Válasz Deák István opponensi bírálatára

Köszönöm Deák István részletes és alapos értékelését és támogatását.

A Bíráló tanácsát megfogadva hozzákezdtem egy tankönyv írásához az értekezés alapján. Mostanáig két fejezet átdolgozásával végeztem. A javasolt bővítések mellett gyakorló feladatokat is kidolgoztam, és projekt-javaslatokat, amelyeket követve az érdeklődő olvasó maga is megtapasztalhatja a különböző eljárások tulajdonságait.

A Bíráló hiányolja a számítástechnikai eredmények részletezését. A készülő tankönyvben tárgyalni tervezem ezeket – éppen a módszertani és számítási nézőpont lesz a könyv specialitása. Az értekezésből hiányzó leírásokat az alábbiakban szeretném pótolni.

A 2. fejezetben említett számítási tapasztalatok sztochasztikus programozási feladatokon folytatott kísérleteken alapszanak. Ezeket a 4. és a 6. fejezetben ismertetem és a kiegészítő adatokat alább idézem.

A 3. fejezetben említett számítási tapasztalatok a Brunel Egyetemi kutatócsoporttal végzett kísérleteinken alapszanak. Portfolió optimalizálási feladatokat oldottunk meg másodrendű sztochasztikus dominancia feltétel mellett. A választható befektetési lehetőségek száma 76 volt. A véletlen hozamokat forgatókönyvekkel írtuk le. Ezeket véletlenszerűen generáltuk, lognormális eloszlást feltételezve. (Az eloszlás paramétereit a korábbi 10 évben tapasztalt hozam adatok alapján határoztuk meg.) Kísérleteinkben a forgatókönyvek S száma néhány száztól tízezerig növekedett.

A szokásos eljárás nagyméretű lineáris programozási feladatként fogalmazta meg a problémát. Ebben mind a döntési változók, mind a korlátozó feltételek száma S^2 nagyságrendű. Az általunk alkalmazott vágósíkos modellben a döntési változók száma megegyezik a befektetési lehetőségek számával (ami most 76). Különböző pontossággal oldottuk meg a feladatokat, a legszorosabb alkalmazott relatív tolerancia $1e-7$ volt. A megoldáshoz szükséges vágások száma még ilyen előírt pontosság mellett sem haladta meg a 120-at, és a megoldáshoz szükséges idő 1 perc alatt maradt.

A kísérletekhez használt számítógép adatai: Intel CPU T2250; 1,7 GHz; 2 GB RAM. A megoldó eljárásokat AMPL modellező rendszerben implementáltunk, a megoldáshoz a Brunel Egyetemen kifejlesztett FortMP szolvert használtuk.

A 4. fejezetben kétlépcsős sztochasztikus programozási feladatok megoldásával foglalkozom.

A Brunel Egyetemi kutatócsoporttal végzett kísérleteinkben a legkisebb feladat az *Airlift*, amely az *SLPtestset* gyűjteményből származik. Itt az első lépcső feladat mérete 2×4, a második lépcső feladaté 6×8. A szcenáriók száma 25, és így az ekvivalens LP feladat mérete 152×204, benne a nem-nulla elemek száma 604. A legnagyobb feladat a paderborni kollégáktól származó *Saphir* volt. Itt az első lépcső feladat mérete 32×53, a második lépcső feladaté 8.678×3.924. Ezer szcenárióval felépítve, az ekvivalens LP feladat mérete $8.678.032 \times 3.924.053$, benne a nem-nulla elemek száma 22.733.103. Ennek megoldása a level módszert alkalmazó szolverrel kb 800 másodpercig tartott. A kísérletekhez használt számítógép paraméterei: Intel CORE i5M520 CPU; 2,4 GHz; 6 GB RAM.

A paderborni kutatócsoporttal végzett kísérletekben a legkisebb feladat a Bíráló által összeállított tesztfeladat-gyűjteményből származik. Itt az első lépcső feladat mérete 10×20, a második lépcső feladaté 20×30. A szcenáriók száma 400, és így az ekvivalens LP feladat mérete 8.010×12.020, benne a nem-nulla elemek száma 72.483. A legnagyobb feladat itt is az ezer szcenárióval felépített *Saphir* volt. Ennek megoldása a level módszer részben inegzakt változatát alkalmazó szolverrel kb 240 másodpercig tartott. A kísérletekhez használt számítógép paraméterei: Intel i7-3770 (4 mag); 3,4 GHz; 16 GB RAM.

Általában a kétlépcsős feladatokkal végzett kísérleteinkben a dekompozíciós eljárások során adódó lineáris és kvadratikus feladatokra a CPLEX duál szimplex megoldót használtuk. A feladatokat általában 5 pontos jegyre oldottuk meg.

Az 5. fejezetben kétlépcsős sztochasztikus programozási feladatok megengedettségi kérdéseivel foglalkozom. A *PLTEXPA2* feladattal kísérleteztünk, amely a *POSTS* tesztfeladat-gyűjteményből származik. Az első lépcső feladat mérete 62×188, a második lépcső feladaté 104×272. A legnagyobb feladatban a szcenáriók száma 600.000 volt. Ennek megoldásához 4 percre volt szükség. A kísérletekhez használt számítógép paraméterei: 1,5 GHz; 1 GB RAM.

A 6. fejezetben kockázatkerülő kétlépcsős sztochasztikus programozási feladatok megoldásával foglalkozom. A kísérleteket a paderborni kutatócsoporttal végeztük. A 4. fejezettel kapcsolatban említett teszt-feladatok közül a 44 legnagyobbal foglalkoztunk, és ezekhez kockázati korlátokat adtunk. Az ezer szcenárióval felépített kockázatkerülő *Saphir* feladat megoldása a korlátos level módszer részben inegzakt változatát alkalmazó szolverrel kb 250 másodpercig tartott. (A használt számítógép azonos volt a 4. fejezettel kapcsolatban említettel.)

A 7. - 10. fejezetekben valószínűséggel megfogalmazott feladatokkal foglalkozom. Egyelőre csak a valószínűség maximalizálási eljárásokat implementáltuk. Ismert tesztfeladatokkal kísérleteztünk. (Ezek eredeti formája költségminimalizálás volt valószínűségi korlát mellett. Költség-korlát hozzáadásával valószínűség maximalizálási feladatokká alakítottuk őket.) Mindegyik tesztfeladatunkban a véletlen paraméterek nem-degenerált normális eloszlást követnek.

A talált optimális megoldásokban a valószínűségi függvények Hesse mátrixai jól kondicionáltak. Kisebb tesztfeladatainkban a véletlen paraméterek száma 3 - 5, a nagyobban pedig 15; ez utóbbi a szakirodalomban 'cash matching' néven ismert feladat.

Nagyobb tesztfeladatunk különlegessége az időt leíró szerkezet. A véletlen paraméterek egymásra következő években felmerülő fizetési kötelezettségeket reprezentálnak. Távolabbi jövőben a bizonytalanság nagyobb. Bármely két véletlen paraméter korrelációja pozitív. A korrelációs mátrixban a főátlótól távolodva a komponensek csökkennek.

Először egy determinisztikus megoldó eljárást fejlesztettünk ki, és mindenütt nagy pontossággal igyekeztünk számolni. Ez időigényes volt, a teszteket kötegelve futtattuk, és általában egy-egy nap múlva volt érdemes megnézni az eredményeket. Javításként az újabb próbapontok meghatározását egyetlen vonalmenti keresésre egyszerűsítettünk, és egy-egy vonalmenti keresés során is csak egy vagy két lépést tettünk. A futási idők lényegesen rövidebbek lettek, de a 'cash matching' feladat megoldása még így is több órát vett igénybe (aminek oka, hogy a valószínűségi függvény gradienseit túlzott pontossággal számoltuk).

Kezdeti tapasztalataink alapján véletlenített eljárást dolgoztam ki, amely becsült gradienseket használ. Ez a szimulációs eljárás percek alatt megfelelő pontossággal megoldja a 'cash matching' feladatot. – Az [54] (Ann. Oper. Res. 2019) cikk esettanulmányához tíz futtatást végeztünk, mindegyik futtatásban 50 iterációt hajtottunk végre. Egyik futtatás sem tartott tovább 2 percnél. A különböző futtatások során talált optimumok eltérése $0,0003$ alatt volt. – Megállási feltételt is kidolgoztam, amely egy adott futtatásban a megoldás pontosságát becsli a szimulált gradiens konfidencia-intervalluma alapján. (Ez a megállási feltétel egyelőre nem adja vissza a valójában elért pontosságot. A fent említett tíz futtatásunkban a becsült hiba-korlátok $0,025$ és $0,03$ között voltak.)

Kísérleteinkben a valószínűségi függvény gráfját közelítő keretet MATLAB-ban implementáltuk. A maszter feladatok megoldására a CPLEX for MATLAB toolbox-ot használtuk. A normális eloszlásfüggvény értékének számítása illetve becslése a QSIMVNV függvénnyel történt. A kísérletekhez használt számítógép paraméterei: Intel Core i7-7500U; 2,7 GHz; 8 GB RAM.

Budapest, 2020. november 5.

Fábián Csaba