

NÉHÁNY IDŐSZERŰ KÉRDÉS SZÁMOLÓGÉPEKKEL KAPCSOLATBAN, I.

Írta: POGÁNY CSABA

Varga Ottó akadémikus emlékére

I. Extremális programok*

1. Bevezetés

Érdekes és paradox az a jelenség, hogy az „operációkutatás” — nyugodtan nevezhetjük *optimalizáláselméletnek* — mind a mai napig nem jutott el addig, hogy vizsgálja az optimális megoldások megkeresésének optimális útjait. Arról nem is beszélve, hogy esetleg az volna az „optimális”, ha az optimalizálandót és az optimalizáló tevékenységet együtt tekintenék optimalizálandó rendszernek.

Nem ismeretes e témakörnek egyetlen akár csak említésre méltó, alkalmazható eredménye sem. Különösen kirívó és érzékeny hiányosság az „operációkutatásnak” a számológéppel kapcsolatos szinte teljes kiépitetlensége — és ez éppen olyan időkből, amikor minden operációkutató szinte naponta „operál” számológéppel.

Az itteni, extremális programokkal kapcsolatos fejtegetéseknek az a fő célja, hogy ráirányítsa a figyelmet erre a fontos problémakörre.

2. Nehézségek

Az optimalitás definiálása rendkívül bonyolult probléma. Valószínű, hogy a valóságos élet kérdéseiben nem is fogalmazható meg pontosan e fogalom. A tudományos kutatás céljaira szimplifikált „valóság modellek” esetében egy-egy konkrét folyamatra azonban sokszor megadhatók optimalitási szempontok és kritériumok.

Komoly nehézség lép fel azonban akkor, ha egy folyamatot többféle szempontból kell optimalizálni. E „dimenziós” probléma egzakt eszközökkel nem is oldható meg (ha megoldható volna, akkor lényegében nem volna dimenziós jellegű). Mégis válaszolni kell, mi legyen az optimalitás-kiválasztási politika: hogyan kell „átszámítani” egyik dimenziót a másikba? Tovább súlyosbodnak a nehézségek, ha figyelembe kell venni még azt is, hogy az előbbi kérdést időben lejátszódó folyamatokra kell (kellene) megoldani.

3. Szuboptimumok

Fontos, de teljesen elhanyagolt terület az alábbi. Eddig az optimalizációs vizsgálatok többsége úgy folyt le, hogy felmerült egy probléma, amelynek megoldására bizonyos erőforrások álltak rendelkezésre. Elkezdtek „optimalizálni” és folytatták ezt az erőforrások kimerüléséig. Nem vizsgálták azt, hogy az optimalizálásra fordítandó erőket hogyan célszerű felosztani az optimalizált terület és az optimalizáló apparátus között; nem vizsgálták, hogy mennyire kell optimalizálni magát az optimalizálási területet és mennyire az optimalizáló apparátust. Mihelyt

* Ez a dolgozat lényegében a szerzőnek „*A számítástechnika alkalmazásai új tudományterületeken*” című kollokviumon 1969. június 4-én tartott előadását tartalmazza.

ennek a vizsgálatába kezd azonban valaki, azonnal tovább bővíti az optimalizálandó rendszer: meg kell mondani, hogy milyen legyen az erőforrások felosztása, most már

1. az (eredetileg egyedül) optimalizálandó terület,
2. az 1-et optimalizáló apparátus,
3. az erőforrások 1 és 2 közötti „optimális” felosztását végző apparátus között.

Ha erre választ adunk, akkor ez is erőforrásokat igényel, amit fedezni kell valamiből, tehát a fenti 3. pont mellé egy negyedik lép, amely az erőforrások 1, 2, 3 közötti felosztását végző apparátusra vonatkozik, és ez így bővílné a végtelenségig. Megállni azonban célszerű valahol. Mi ennek a paradoxon jellegű problémának az egzakt megközelítése? Vagy másképp: mi az „optimális” szuboptimum?

4. Szuboptimumok számológépeknél

Az itt következő minden kérdés valójában több kérdésre bontható aszerint, hogy a szóban forgó számológépnek hány műveletvégző egysége működhet párhuzamosan és hány program futtatása bonyolítható le egyidejűleg. E részproblémák megfogalmazása azonban helykímélés végett az olvasóra marad.

4.1. Hely szuboptimum

Mi azoknak a („külső”, „belső” stb.) tároló rekeszeknek a minimális (maximális, átlagos stb.) száma, amelyeken egy adott feladatot elvégző program elfér?

4.2. Idő szuboptimum

Mi az a minimális (maximális, átlagos stb.) idő, amelynek elteltével adott feladatra készült program eredményesen lefut?

4.3. Megbízhatósági szuboptimum

Ismeretes, hogy a számológépek különböző részei más-más megbízhatósággal dolgoznak. Mi az elérhető maximális (minimális, átlagos stb.) megbízhatóság?

4.4. Pontossági* szuboptimum

Mi egy adott funkciójú programnál a fellépő maximális (minimális, átlagos) pontosság?

* A legtöbb numerikus eljárás konvergenciáját, végességét, hibakorlátait stb. csak valós számokra, a valós számtest műveleti azonosságainak felhasználásával bizonyították. A legtöbb digitális számológép azonban nem valós számtesttel, hanem diadikus racionális számok egy részhalmazával dolgozik, és olyan, hogy a benne előállítható diadikus racionális számok halmazának elemei a valósanalóg műveletekre nem alkotnak testet. (Sok esetben pl. nincs értelme a disztributív azonosságnak, vagy ha van is, akkor sem mindig igaz; vagy például az $(a/b) b$ általában nem lesz mindig a -val egyenlő stb.) A valós számokra és a valós testbeli műveletekre vonatkozó algoritmusokat valós algoritmusoknak nevezve kimondható tehát, hogy valósan konvergens vagy véges algoritmus nem szükségképpen lesz konvergens, illetve véges, ha a benne szereplő számokat és műveleteket konkrét számológépi megfelelőikkel helyettesítik. Ténylegesen sok, valósan konvergens vagy véges algoritmus — a közhittel ellentétben — nem lesz konvergens illetve véges, ha eredeti formájában számológépi megvalósítására kerül sor, a valóban kiszámított hibakorlátok érvényességéről nem is beszélve.

Ebben a dolgozatban konvergencián, végességen, pontosságon stb. a *gyakorlatilag* valóságos (számológépen realizált, illetve realizálható) konvergencia-, végességi-, pontossági- stb. jelenségeket kell érteni.

4. 5. Információs szuboptimumok

Az összes itt felvetett kérdés egy információfeldolgozási folyamatra vonatkozik. A következőkben azonban a folyamatnak időbeli lefolyásán van a hangsúly. Mi az optimális információbemeneti tárolási, kihozatali forma?

Itt egy bizonyos információmennyiség időben lefolyó bevitele, feldolgozása és kiadása a vizsgált folyamat. Az információ kihozatala is időben történik, nem egyforma sebességgel. Mi az optimális időbeli ütemezés?

Minden információfeldolgozási folyamatot ellenőrizni szoktak. Ellenőrzési szempontból mely folyamatok (programok) az optimálisak?

Szintaktikus és szemantikus hibamegkeresési szempontból melyek az optimális információfeldolgozási rendszerek (programozási nyelvek, programszerkesztési elvek stb.)?

5. További problémák

A következőkben felsorolt problémák az információfeldolgozási folyamat egy szakaszának néhány jellemzőjével kapcsolatosak. Ahhoz, hogy e tudományterületet eredményesen lehessen művelni, e jellemzők közötti legfontosabb összefüggéseket kell először felderíteni.

Ha az olvasó figyelmesen végignézi a fenti problémákat, megállapíthatja, hogy szigorúan véve egyiknek sincsen értelme. Mindegyikhez hozzá kell még venni a korlátozó feltételeknek egy halmazát, hogy a feladatok értelmessé váljanak. Könnyen belátható azonban, hogy a megemlített öt szemponton kívül még mások is vannak (pl. egy információfeldolgozási folyamat megszakíthatósága, az általa felhasznált algoritmus, utasításkészlet, a folyamat végrehajtásának ára stb.). Ezenkívül a fenti szempontok szerint sem állandók egy folyamat jellemzői (például egy program helyfoglalása időben változhat).

-Csupán mintának álljon itt egy pontosabban megfogalmazott probléma. (Az olvasó érdekes problémák gazdag sokaságát nyerheti, midőn a jellemzők egy részét korlátozva más jellemzők extrémális értékeit keresi.)

Adott bemenőadat-sorozat mellett, adott helyen, adott megbízhatósággal mely algoritmus adja a legpontosabb eredményt? (Tudott dolog, hogy a számológép műveletei nem tekinthetők pontosaknak. Ennek ellenére pontosaknak feltételezve őket, ugyanarra a célra szolgáló számolási folyamatok végén kapott eredmények valójában különböző pontosságúak lehetnek. A probléma ekkor e „pontossági” hipotézis mellett is megfogalmazható.)

Elméletileg és gyakorlatilag is érdekes probléma egy információfeldolgozási folyamatot egyértelműen meghatározó feltételrendszerek megadása is. (Tágabban: „egyenletek” megoldása információfeldolgozási folyamatokra mint ismeretlenekre.)

Végezetül még néhány gyakorlati szempontból fontos problémára hívjuk fel a figyelmet.

Vannak-e olyan programtranszformációk, amelyekkel pl. egy program lefutási ideje csökkenthető, esetleg a helyigény növelésével vagy fordítva. Általában lehet-e eljárásokat adni, amelyeknek segítségével a számolási folyamatnak valamely jellemzője adott irányban megváltoztatható?

Keresendők abszolút (és relatív) korlátok arra vonatkozóan, hogy egy számolási folyamat adott jellemzője legalább és legfeljebb mennyi? (Pl. adott feladat elvégzése

adott kódrendszer és műveleti idők mellett, adott tárolórekesz-mezőn legalább és legfeljebb mennyi időt igényel?)

Hogyan lehet a számolási folyamatok jellemzése révén a számológépek globális jellemzését megközelíteni, azaz választ adni arra a kérdésre, hogy „melyik gép a jobb?” (Ez a probléma összefügg az optimális kódrendszer problémájával, annak szélső eseteként is felfogható.)

(Beérkezett: 1969. VI. 24.)