

Hálózati Algoritmusok

Öntelepítő szenzor hálózatok fókuszált lefedéshez

Mobil Szenzor Hálózatok

- Szenzor csomópontok, melyek képesek
 - Érzékelni
 - Egyszerű számítást végezni
 - Kommunikálni korlátos hatótávolságon belül
 - Mozogni külső segítség nélkül

Lokális algoritmusok

- Lokális algoritmus: minden szenzor önállóan hoz döntést a lépésekben
 - függetlenül,
 - csak k -hop szomszédokról van információ, ahol k konstans.

Lefedés

- Lefedés alapvető követelmény
 - Adott régió lefedése (Region of Interest, ROI)
 - Határ/akadály lefedés (Barrier Coverage)
 - Fokuszált lefedés egy „Point of Interest” (POI) körül

- Felhasználási területek
 - Környezet monitorozás
 - Felügyelet
 - “Smart spaces”



Remote Monitoring



Sustainability



Industrial Measurements



Air/Climate Water/Soil Indoor Monitoring



Power Quality and Consumption Monitoring Solar Monitoring Wind Farm Health



Structural Health Monitoring Machine Condition Monitoring Process Monitoring

Fokuszált lefedés

Egy stratégiai pont, “Point of Interest” (POI) körüli terület monitorozása

- Vegyi üzem:
 - Távolság függő szennyezési hatás
- Katasztrófa epicentruma:
 - Túlélők keresése
- Központ / „főhadiszállás”
 - Betolakodási esemény felderítése

Fokuszált lefedés

Lefedendő régió

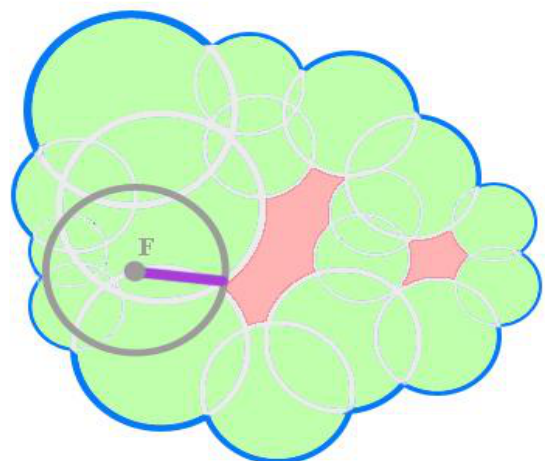
Vakfolt / érzékelési lyuk

„Point of Interest” (POI): F

Lefedési sugár:

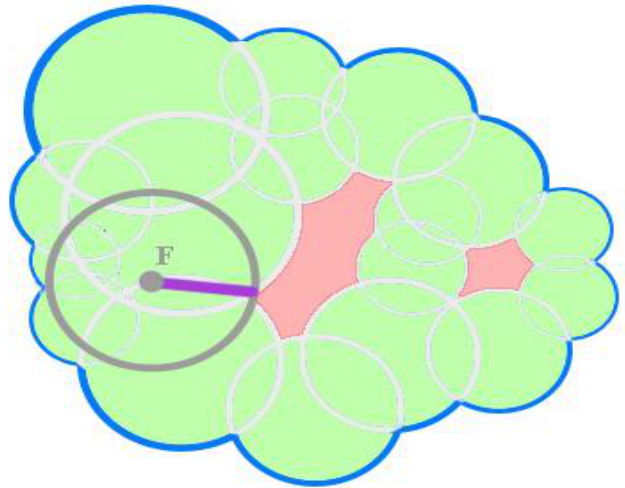
maximumális sugárú, vakfolt mentes lefedett körlap sugara

Cél: maximalizálni a lefedési sugarat



Fokuszált Lefedés

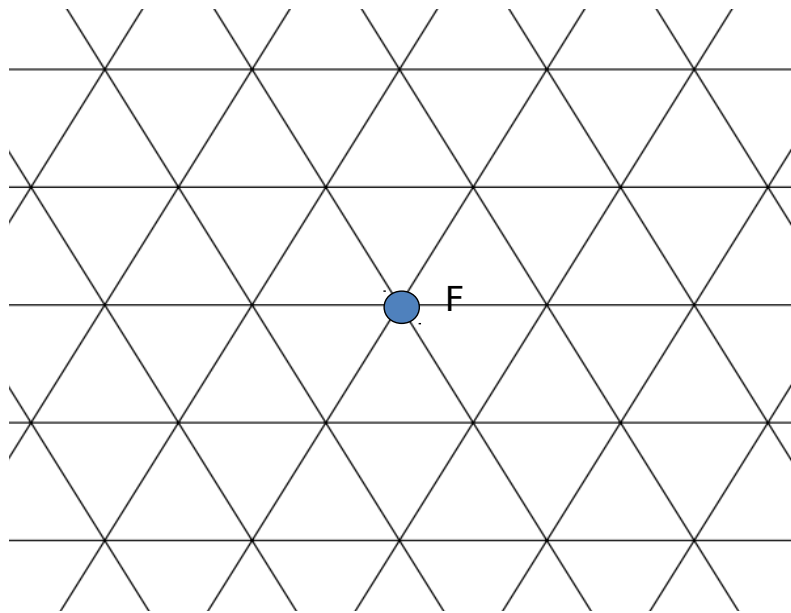
- Megoldható a Fokuszált lefedési probléma lokális algoritmussal?
 - Igen [Li et al. '09, '11]
 - 1-hop szomszédokról használ információt
 - Véges idő alatt



Modell

- Szensorok közös koordináta rendszert használnak
- Minden szenzor ismeri a saját koordinátáját
- Minden szenzor ismeri a POI koordinátáját
 - w.l.o.g. the POI is at the origin
- r_s : érzékelési sugár
- r_c : kommunikációs sugár
$$\sqrt{3} * r_s \leq r_c$$
- (Aszinkron) Look-Compute-Move modell

Equilateral Triangle Tessellation (TT)

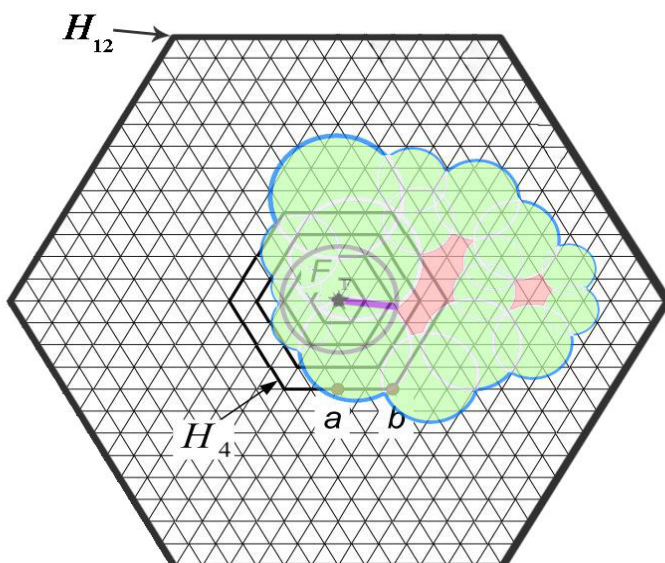


Adott r_s érzékelés sugár esetén TT maximalizálja a vakfolt mentesen lefedett területet

TT garantálja hogy a kommunikációs hálózat összefüggő, ha

$$\sqrt{3} * r_s \cdot r_c$$

G_{TT} planár gráf



r_s : érzékelő sugár
 r_c : él hossza
(kommunikációs sugár)

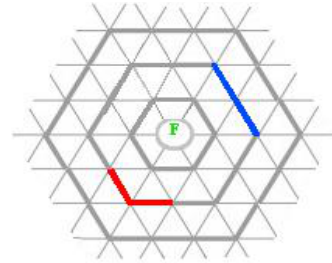
$$\sqrt{3} * r_s = r_c$$

H - polygon

6-szög, koncentrikusan F körül

minden $v (\neq F)$ csúcsra a szomszédsági minták:

1. él
2. konvex



n : szenzorok száma, i : hop-réteg távolság F-től

$u_H(i)$: #rácspontok az i -edik rétegegig

$$u_H(i) = 1 + 6 + 2 \cdot 6 + \dots + i \cdot 6 = 3i(i+1) + 1$$

$$y_H(n) = \max(i \mid u_H(i) \leq n)$$

Algoritmusok

Greedy Advance

Greedy-Rotation-Greedy

egyszerű hop szelekciós szabályok

1-hop távolság

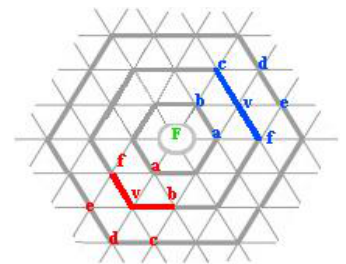
hexagonal F-coverage-hez H-polygon szükséges

Greedy Advance

greedy advance: minden lépéssel egy réteggel beljebb $\rightarrow F$, amíg csak tud

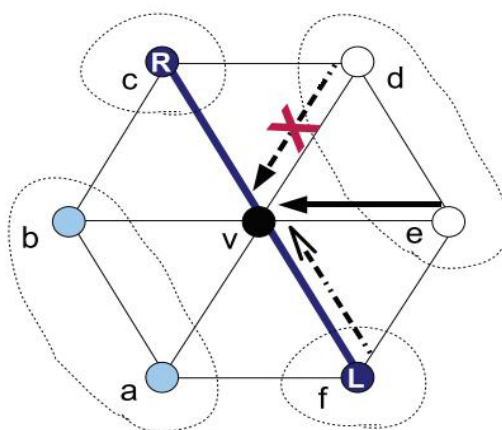
- Holtverseny:
 - Random
- Ütközések elkerülésére:
 - “priority”-szabály
 - “forbiddance”-szabály
 - “innermost-layer”-szabály
- Alaphelyzetre:
 - “alignment”-szabály
- Ütközés megoldására:
 - “retreat”-szabály

Greedy Advance szabályok

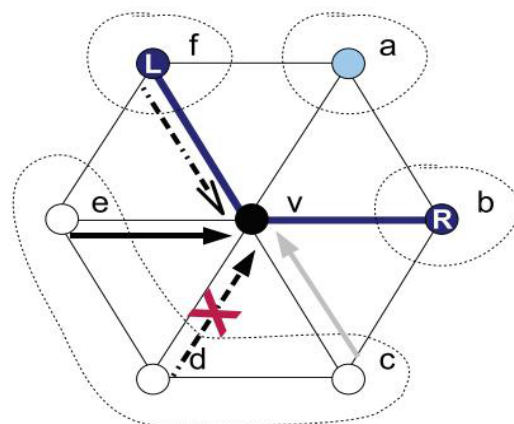


prioritás: $e, d \rightarrow v$

Külső balszélső mindig magasabb prioritású (itt: e)



Edge (—) ●

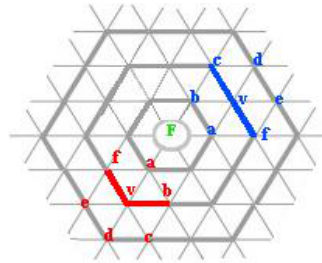
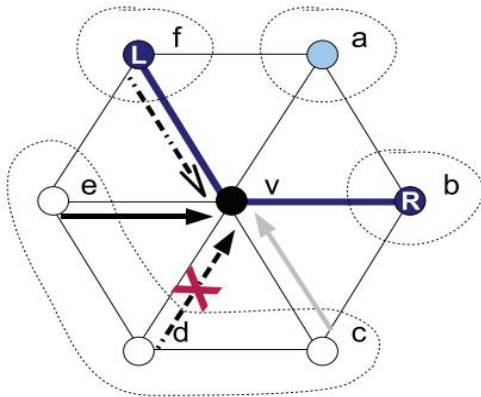


Convex corner (\wedge) ●

Greedy Advance szabályok

forbiddance: e, c -> v

Sarokra külső jobbszélről nem mehetünk (itt: c)

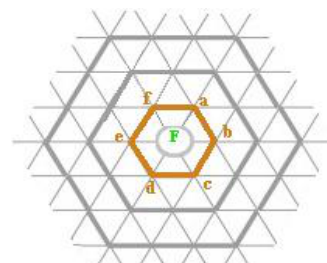
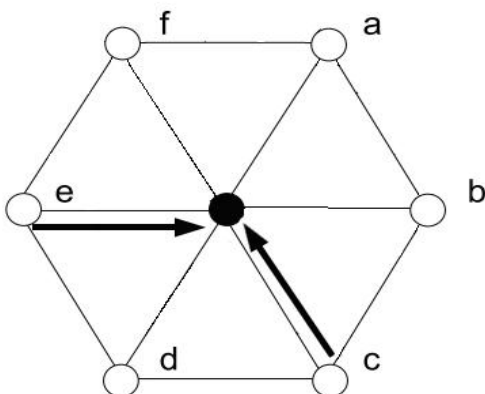


Convex corner (\wedge) ●

Greedy Advance szabályok

innermost-layer:

F elfoglalható bármelyik külső szomszédja által mindaddig, amíg úgy tudják, hogy szabad

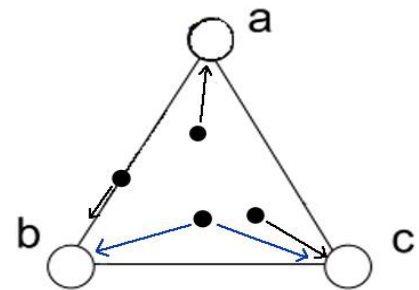


Greedy Advance szabályok

alignment:

3-szögön vagy 3-szögben lévő csomópont azt a csúcsot választja, amelyik a legkevesebb csomópontot tartalmazza és azok közül a legközelebbit.

Holtverseny: random

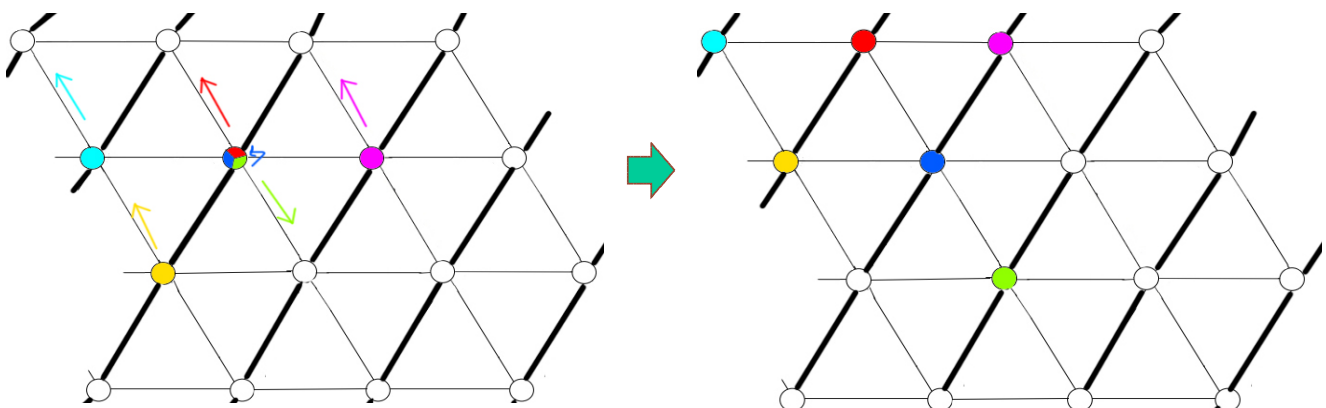
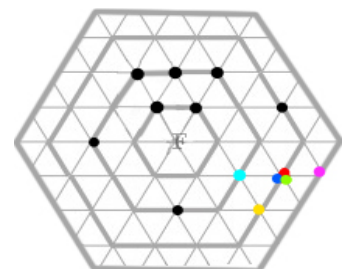


Greedy Advance mozgás

új mozgás **retreat**:

középponttól eggyel távolabbi körbe található legkevesebb csomópontot tartalmazó csúcsba távozik

Holtverseny: random



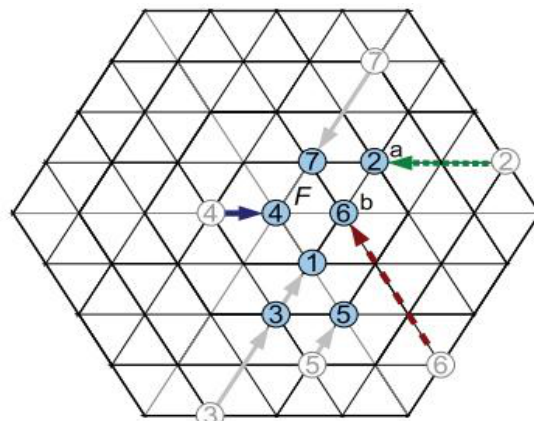
Greedy Advance szabályok

retreat: csúcsütközés esetén lokális rangsoroló folyamat: mind csúcs egy ranggal rendelkezik.

Legmagasabb rangú dönt először, majd csökkenő sorrendben többi

Ha az i . rangú úgy dönt, hogy helyben marad, az utána jövő alacsonyabb rangúak retreat mozgást hajtanak végre

Greedy Advance példa



Greedy-Rotation-Greedy

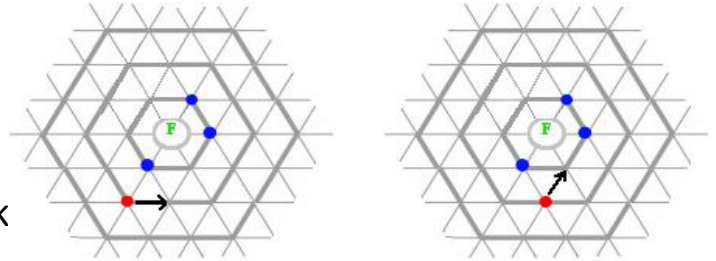
greedy advance + új mozgás **rotation**:

hatszög mentén órajárással ellentétes irányba

megállás:

1. greedy
2. forgás kezdőhelye

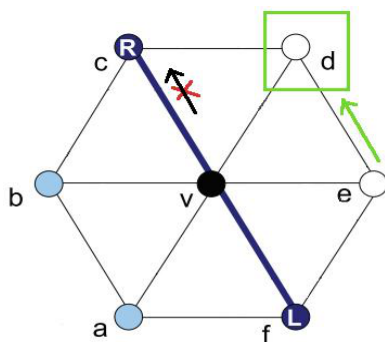
ha a forgási célcsúcs foglalttá válik
kezdőhely értéke null-ra állítódik



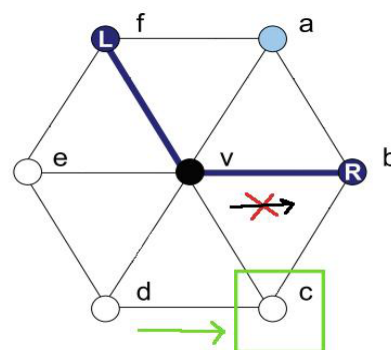
Greedy-Rotation-Greedy szabályok

suspension:

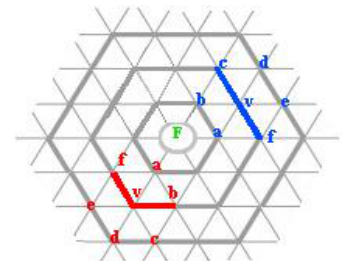
v nem forog, ha valaki a külső jobbszélő szomszédjába (zölddel jelölt téglalapba) forog



Edge (—) ●



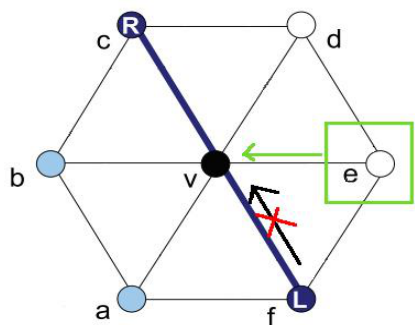
Convex corner (∧) ●



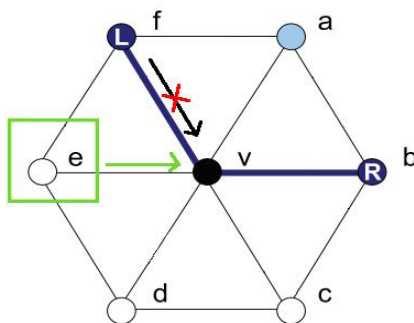
Greedy-Rotation-Greedy szabályok

competition: $e, f \rightarrow v$

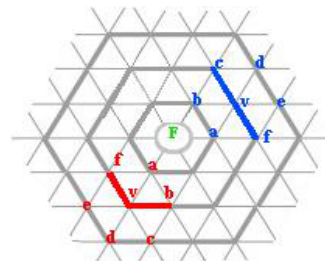
Bal szomszéd (f) és külső balszélső szomszéd (e) esetén külső balszélső nyer (e)



Edge (—) ●



Convex corner (∧) ●



GRG/CW Collision allowWance

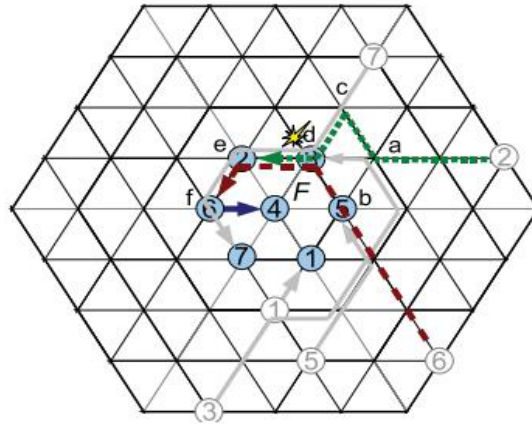
nincs további megkötés

greedy-rotation ütközés esetén retreat szabály

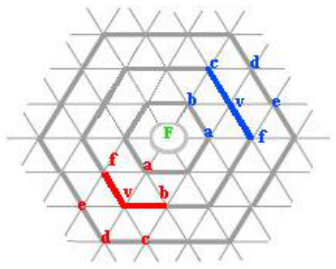


forgó csomópont mindig a legnagyobb prioritáú a lokális rangsorolásban

GRG/CW példa

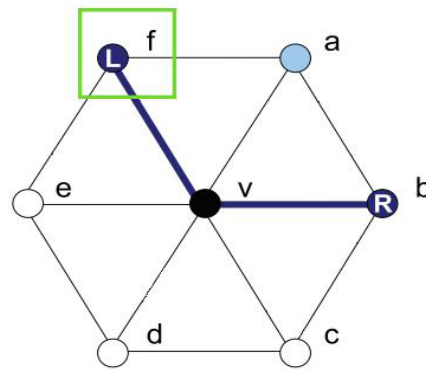
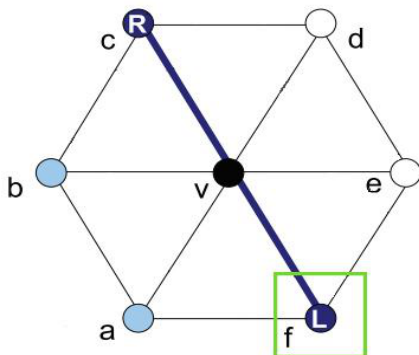


GRG/CV szabályok



safety: -> v

csak akkor választja v-t, ha szomszédja a v csúcs balszomszédjának (zöld négyzet)



Edge (—) ●

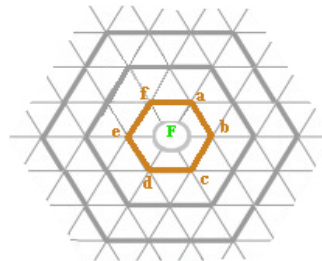
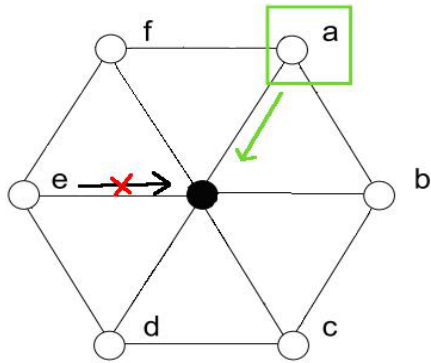
Convex corner (∧) ●

GRG/CV szabályok

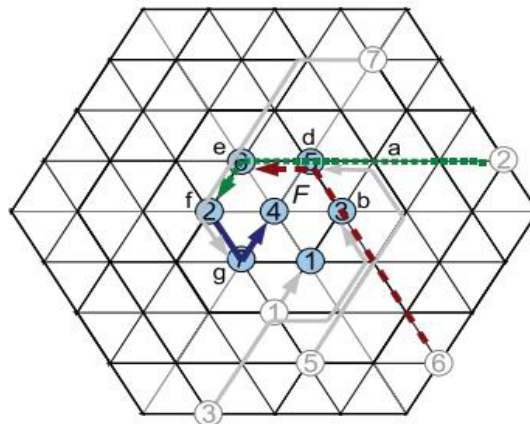
gateway:

előre definiált átjáró (itt ez legyen: a)

többiből nem engedélyezett a greedy, csak rotation



GRG/CV példa



Elemzés

GA

- garantálja, hogy az F csúcs el lesz foglalva véges időn belül
- véges időn belül terminál
- hálózati lyuk nélküli lefedettséget eredményez

GRG

- garantálja, hogy az F csúcs el lesz foglalva véges időn belül
- P_0, \dots, P_{i-1} teljesen el van foglalva csomópont ütközések nélkül, $n \geq u(i)$.
 P_i teljesen el lesz foglalva csomópont ütközések nélkül véges időn belül
- P_0, \dots, P_{i-1} teljesen el van foglalva csomópont ütközések nélkül,
 $u(i-1) < n < u(i)$. P_i -n lévő csomópontok megállnak véges időn belül
- véges időn belül terminál
- érzékelési lyuk nélküli lefedettséget eredményez
- (közel) optimális F-lefedés

Teljesítmény értékelés

3 nézőpont:

konvergencia idő

n_T : a stabilizált hálózat kialakulásához szükséges időegységek száma

energia felhasználás

n_M : lépések száma

M_g : hányszor indult el a csomópont

P_g : teljes megtett táv

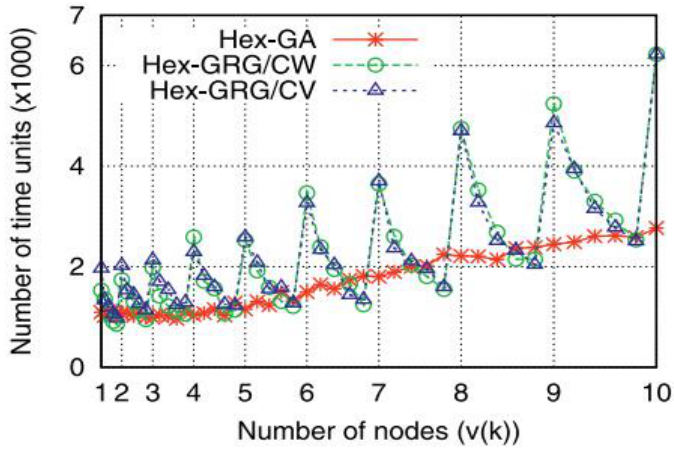
D_{ini}, D_{fin} : kezdőpont, végpont euklideszi távolsága F-től

$P_g = |D_{ini} - D_{fin}|$

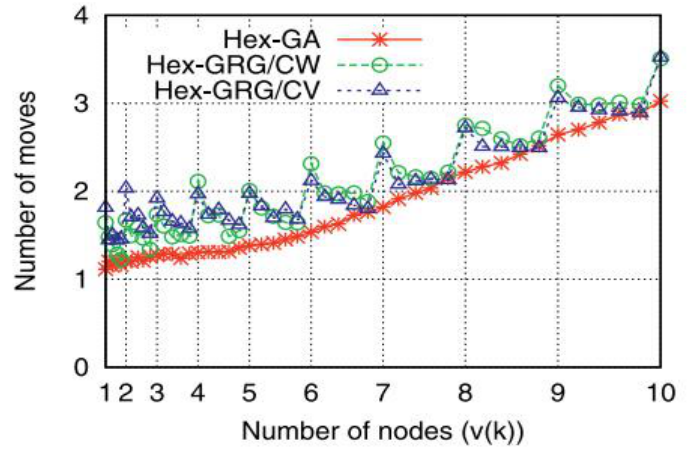
MoP: megadja milyen költséges egy cikcakkba haladó csomópont

$MoP = M_g/P_g$

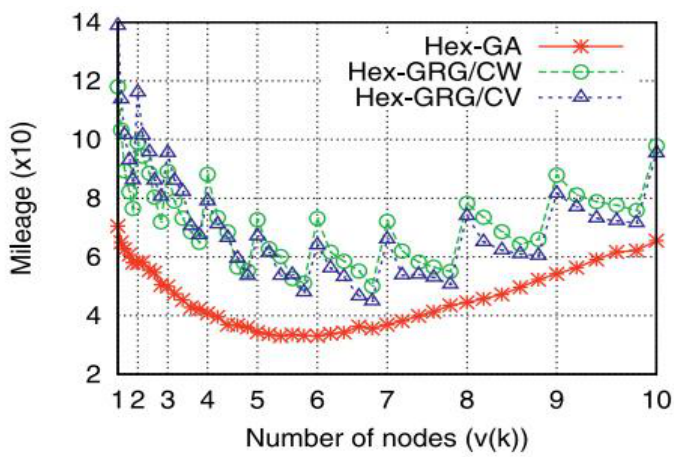
csomópont ütközés (n_C)



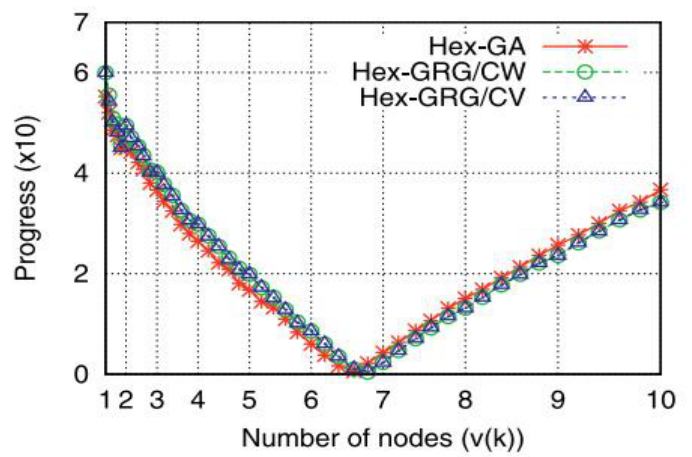
(a)



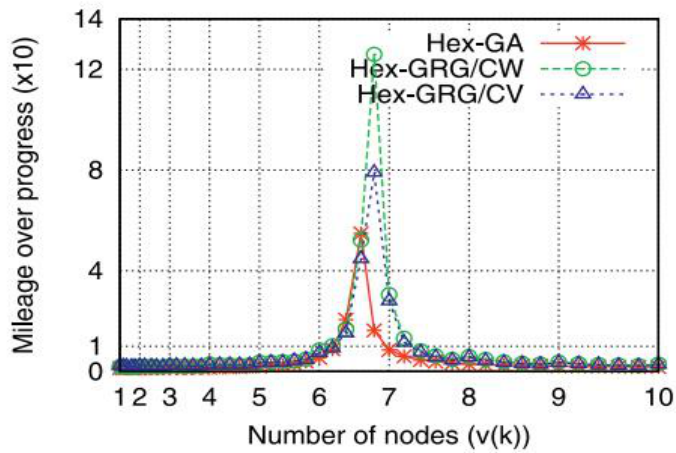
(b)



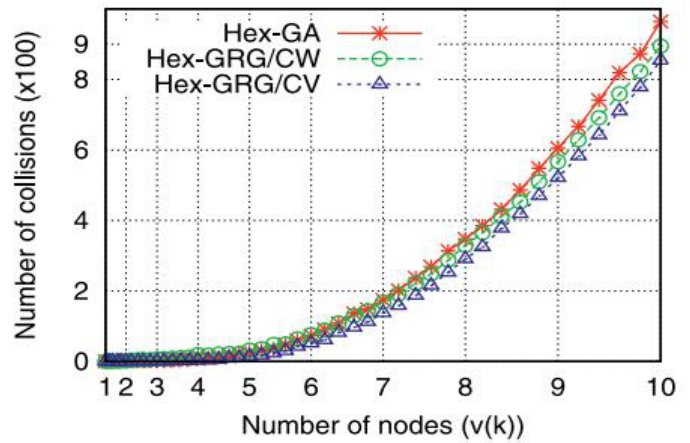
(c)



(d)



(e)



(f)

Irodalom

- Xu Li, Hannes Frey, Nicola Santoro, Ivan Stojmenovic: Strictly Localized Sensor Self-Deployment for Optimal Focused Coverage. IEEE Trans. Mob. Comput. 10(11): 1520-1533 (2011)