**Kényszer-kielégítési problémák – Összefoglaló**

1. **Különbség az általános keresési probléma és a kényszer-kielégítési probléma között  
   Az általános keresési probléma esetén az állapot egy „fekete doboz”, amit bármilyen adatstruktúra képviselhet. A kereséshez elegendő az állapotátmeneteket, a heurisztikát és a céltesztet ismerni.  
   A kényszer-kielégítési probléma (CSP) esetén az állapot a változók értékeinek halmaza. Minden változó XiX\_iXi​ a saját tartományából DiD\_iDi​ vesz fel értéket. A céltesztet kényszerek halmaza határozza meg, amelyek meghatározzák, hogy mely értékkombinációk engedélyezettek.**
2. **A kényszer-kielégítési feladat megadása  
   Egy CSP három halmazzal adható meg:  
   (X, D, C)  
   ahol**

* **X = {X₁, X₂, ..., Xₙ}: változók halmaza**
* **D = {D₁, D₂, ..., Dₙ}: a változók tartományai**
* **C: a kényszerek halmaza**

**Példa: térképszínezés  
Változók: WA, NT, Q, NSW, V, SA, T  
Tartomány: {piros, zöld, kék}  
Kényszerek: szomszédos tartományok különböző színűek.**

1. **Kemény és puha kényszerek**

* **Kemény (abszolút) kényszerek: kötelezően teljesülniük kell (pl. WA ≠ NT).**
* **Puha (preferencia) kényszerek: megszeghetők, de költség vagy preferencia jár hozzájuk (pl. „a piros jobb mint a zöld”).  
  Puha kényszereknél a cél az összköltség minimalizálása → optimalizációs probléma.**

1. **A kényszer-kielégítési probléma megoldása**

* **Standard keresés: mélységi keresés, minden lépésben egy új változót kap értéket.**
* **Visszalépéses keresés (Backtracking): alapvető módszer CSP-kre – ha ellentmondás adódik, visszalép az előző változóhoz.**
* **Fejlettebb megoldási technikák:**
  + **Heurisztikák (változó- és értékválasztás)**
  + **Előrenéző ellenőrzés**
  + **Kényszerek terjesztése (pl. élkonzisztencia)**
  + **Lokális keresés (minimális konfliktus módszer).**

1. **Kényszerek típusai**

* **Unáris kényszer: egy változóra vonatkozik (pl. SA ≠ zöld).**
* **Bináris kényszer: két változóra vonatkozik (pl. SA ≠ WA).**
* **Magasabb rendű kényszer: három vagy több változóra vonatkozik (pl. betűrejtvény).**
* **Puha kényszerek: preferenciákat vagy súlyokat tartalmaznak.**

1. **Kényszergráf  
   A kényszergráf a bináris CSP-k vizuális ábrázolása:**

* **Csúcsok → változók**
* **Élek → kényszerek a változók között  
  A gráf szerkezetének vizsgálatával a keresés gyorsítható.  
  Ha a gráf több összefüggő komponensből áll, azok függetlenül megoldhatók (pl. Tazmánia külön részprobléma).**

1. **Visszalépéses keresés működése  
   A visszalépéses keresés mélységi keresés, ahol minden szinten egy új változónak rendelünk értéket.  
   Ha az érték ütközik valamely korábbi kényszerrel, visszalépünk (backtrack).  
   A változók hozzárendelése kommutatív, ezért a keresési fa mélysége a változók számával egyezik meg.**
2. **Heurisztikák a visszalépéses keresés gyorsítására**

* **MRV (Minimum Remaining Values): válaszd azt a változót, amelynek a legkevesebb megengedett értéke maradt (legkorlátozottabb változó).**
* **Fokszám heurisztika: ha több MRV azonos, válaszd azt, amely a legtöbb kényszerben szerepel.**
* **Legkevésbé korlátozó érték: válaszd azt az értéket, amely a legkevesebb értéket zárja ki a többi változó számára.**

**Ezek a heurisztikák jelentősen csökkentik a keresési tér méretét.**

1. **Előrenéző ellenőrzés (Forward Checking) és kényszerek terjesztése (Constraint Propagation)**

* **Előrenéző ellenőrzés: nyilvántartja a még értéket nem kapott változók lehetséges értékeit, és leáll, ha valamelyik változónak már nincs engedélyezett értéke. Csak egy lépéssel néz előre.**
* **Kényszerek terjesztése: a kényszereket lokálisan érvényesíti, több lépést előre következtet, így korábban felismeri a zsákutcákat.**

1. **Élkonzisztencia (Arc Consistency)  
   Egy él Xᵢ → Xⱼ konzisztens, ha minden x ∈ Dᵢ értékhez létezik olyan y ∈ Dⱼ, ami kielégíti a kényszert.  
   Ha nincs ilyen y, x törölhető Dᵢ-ből.  
   Az AC-3 algoritmus iteratívan biztosítja az élkonzisztenciát: ha egy tartomány változik, újraellenőrzi a szomszédokat.  
   Időbonyolultság: O(n²d³), AC-4 esetén O(n²d²).  
   Korlát: csak bináris kényszerek esetén működik, és a konzisztencia ellenőrzése NP-nehéz.**