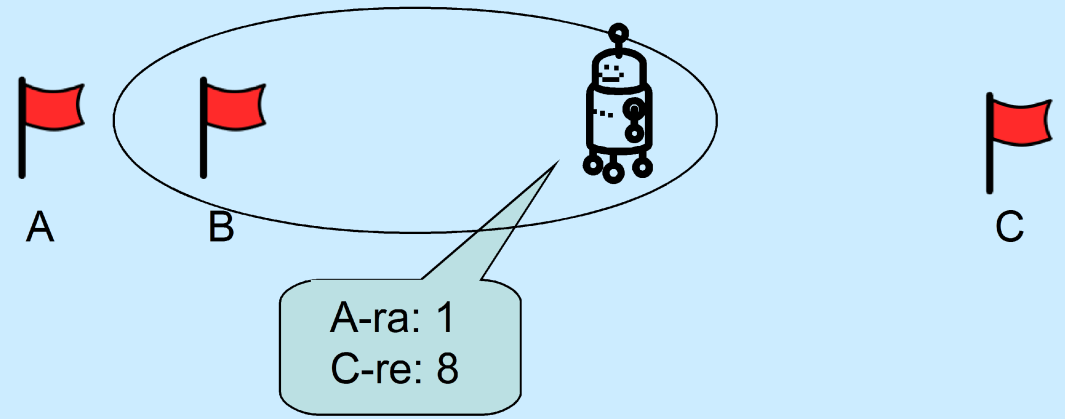
Minden ágens az új célpontra az adott pozíciójából kiinduló és az összes célpont meglátogatásához szükséges minimális költségű pályában jelentkező költségnövekményt licitálja, ha a célpontot történetesen ő nyerné meg. (**BidSumPath**).

Szekvenciális aukció - Kölcsönhatás



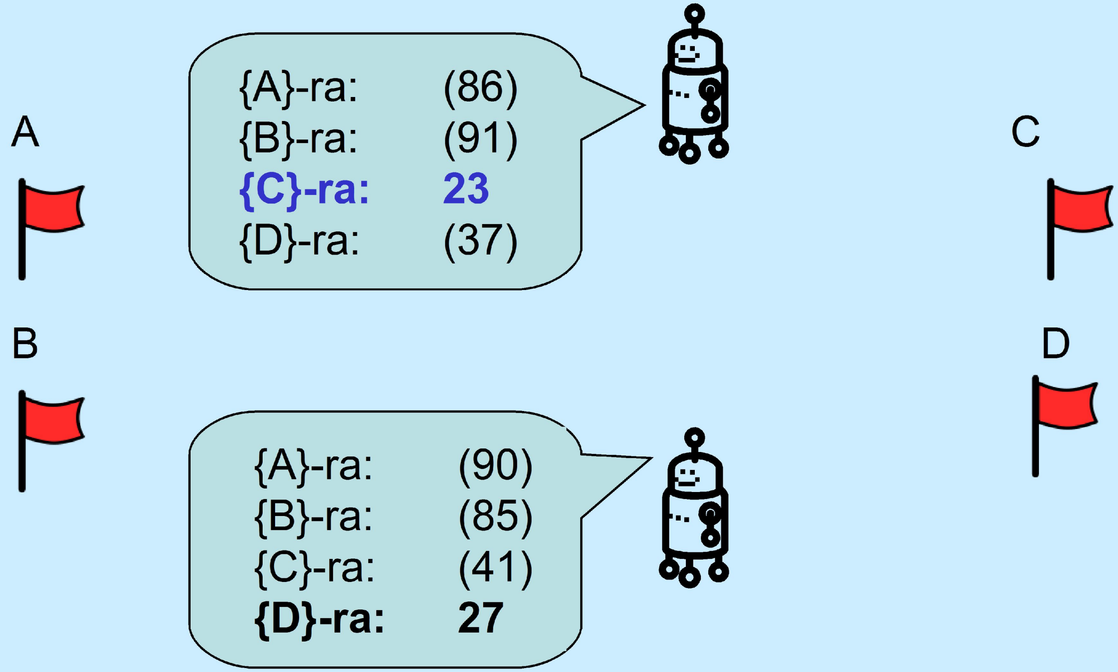
Minden ágens az új célpontra az adott pozíciójából kiinduló és az összes célpont meglátogatásához szükséges minimális költségű pályában jelentkező költségnövekményt licitálja, ha a célpontot történetesen ő nyerné meg. (**BidSumPath**).

Szekvenciális aukció - Kölcsönhatás

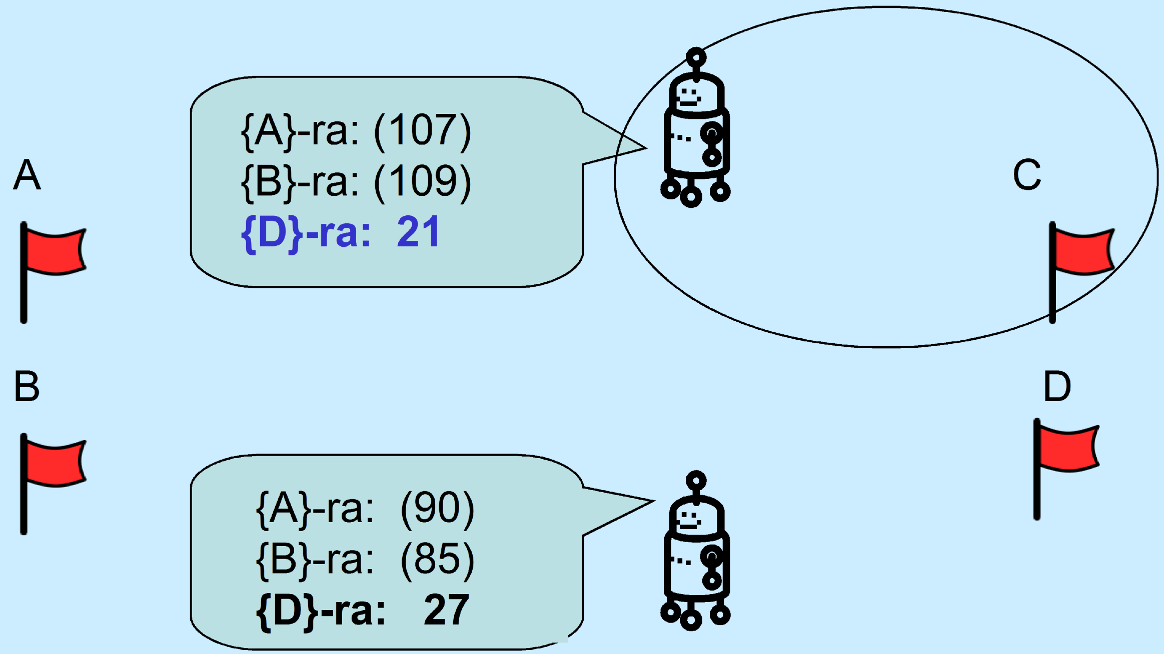


Minden ágens az új célpontra az adott pozíciójából kiinduló és az összes célpont meglátogatásához szükséges minimális költségű pályában jelentkező költségnövekményt licitálja, ha a célpontot történetesen ő nyerné meg. (**BidSumPath**).

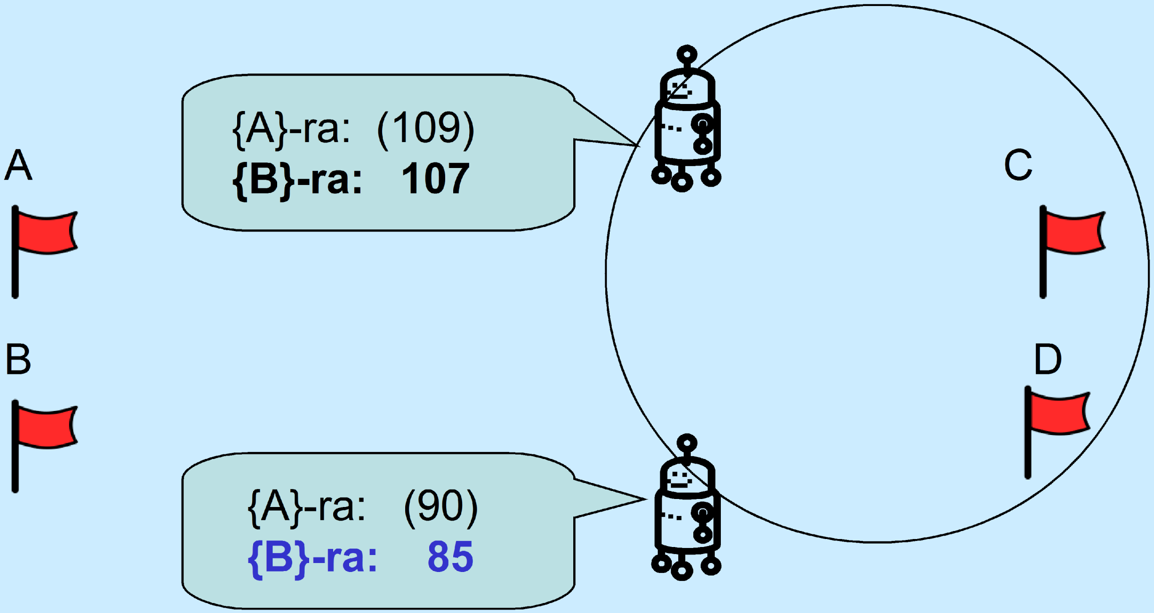
Szekvenciális aukció - Példa



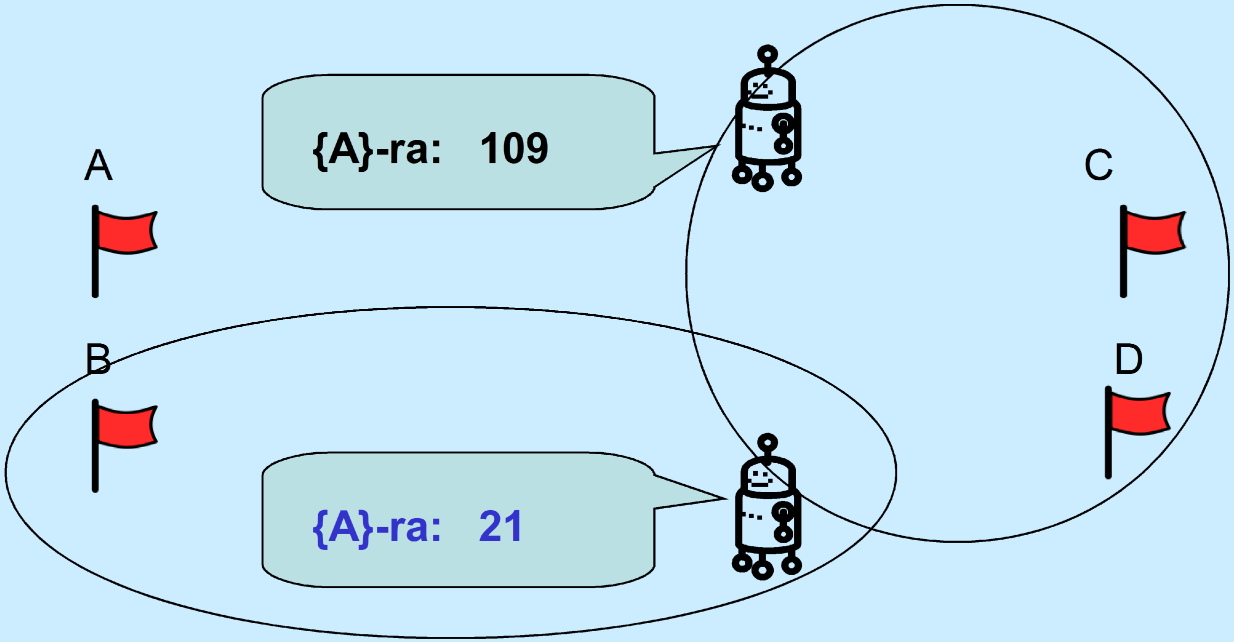
Szekvenciális aukció - Példa



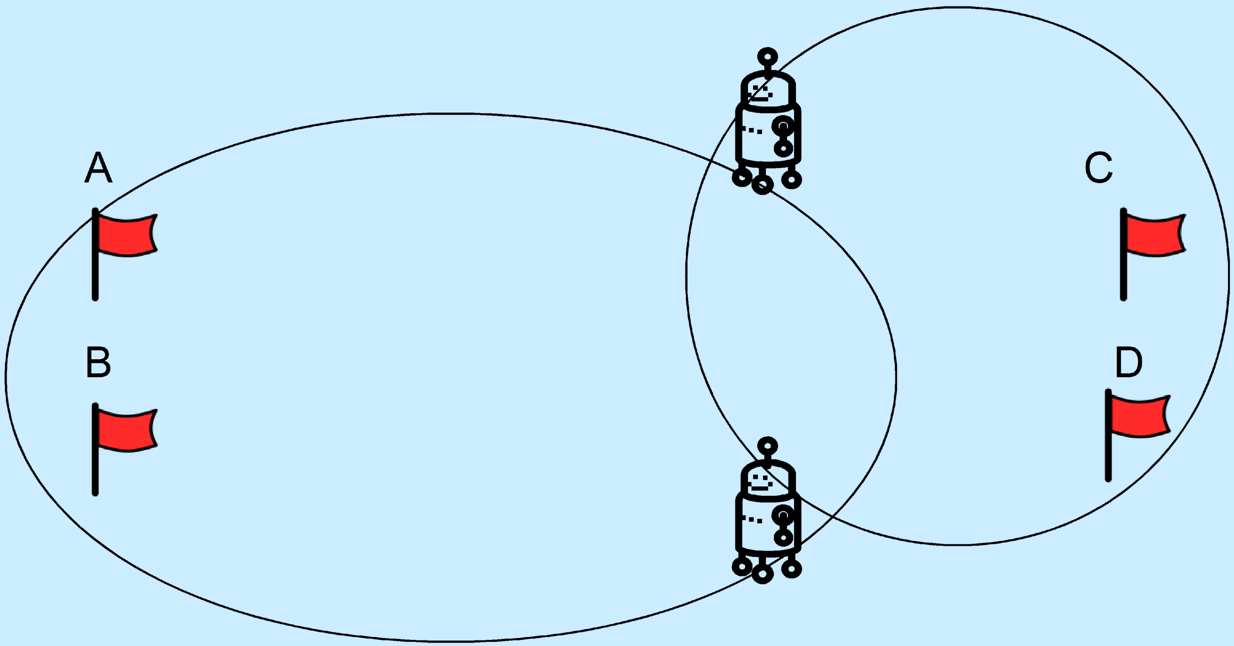
Szekvenciális aukció - Példa



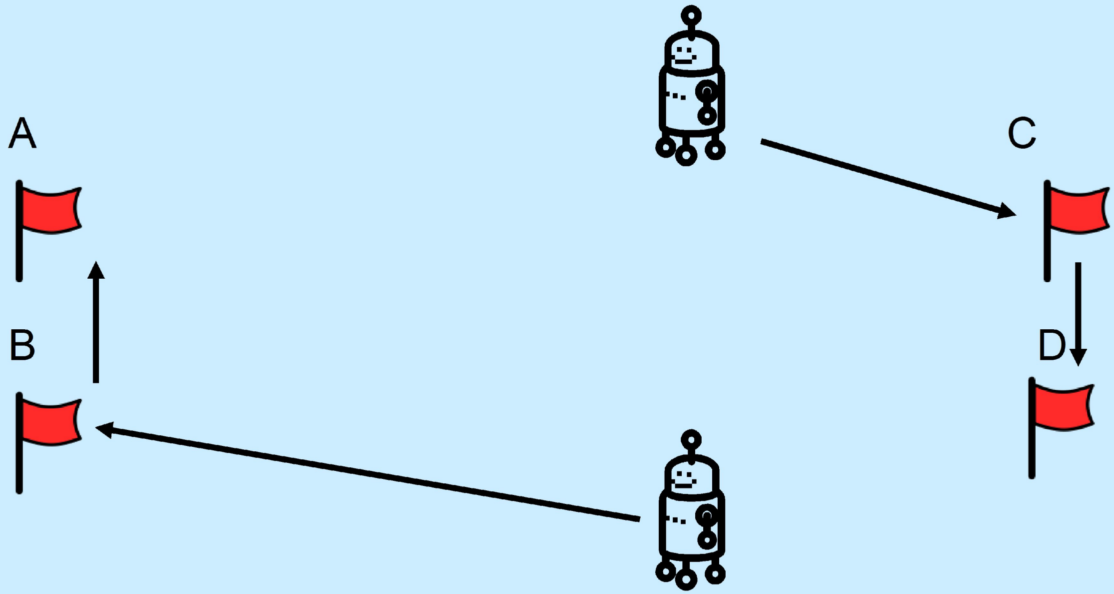
Szekvenciális aukció - Példa



Szekvenciális aukció - Példa



Szekvenciális aukció - Példa



Szekvenciális aukció

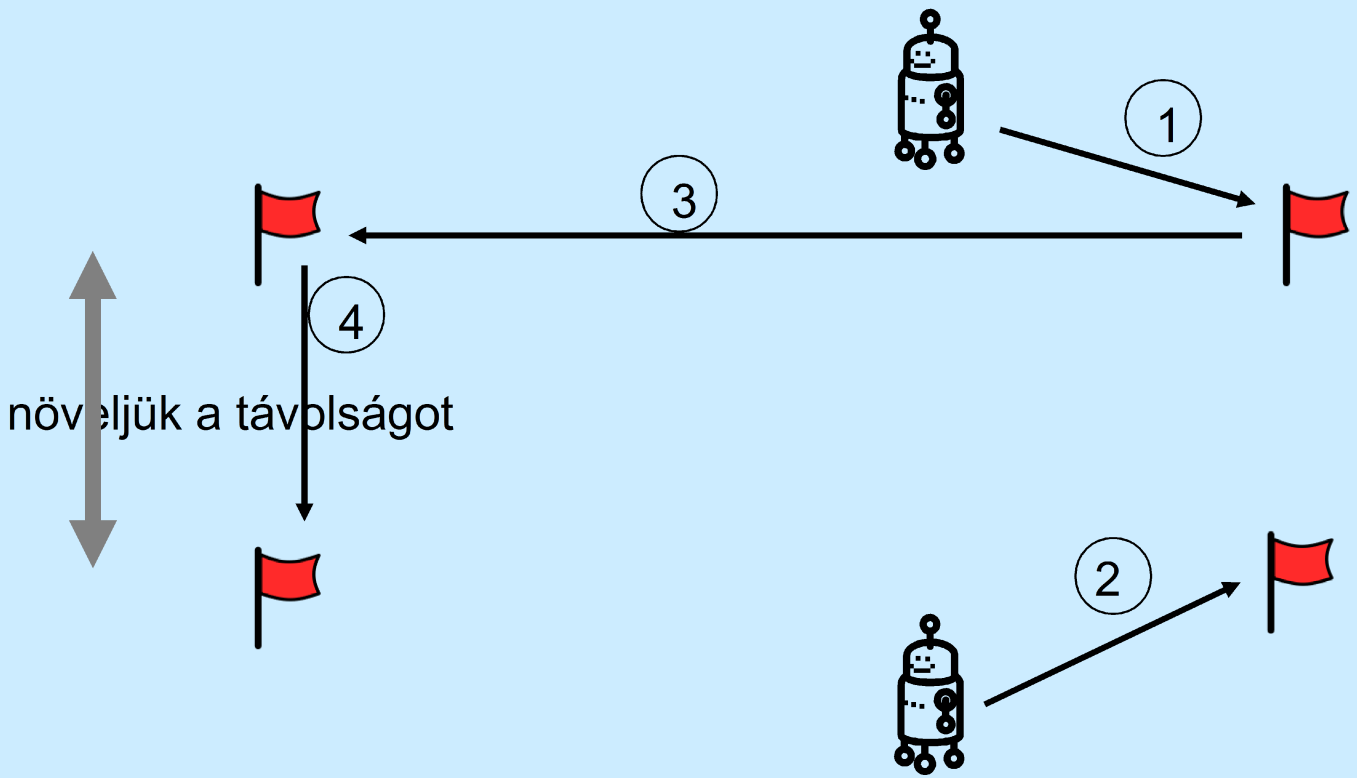
Minden ágensnek elegendő csak a minimális licitek egyikét jelezni.

Minden ágensnek az új licitet csak akkor kell jeleznie, ha az eddig licitált célpontot valaki elnyerte. (vagy ő maga,vagy más ágens).

Minden ágens legfeljebb egy licittel jelentkezik körönként és a **körök száma azonos a célpontok számával**.
Így az összes licit száma nem több, mint a parallel aukciónál, és a licitkommunikáció olcsó.

(példában a jelezni nem szükséges licitek zárójel vannak).

Szekvenciális aukció - Példa



Szekvenciális aukcióból adódó team költség **nem szükségképpen minimális**, mert nem minden kölcsönhatás van figyelembe véve

Szekvenciális aukció

Implementáció: **viszonylag egyszerű**

Decentralizáció: **egyszerű**

Licitgenerálás: **olcsó**

Licitkommunikáció: **olcsó**

Aukció lebonyolítás: **olcsó**

Team hatékonyság: **nagyon jó**
kölcsönhatások egy része figyelembe vett

Szekvenciális aukció Licitszabályok származtatása

Javaslat: automatikus licittervezés hegymászó kereséssel.

Legyen az a győztes, amely mellett a team költség valamilyen mérce legkevesebbet nő.

Egy ágens egy célra azt az összeget licitálja, ami a minimális team költségben jelentkező különbség a jelenlegi célhozzárendelés és a nyerésével módosított célhozzárendelés között.
(a még nem elnyert célokat figyelmen kívül kell hagyni.)

Pályák licitálása ("direkt megközelítés")
Közvetlenül találja meg a pályát
Fák licitálása ("indirekt megközelítés")
Fákat talál és azokat pályákká konvertálja

(Pl. min. költségű minimális kifeszítő fa = P komplexitás
min. költségű pálya = NP komplexitás)

Szekvenciális aukció

Pályalicit szabályok származtatása

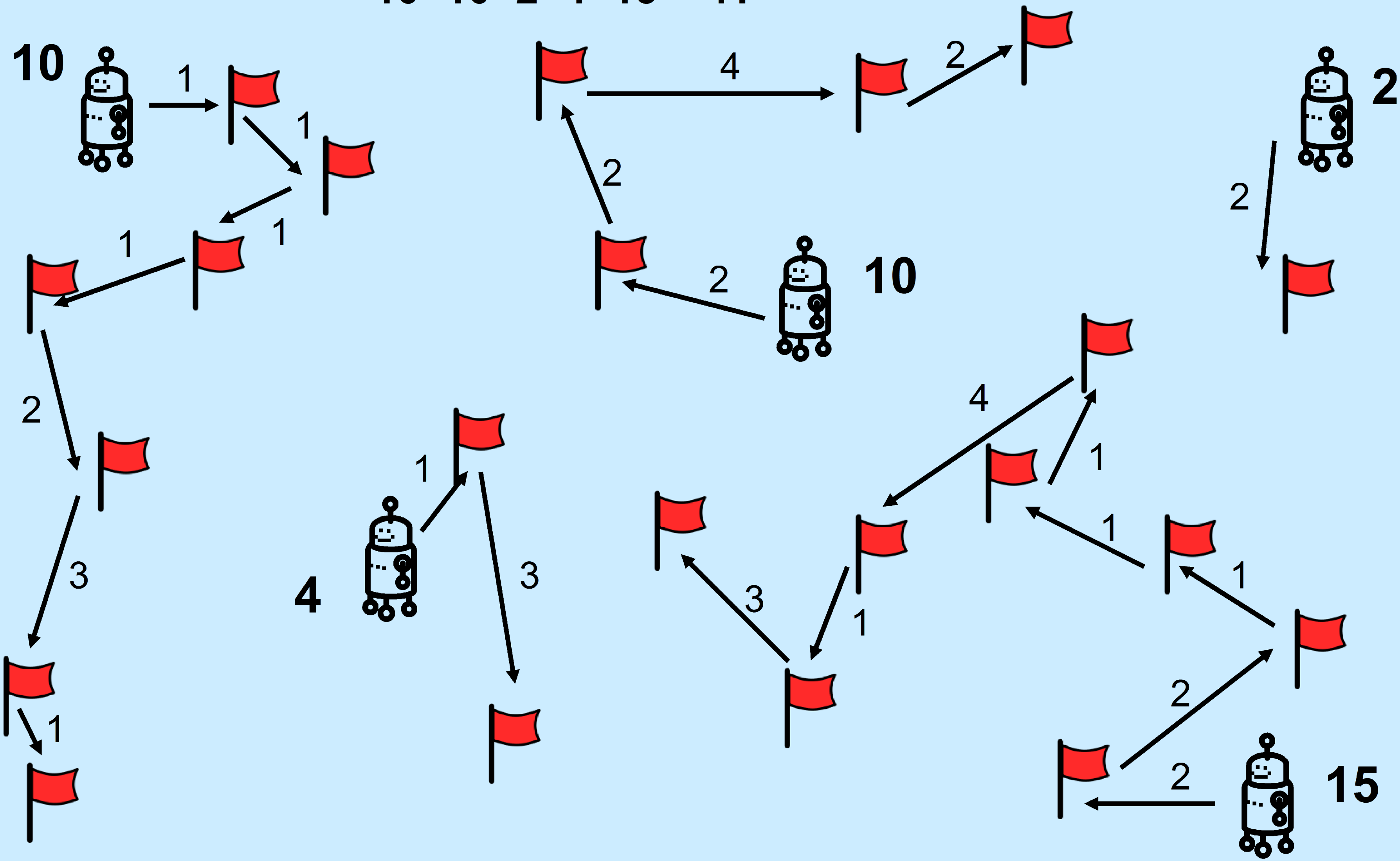
MiniSum
Pályaköltségek összegének minimálása az összes ágensre nézve
Teljes energia, vagy távolság minimálása
Alkalmazás: bolygófelszín kutatása

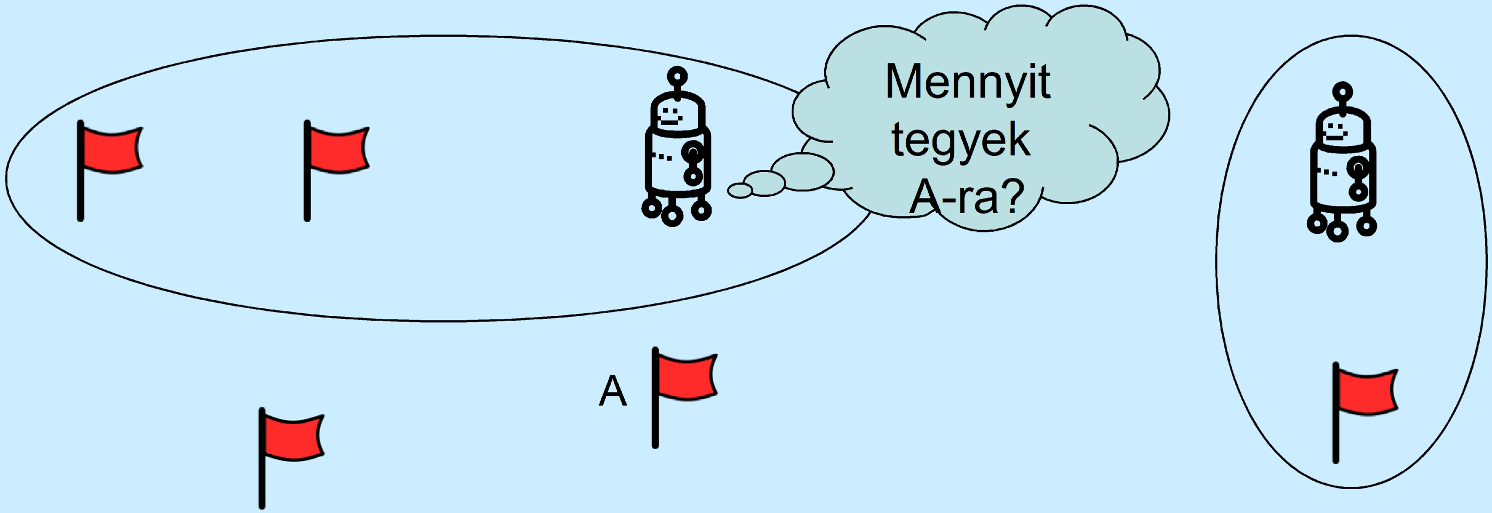
MiniMax
Maximális pályaköltség minimálása az összes ágensre nézve
Teljes feladatvégzési idő minimálása (makespan)
Alkalmazás: objektum-felügyelet, bányatakarítás

MiniAve
Átlagos érkezési idő minimálása az összes célpontra nézve
Átlagos kiszolgálási idő minimálása (flowtime)
Alkalmazás: search and rescue

Tipikus koordinációs feladat: MiniSum Team célfüggvény

10+10+2+4+15 = 41





Szekvenciális aukció

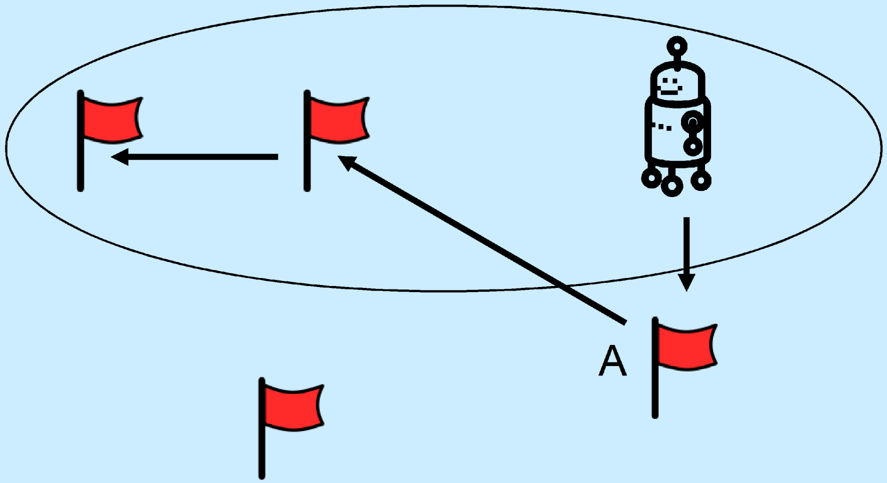
Pályalicit szabályok származtatása

MiniSum = energia vagy távolság

Szekvenciális aukció

Pályalicit szabályok származtatása

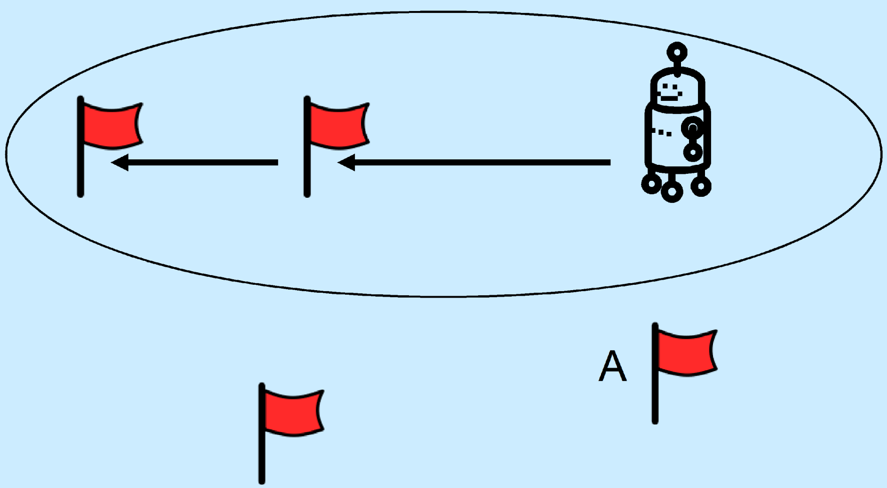
MiniSum = energia vagy távolság



(annak a minimális költségű pályának a költsége, amivel az összes elnyert célpontot bejárhatja, feltéve hogy a licitált célpontot elnyeri)

minusz

mínusz



(annak a minimális költségű pályának a költsége, amivel az összes eddig elnyert célpontot bejárhatja)

Szekvenciális aukció

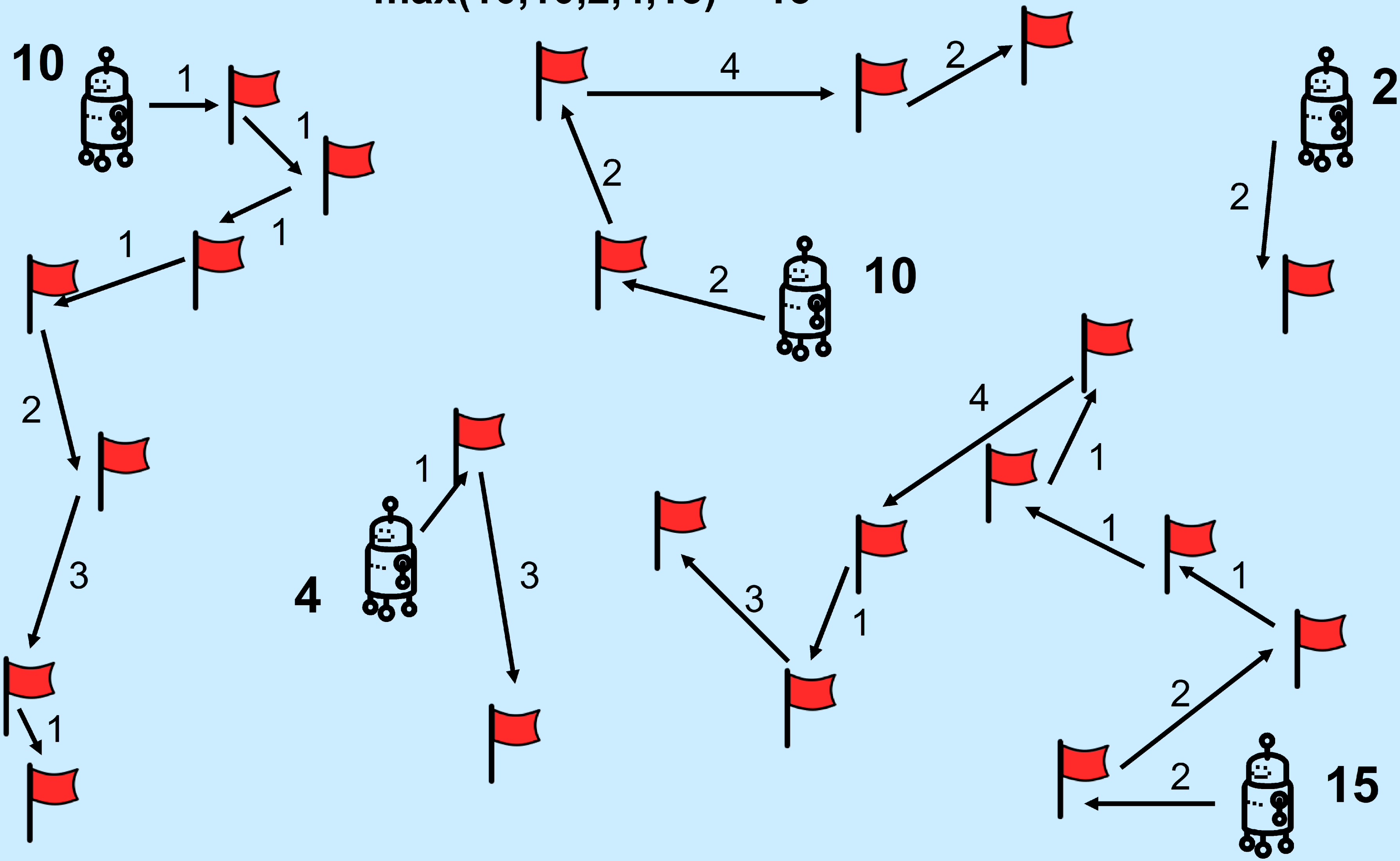
Pályalicit szabályok származtatása

MiniSum
Pályaköltségek összegének minimálása az összes ágensre nézve
Teljes energia, vagy távolság minimálása
Alkalmazás: bolygófelszín kutatása

MiniMax
Maximális pályaköltség minimálása az összes ágensre nézve
Teljes feladatvégzési idő minimálása (makespan)
Alkalmazás: objektum-felügyelet, bányatakarítás

MiniAve
Átlagos érkezési idő minimálása az összes célpontra nézve
Átlagos kiszolgálási idő minimálása (flowtime)
Alkalmazás: search and rescue

Tipikus koordinációs feladat: MiniMax Team célfüggvény
max(10,10,2,4,15) = 15

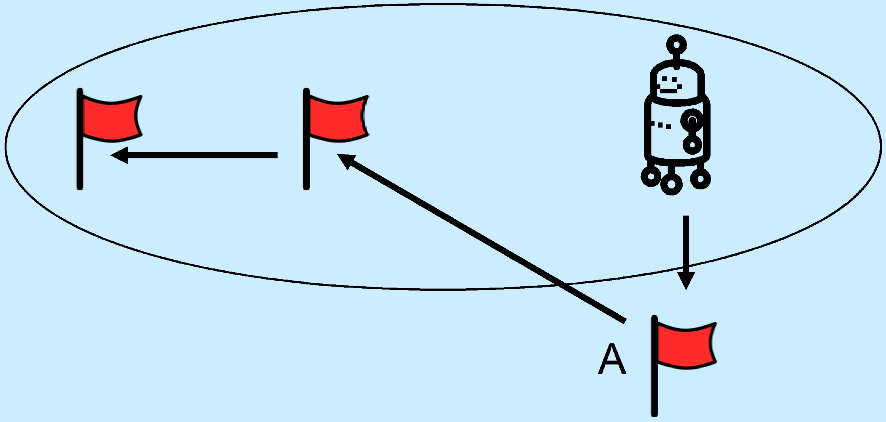


Szekvenciális aukció

Pályalicit szabályok származtatása

MiniMax = makespan

Licit a minimális pályaköltséggel az összes elnyert célpont meglátogatásához, feltéve, hogy a licitált célpontot elnyeri (BidMaxPath), ami az összes ágensre nézve egyenlíti ki a pályaköltségeket.



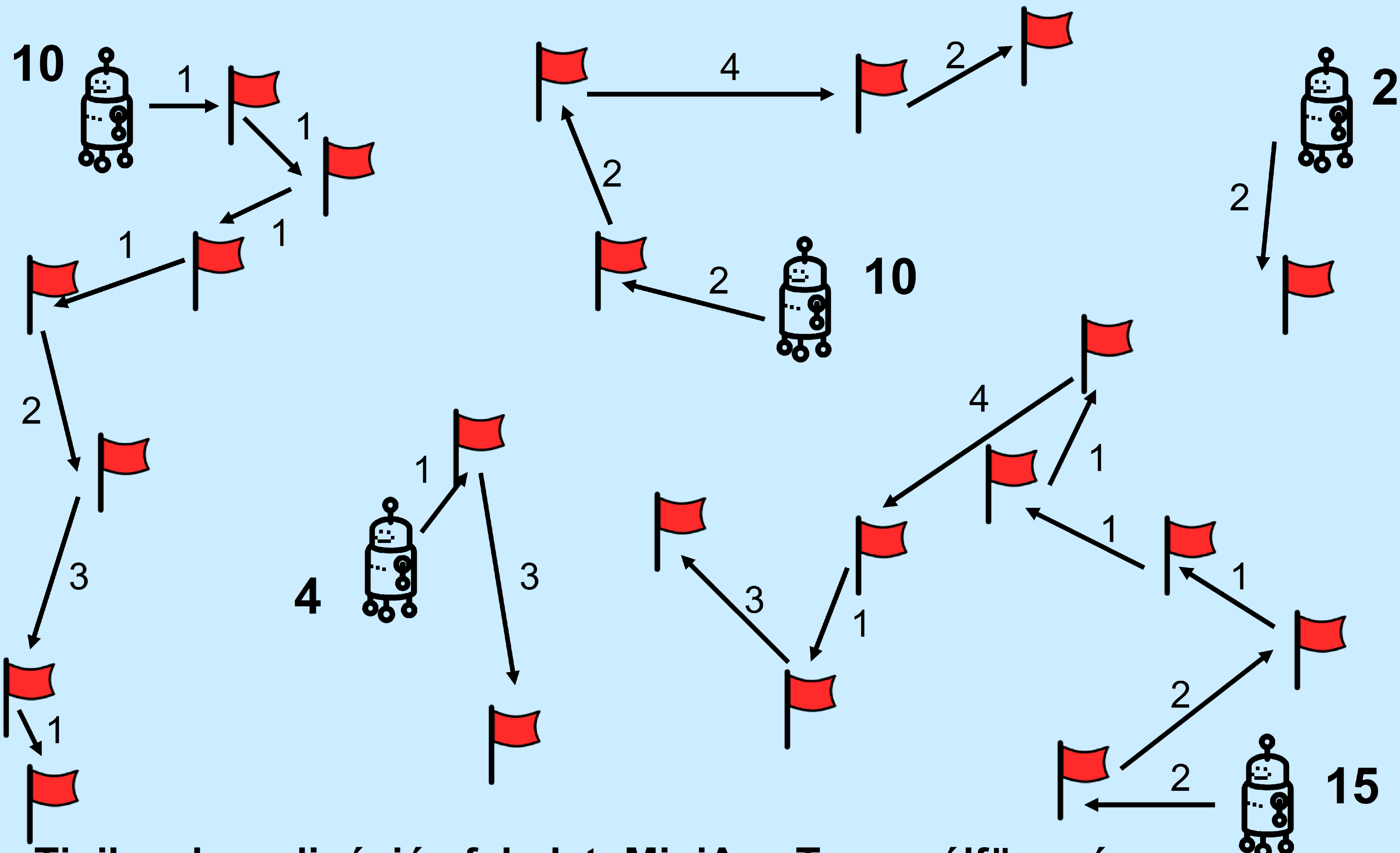
Szekvenciális aukció

Pályalicit szabályok származtatása

MiniSum
Pályaköltségek összegének minimálása az összes ágensre nézve
Teljes energia, vagy távolság minimálása
Alkalmazás: bolygófelszín kutatása

MiniMax
Maximális pályaköltség minimálása az összes ágensre nézve
Teljes feladatvégzési idő minimálása (makespan)
Alkalmazás: objektum-felügyelet, bányatakarítás

MiniAve
Átlagos érkezési idő minimálása az összes célpontra nézve
Átlagos kiszolgálási idő minimálása (flowtime)
Alkalmazás: search and rescue



Tipikus koordinációs feladat: MiniAve Team célfüggvény

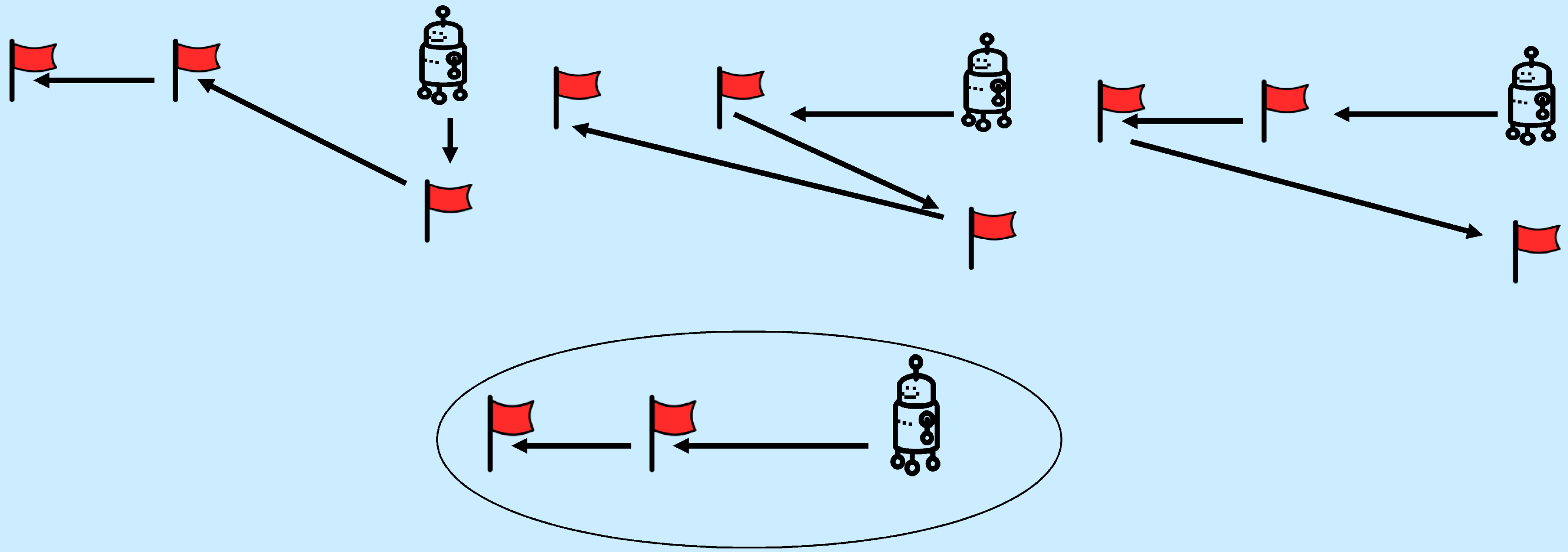
(1+2+3+4+6+9+10+1+4+…)/22 = 5.8

Szekvenciális aukció

Pályalicit szabályok származtatása

MiniAve = flowtime

Licit az érkezési idők minimális összegében jelentkező inkrementummal, ami az összes elnyert célpont meglátogatásához szükséges, feltéve, hogy a licitált célpontot elnyeri (BidAvePath).



Szekvenciális aukció

Pályalicit szabályok származtatása

Egy adott célponthalmaz meglátogatásához szükséges minimális költségű pálya megkeresése NP-nehéz. Polinomiális idejű legolcsóbb beszúrás (**cheapest insertion**) heurisztikát (vagy más bonyolultabb heurisztikát) használjuk.

Licitszabályok

- n db. ágens $r_1, \ldots, r_n$, m db. még nem kiosztott célpont: $t_1, \ldots, t_m$,
- az eddig kiosztott célok partíciója ágensenként: $T = (T_1, \ldots, T_n)$,
- team hatékonyságát optimalizáló $f(g(r_1,T_1), \ldots, g(r_n,T_n))$ függvény ágensek g(.) hatékonyságaik alapján,
- $PC(r_i,T_i)$ az i-edik ágens minimális pályaköltsége T_i particióban (Path Cost)
- $STC(r_i,T_i)$ minimális kummulatív célpontköltség T_i minden célpontjára (Sum per Target Cost)

$$\text{MiniSum} = \min_T \Sigma_j\, PC(r_j,T_j)$$

$$\text{MiniMax} = \min_T \max_j PC(r_j,T_j)$$

$$\text{MiniAve} = \min_T 1/m\, \Sigma_j\, STC(r_j,T_j)$$

Licitszabályok

MiniSum	energiafogyasztás
MiniMax	feladatvégzési idő
MiniAve	átlagos célpont elérési idő

Pálya: mindegyik kiszámítása NP-nehéz

Fa: MiniSum – min. kifeszítő erdő P,
MiniMax – minmax kif. erdő NP,
MiniAve – triv. erdő = csillagok,

pályakonverzió: shortcutting, heur. 2 x (jobb = nehezebb)

Minimális kifeszítő fa problémája

Adva nem irányított gráf G, d(u, v) élköltségekkel (távolságokkal). Találjuk meg azt a T részgráfot, ami az összes csomópontot összekapcsolja és a távolságok összegét minimalizálja. (Fa lesz az!)

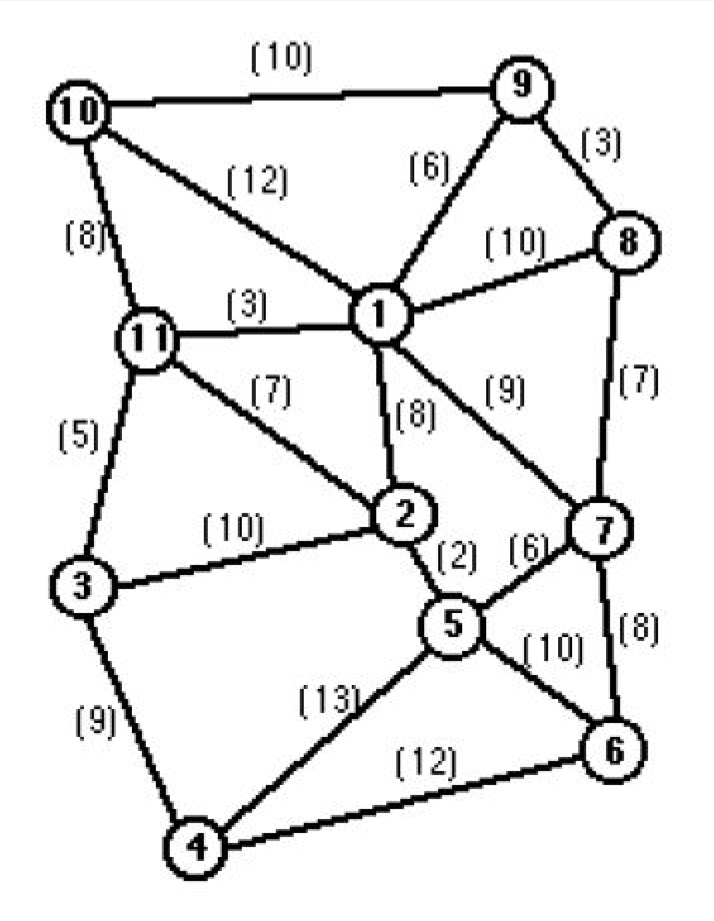
Algoritmusok (legyen minden élhossz eltérő):

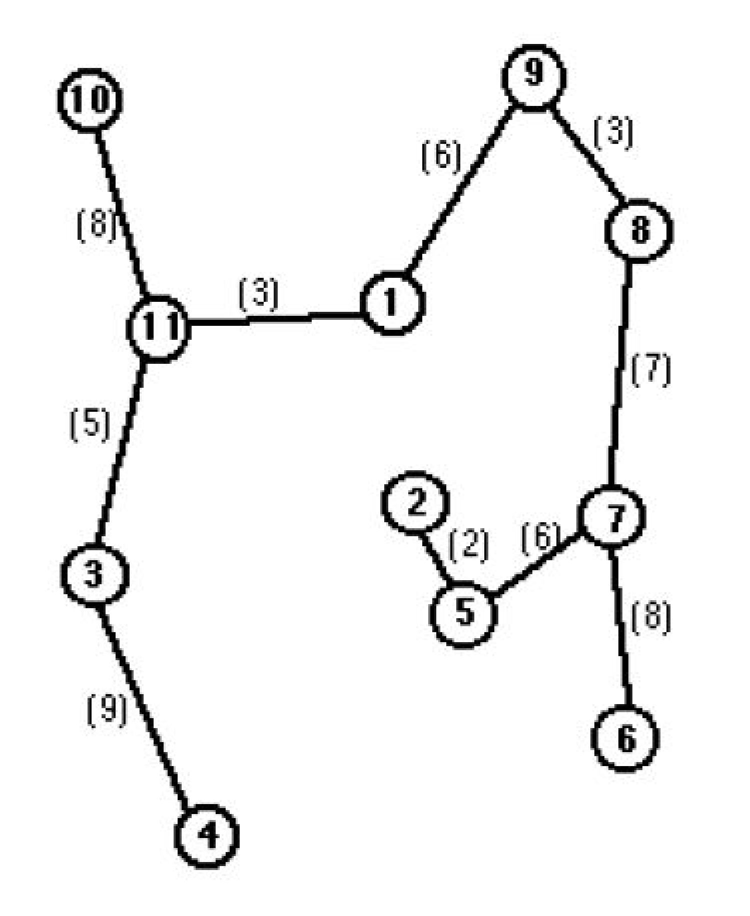
1. Adjunk hozzá éleket, növekvő sorrendben, kihagyva azokat, amelyek ciklusokhoz vezetnek (Kruskal algoritmus)

2. Tetszőleges gyökértől fát növeszteni, a legrövidebb perem-él hozzáadásával (Prim algoritmus)

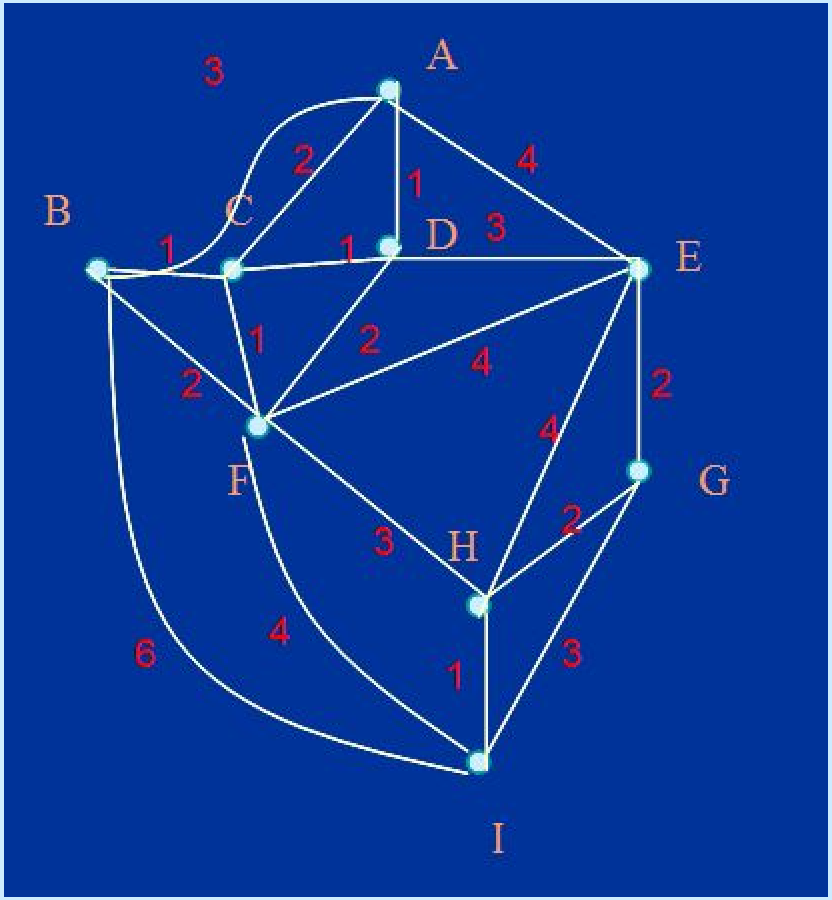
3. Kezdjünk az összes éllel, a csökkenő érték szerint eliminálva, kihagyva azokat, amelyek miatt a gráf darabokra esne szét („Reverse-Delete" algoritmus)

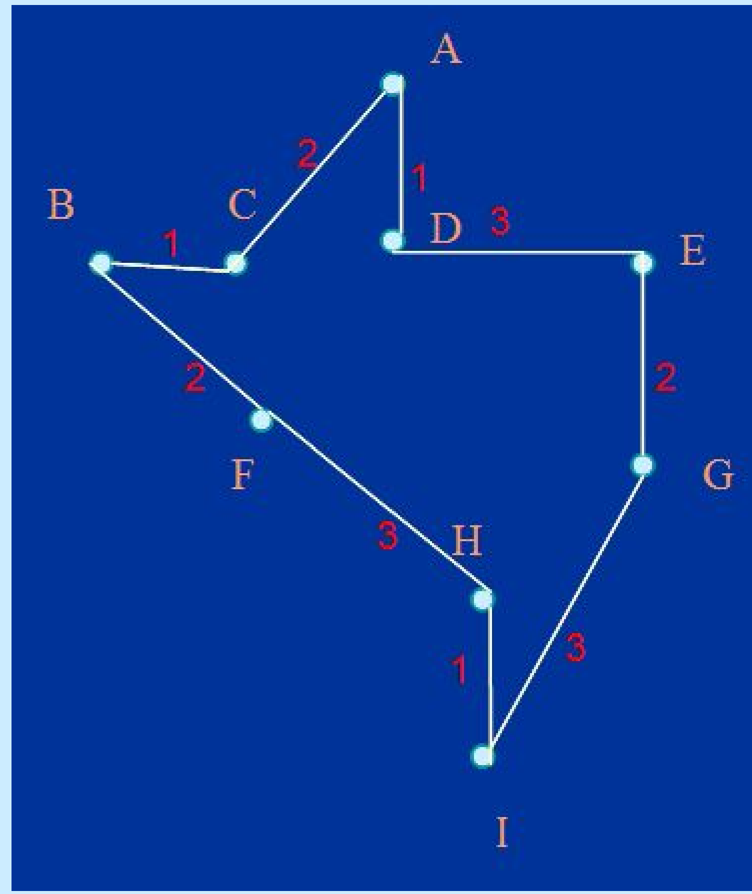
Minimális kifeszítő fa problémája





Utazó ügynök problémája





Utazó ügynök problémája – közelítő sémák

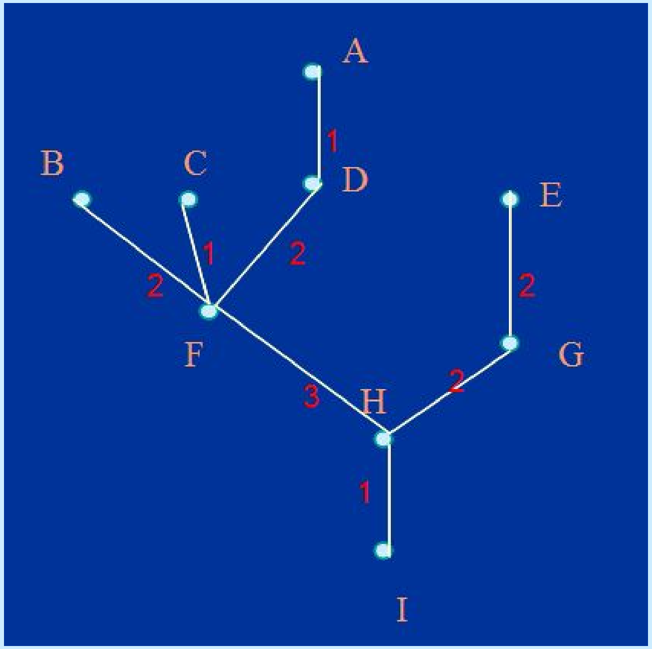
heuriszt. megoldás ≤ (1+ ε) minimális TSP pálya

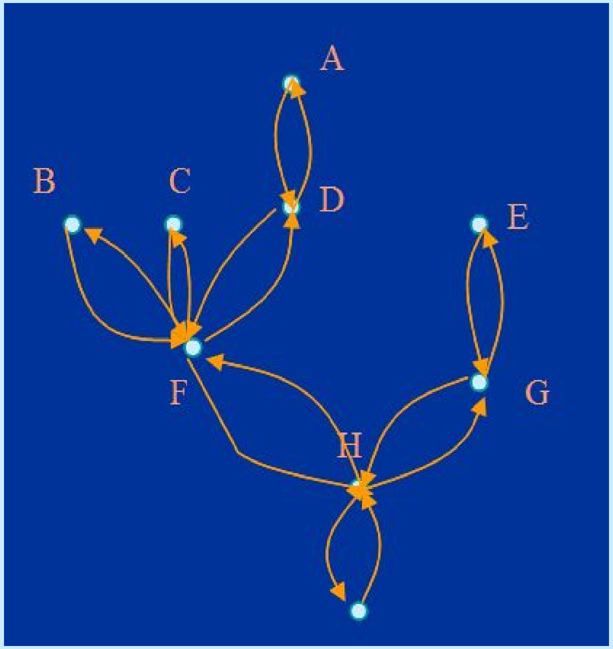
Szimm.költség + háromszög egyenlőtlenség

Egy él törlése az optimális megoldásból = kifeszítő fa

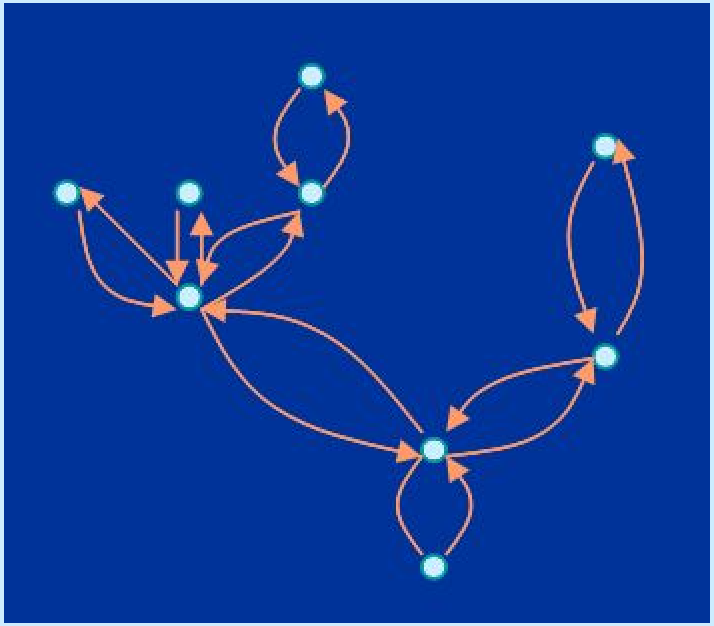
opt(TSP) >= kif.fa >= MKF

mélységi bejárás = 2 MKF =< 2 opt(TSP)



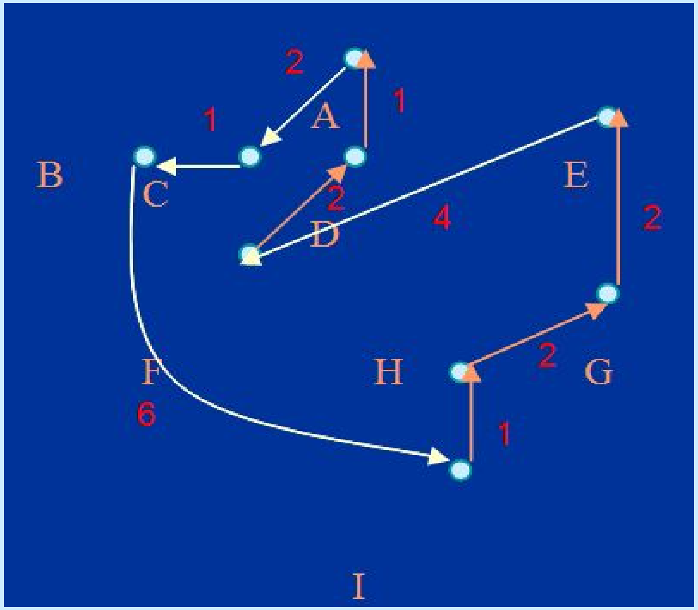


Utazó ügynök problémája – közelítő sémák



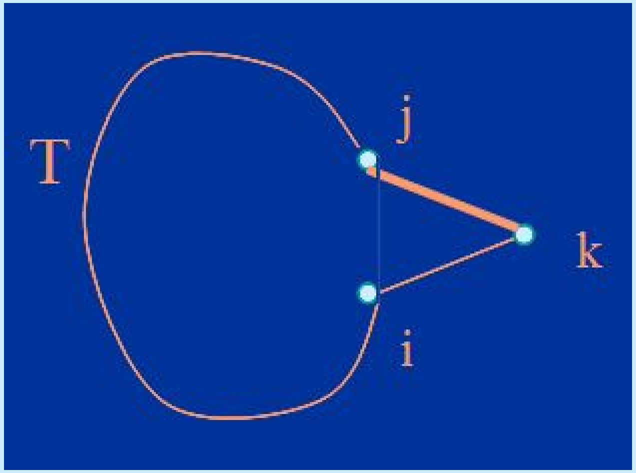
Shortcuts – mélységi bejárásból TSP pálya: vissza nem a szülőre, hanem az első további még nem meglátogatott ősre
MKF pálya =< mélységi bejárás (h.szög egnl.)

MKF pálya =< 2 opt(TSP)



Cheapest insertion

min (d(j,k) + d(i,k) – d(i,j))



(Szintén 2-opt)

Licitszabály

Ágens r_i licitál:

különbség = team hasznosság(hozzárendelés + ő győzelme)
- team hasznosság (kiinduló helyzet)

$f(g(r_1,T'_1), \ldots, g(r_n,T'_n)) - f(g(r_1,T_1), \ldots, g(r_n,T_n))$,

$T'_i = T_i \cup \{t\}$, $T'_j = T_j$ $i \neq j$ (hegymászó algoritmus, lokális optimum)

BidSumPath: $\Sigma_j PC(r_j,T'_j) - \Sigma_j PC(r_j,T_j) =$ $\mathbf{PC(r_i,T'_i) - PC(r_i,T_i)}$

BidMaxPath:

$\max_j PC(r_j,T'_j) - \max_j PC(r_j,T_j) = PC(r_i,T'_i) - \max_j PC(r_j,T_j) =$ $\mathbf{PC(r_i,T'_i)}$

($\max_j PC(r_j,T'_j) = PC(r_i,T'_i)$, különben korábban elkelne)

BidAvePath:

$1/m\ \Sigma_j STC(r_j,T'_j) - 1/m\ \Sigma_j STC(r_j,T_j) = 1/m\ (STC(r_i,T'_i) - STC(r_i,T_i))$

$=$ $\mathbf{STC(r_i,T'_i) - STC(r_i,T_i)}$

(fa licitek: $FC(r_i,T_i)$ – min. kifeszítő fa költsége

$SFC(r_i,T_i)$ – kum. célfa költsége (gyökér-cél költségek összege)

– tovább egyszerűsíthetők, polinomiális komplexitás, …

Pályák v. fák

Direkt indirekt (fa-pálya konverzió) Fák számításbarát (polinomiális)

MiniSum	minimális kifeszítő erdő	polinomiális
MiniMax	minimax kifeszítő erdő	NP
MiniAve	minimum átlagos költségű kifeszítő erdő - triviális (csillagok)	

Licitszabályok összehasonlítása

BidSumPath, BidMaxPath, BidAvePath
Számítás: lokális
Optimális licit: NP-nehéz
Optimális konverzió: nem kell
Gyakorlat: egyszerű TSP beszúrási heurisztika

BidSumTree, BidMaxTree, BidAveTree
Számitás: lokális.
Optimális licit: polinomiális
Optimális konverzió: NP-nehéz
Gyakorlat: egyszerű MST (minimális kifeszítő fa) heurisztika

Szekvenciális aukció: Elméleti analízis

3 team célkitűzés MiniSum, MiniMax, MiniAve

6 licitszabály

3 pálya licitszabály
BidSumPath, BidMaxPath, BidAvePath
3 fa licitszabály
BidSumTree, BidMaxTree, BidAveTree

team hatékonyság: **18 alsó, felső korlát**
worst-case költségarány
optimális költséghez viszonyítva
első elméleti garanciák aukció alapú koordinálásban

Szekvenciális aukció

Analitikus eredmények *n* ágens *és m* célpont költségarány = team költség licitből / minimális team költség

Bidding Rule	Team Objective					
	MiniSum		MiniMax		MiniAve	
	Lower	Upper	Lower	Upper	Lower	Upper
BidSumPath	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxPath	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAvePath	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$2m^2$
BidSumTree	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BidMaxTree	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BidAveTree	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

Szekvenciális aukció

Pálya licitek

2 ágens, 10 nem klaszter alakú célpont

ismert terep 51 x 51

	SUM	MAX	AVE
BidSumPath	193.50	168.50	79.21
BidMaxPath	219.15	125.84	61.39
BidAvePath	219.16	128.45	59.12
optimális (MIP) = ideális kombinatorikus aukció	189.15	109.34	55.45

Szekvenciális aukció

Pálya licitek

2 ágens, 10 klaszter alakú célpont

ismert terep 51 x 51

	SUM	MAX	AVE
BidSumPath	134.18	97.17	62.47
BidMaxPath	144.84	90.10	57.38
BidAvePath	157.29	100.56	49.15
optimális (MIP) = ideális kombinatorikus aukció	132.06	85.86	47.63