

Egészértékű programozási szimulációk a magyar felvételi rendszerben ponthatáregyeztésekkel

Kováts Endre József

Témavezető: Bíró Péter

Dolgozatom egy projekt részeként született, aminek hosszútávú célja egy módszer megtalálása ami reális időn belül (percek) el tudja dönteni kit melyik egyetemre vegyenek fel, mindezt a felvételi rendszer minden speciális tulajdonságával és heurisztikák nélkül. A felvételi rendszer speciális tulajdonságai a következők:

- ponthatáregyeztés (az egyetem preferenciái nem szigorúak)
- közös kvóták (a kvóták a szakokra nem diszjunktak)
- alsó kvóták
- szakpárok

Az egyetemi felvételi problémát egy stabil párosítás megtalálásaként értelmezzük (egy párosítást akkor tekintünk stabilnak ha senkinek nincs jogos kifogása az eredményeivel kapcsolatban és a kvóták lehető legjobban fel vannak töltve). A speciális tulajdonságok közül a ponthatár egyezésen kívül bármelyik önmagában is NP-nehézzé teszi ilyen párosítások megtalálását, egyszerre több tulajdonság figyelembe vétele pedig csak tovább növeli a komplexitást. Dolgozatomban bemutatok egészértékű programozási technikákat amivel igyekszünk olyan modelleket alkotni, amelyek jól teljesítenek gyakorlatban.

Ezek a technikák a

1. kvóták mellett ponthatárok bevezetése
2. ponthatárok bináris változóval való modellezése
3. stabilitás biztosítása célfüggvény megfelelő választásával
4. stabilitás biztosítása szabad célfüggvénnyel, extra bináris változók bevezetésével

A dolgozatomban bemutatott modellek ezen technikákon és kombinációikon alapulnak, valamint a felvételi probléma alap illetve ponthatár-egyeztés verzióját modellezzük. Futási eredményeimet AMPL modellező és Gurobi IP-solver segítségével kaptam.

Eredményeim azt mutatják hogy ezen technikák közül mind javít a futási időn, azonban leghatékonyabb a 4. technika. Ez sajnos a legnehezebben modellezhető is, mivel már az alap feladat (szigorú preferenciákkal) és a ponthatár egyezéssel verzió esetében is különbözőképpen modelleztük őket.

A különböző technikák együtt is alkalmazhatóak, azonban tesztjeim kimutatták, hogy ez nem mindig vezet gyorsabb megoldáshoz. Bemutatom hogy ilyen esetben előfordulhat hogy a modell túl nagyra válik és egyszerűen nem tud lefutni a megoldó program, mert kifogy a számítógép a memóriából.

Fő konklúzióm, hogy érdemes egészítékű programozással közelíteni a problémát, mert mindkét vizsgált verziójának a problémára találtunk modellt ami 5 perc alatt lefut a 2008as 300 000 jelentkezésre. Továbbá, a komplexebb verziók modellezéséhez érdemes lehet használni a 2.-es és 4.-es technikát, mert szignifikánsan csökkenthetik a futási időt.