Egyetemi doktori (PhD) értekezés tézisei

Optimalizációs feladatok számítógépes feldolgozása

Bekéné Rácz Anett

Témavezető: Dr. Bajalinov Erik

DEBRECENI EGYETEM
Informatikai Tudományok Doktori Iskola
Debrecen, 2012

1. fejezet

Bevezetés

Az optimalizációs technológiák kulcsfontosságúak lettek a modern világ alkalmazott tudományágaiban, az üzleti döntéshozatalban olyan területeken, mint az űrkutatás, biotechnológia, pénzügy, mezőgazdaság, energia ipar, telekommunikáció, szállítmányozás, logisztika. Egy 1993 felmérésből [17] megtudhatjuk hogy az Egyesült Államok ötszáz legnagyobb vállalatának közel 85 százaléka már használt optimalizációs modelleket.

Az operációkutatás (OR) első alkalmazási területe a harcászati operációkutatás volt a II. világháború idején. Majd ezután a kutatók széles körben keresték a lineáris programozás (LP) további alkalmazási lehetőségeit tudományos és ipari területeken is. Ezekben az időkben dolgozta ki G. B. Dantzig a szimplex algoritmust [4], ami azóta a legelterjedtebb módszerré vált. Az OR ezen eredményes időszakához köthető még H. W. Kuhn Magyar módszere [13], amit Egerváry Jenő és Kőnig Dénes munkáin alapulva fejlesztett ki hozzárendelési feladatok megoldására. A kezdeti sikerek után némi visszaesés volt tapasztalható az alkalmazások terén. A következő időszak jellemző célja az újabb megoldó algoritmusok kifejlesztése volt, mint L. G. Khachian ellipszoid algoritmusa [11] (ami az első bizonyítottan polinomiális idejű algoritmus) vagy a belső pontos algoritmusok számos változata, melyek többnyire Karmarkar eljárását [9] veszik alapul. További érdekességeket olvashatunk az OR történetéről a [5], [12] művekben.

A számítógépes hardware-k folyamatos fejlődése és teljesítmény növekedése lehet az oka az optimalizációs algoritmusok (különösen a szimplex algoritmus) sikerének. A legnagyobb feladatot a jól ismert algoritmusok számítógépes környezetben történő hatékony alkalmazása jelenti. Habár a Lineáris Programozási (LP) szoftverek és a számítógépek egyre fejlettebbek, az LP modellek is egyre összetettebbek. Ezért olyan fontos az algoritmusok hatékonysága és fejlett implementációik. Az un. large scale problémák megoldása jelentős számítógépes erőforrást igényel, és a lebegőpontos aritmetikából adódóan numerikus pontatlanságok is megjelenhetnek, amik akár fel is halmozódhatnak a megoldás folyamata alatt. Ezek a hibák nem csak a végeredményre, de a megoldáshoz szükséges iteráció számra is hatással lehetnek, ha egy szimplex iteráció során egy kis értéket, ami valójában csak egy kerekítési hibából adódó pontatlanság, választunk főelemnek. Ez motivált számos kutatót numerikusan stabilabb és ezáltal hatékonyabb algoritmusok fejlesztésére.

A mi munkáink is azon alapulva születtek, hogy a nagyméretű problémák esetén speciális számítógépes feldolgozásra van szükség, ahhoz hogy hatékony és pontos megoldást kapjunk. Ezek a technikák elengedhetetlenek ahhoz, hogy a szoftverek teljesítménye lépést tartson a modellek növekedésével.

A dolgozatomban bemutatom az általam kidolgozott optimalizációs előfeldolgozó eljárásokat (különös tekintettel a hiperbolikus (LFP) és az egészértékű (IP) problémákra), melyek célja, hogy a problémák szoftveres megoldását stabilabbá és gyorsabbá tegyék.

2. fejezet

Optimalizációs technikák gyakorlati alkalmazásai

A feltételes és feltétel nélküli nemlineáris optimalizációs módszerek már régóta széles körben használatosak a tudományos és alkalmazott területeken. Az utóbbi években több ilyen optimalizaciós módszert alkalmazó kutatásban vettem részt a fizika és a hiperbolikus programozás területén.

A Rodolfo Id Betan argentin kutatóval és Vertse Tamással közös vizsgálatokban a komplex energiájú héjmodell (CXSM) komplex potenciálokra való alkalmazhatóságát vizsgáltuk meg. ([8]) Itt közepes méretű komplex mátrixok sajátértékeit hasonlítottuk azokhoz az eredményekhez, amiket feltétel nélküli optimalizációs probléma megoldásaként kaptunk.

Eredmény. A két módszer eredményei elég jól egyeztek, ahhoz képest, hogy mindkét módszer alkalmazása során a lebegőpontos számolások miatt kerekítési hibák halmozódhattak. Bár a hibák terjedését részletesen nem analizáltuk, az egyezés mértéke arra utalt, hogy a CXSM sikerrel alkalmazható komplex potenciálok esetében is.

Egy másik magfizikai problémában, amit Salamon Péterrel és Vertse Tamással közösen végeztünk és publikáltunk ([16]) egy olyan feltétel nélküli optimalizációs problémát oldottam meg, amellyel a szerzőtársaim által javasolt SV potenciál optimális paramétereit határoztam meg úgy, hogy az a lehető legjobban illeszkedjen a Woods-Saxon (WS) alakhoz, azaz a minimalizálandó célfüggvény a két potenciál távolságából adódott. Az SV potenciál véges távolságon simán tart nullához, míg az általánosan használt WS alakot mesterségesen kell csonkolni egy véges távolságnál.

Eredmény. Kimutattuk, hogy rezonanciák esetében a WS potenciálbeli trajektóriák alakja függött a levágási távolságtól. Ilyen függés az SV potenciálnál természetesen nem lépett fel. Ez egy újabb érv volt az SV potenciál használata mellett a magfizikai számolásokban.

Egy most beküldött munkában ([6]) az SV potenciál alakját többcélú optimalizációval határoztuk meg. A két függvény távolságának minimalizálása mellett azok deriváltjaiban vett távolság minimalizálása is cél volt. A deriváltak figyelembe vétele azért történt, hogy az SV potenciál alakja sima maradjon.

Kutatásaim folyamán tapasztaltam, hogy az összetettebb optimalzációs problémákban, amelyek általában nagy mátrixokkal manipulálnak és nagyon sok lebegőpontos művelet elvégzését igénylik, speciális technikák alkalmazása lehet szükséges, ha a probléma rosszul kondicionált. Ez motiválta a további kutatásaimat az előfeldolgozó eljárások terén.

3. fejezet

Preprocessing

Az előfeldolgozó műveletek nagyon fontosak amikor nagyméretű optimalizációs feladatot oldunk meg, különösen ha szimplex vagy belső pontos algoritmust alkalmazunk. A legtöbb professzionális solver (IBM CPLEX, LINGO, GUROBI, stb.) automatikusan használ valamilyen előfeldolgozó technikát annak érdekében, hogy megtartsa a numerikus stabilitást és javítsa a teljesítményt. Bár a számítógépek teljesítménye egyre fejlődik, az alkalmazott modellek mérete egyre nagyobb. Ennek oka lehet a növekvő igény a minél realisztikusabb és összetettebb modellek iránt, vagy a modellgenerátorok használata. A dolgozat 2.2 fejezetében bemutatok néhány előfeldolgozó technikát és azok általam készített hiperbolikus változatát.

3.1. Presolve

A presolve eljárások a modell méretének csökkentéséért felelősek, amit különböző műveletekkel, mint a változó rögzítés, a redundáns feltételek eltávolítás, valamint a nem megoldható problémák előzetes felismerésével érnek el. Ezen egyszerűsítések, módosítások elvégzésekor az optimalizációs modell más adatai is elveszhetnek, mint például a duális információk. Ahhoz, hogy helyreállítsuk ezeket a fontos információkat speciális utómunkálatokra van szükség.

Eredmény. A dolgozat 2.2.1 fejezetében bemutatom a LFP problémák presolve műveletei kapcsán elért eredményeimet. Ezek a kutatások a LP-ban jól ismert presolve eljárásokon alapulnak. Megmutattam, hogy ezek az LP presolve technikák nem alkalmazhatóak az eredeti formában LFP feladatokra, ezért szükséges az adaptációjuk. Kidolgoztam a presolve eljárások LFP problémákra alkalmazható megfelelő változatait. Nem csak a presolve, de a szükséges utófeldolgozó, azaz postsolve műveletek is különböznek LP és LFP esetekben, így ezek megfelelő vizsgálatát is elvégeztem, és megadtam az LFP presolvehoz kapcsolódó megfelelő postsolve műveleteket is. Az LP és LFP esetben történő presolve folyamatok különbségének vizsgálata után megmutattam azt, hogy miért érdemesebb az általam kidolgozott LFP presolve műveleteket alkalmazni egy LFP modellen ahelyett, hogy a modellt Charnes-Cooper transzformációval (CCT) lineáris analóg alakúra hoznánk és ezután alkalmaznánk rá a LP presolve eljárásokat. Bebizonyítottam, hogy a CCT olyan szerkezeti változásokat eredményez a LFP modellben, melyek miatt az LFP presolve során elvégezhető egyszerűsítések többé már nem lesznek elvégezhetőek a lineáris analóg modellen.

3.2. Skálázás

A nagy méretű problémák megoldása lebegőpontos műveletek százezreit igénylik. A számítógépes aritmetika véges pontosságából adódóan kis kerekítési hibák keletkezhetnek a műveletek alatt, mely hibák tipikusan fel is halmozódnak, ami gyakran egy numerikusan instabil problémához és viszonylag nagy pontatlansághoz vezet a megoldást illetően. Annak érdekében, hogy elkerüljük ezeket a hibákat, a legtöbb professzionális optimalizáló szoftver (solver) speciális kifinomult technikákat alkalmaz, amelyek nagymértékben csökkentik a kerekítési hibák felhalmozódását és ezáltal növelik a solverek teljesítményét. Az egyik legegyszerűbb és legelterjedtebb módszer erre a skálázás. A célja, hogy az input modellt egy numerikusan stabilabb formára hozza. A skálázás, hasonlóan a presolve műveletekhez, igényel bizonyos utómunkálatokat (re-skálázást), ahhoz hogy a skálázott feladat optimális megoldásából visszanyerjük az eredeti feladat optimális megoldását. Legtöbb esetben a skálázás javítja az optimalizációs probléma numerikus tulajdonságait, azaz csökkenti a $\sigma(A)$ értékét (lásd a dolgozat 2.1 fejezete) valamint a szimplex algoritmus iterációszámát is képes csökkenteni.

Eredmény. A dolgozat 2.2 fejezetének második részében bemutatom a skálázási eljárások LFP problémákra adaptált változatát. A LFP esetén szükséges szabályrendszer kidolgozása során különböző skálázó eljárásokkal ismerkedtem meg ([7], [14], [2]), ami arra ösztönzött, hogy kidolgozzak egy új hatékonyabb változatot ([3]). A saját Min-Max nevű eljásomat összehasonlítottam az Edinburghi Egyetem, Matematika és Statisztika tanszékén fejlesztett két másik eljárással, a Középérték és a Geometriai szabállyal ([2], [14]).

A számítógépes teszteket a NETLIB-ről származó hivatalos tesztfájlokon végeztem. Ezek a tesztek azt mutatják, hogy a Min-Max módszer sokkal hatékonyabb, mint a Geometriai szabály, mert ellentétben azzal minden teszt feladat esetén el tudta érni az előre definiált $\sigma(A)$ értéket. A szükséges futási idő a Min-Max módszer és a Középérték szabályok esetén közel azonos, azonban találunk olyan problémákat (például STOCFOR2, SHIP12L), ahol kicsit nagyobb eltérés figyelhető meg hol az egyik hol a másik módszer javára.

Egy könnyen implementálható eljárást dolgoztam ki, ami a Geometriai szabálynál hatékonyabbnak bizonyult, a Középérték szabályhoz hasonlítva azt tapasztaltam, hogy a $\sigma(A)$ értékében vett javulás tekintetében közel egyformák néhány speciális esetet kivéve, ahol kisebb eltérések jelennek meg.

4. fejezet

Sugár-módszer

A dolgozatom harmadik részében (Chapter 3) az optimalázációs problémák egy másik csoportjával az egészértékű feladatokkal (IP) foglalkoztam. A célom itt is a számítógépes megoldás hatékonyságának javítása volt, a branch and bound (B&B), azaz a korlátozás és szétválasztás algoritmusra koncentrálva, ami a professzionális solverek között a legszélesebb körben implementált algoritmus a vegyes (MIP) és tiszta egészértékű (IP) feladatok megoldására. A dolgozat 3.2 fejezetében részletesen bemutatom az általam kidolgozott eljárást, mely egy induló korlát meghatározással javítja a B&B algoritmus hatékonyságát. Az ötlet V. R. Khachaturov és N. A. Mirzoyan ([10]) munkáján alapul, ami LP feladatok megoldására irányult. Az itt leírt elméleti eljárás számos implementációs problémával rendelkezik. Khachaturov sugár-módszer ötletét használva dolgoztam ki a saját eljárásomat, amely egészérétkű feladatok egy lehetséges egész megoldását keresi, melynek célfüggvény értéke hatékonyan alkalmazható, mint induló korlát a B&B algoritmus esetén. Egy ilyen eljárástól észszerű elvárás, hogy gyors legyen és törekedjen arra, hogy az optimális megoldáshoz minél közelebb eső lehetséges egész megoldást szolgáltassa.

Eredmény. A kidolgozott eljárást először a WINGULF[1] programba beépítve véletlenszerűen generált feladatokon teszteltem. A pozitív eredmények hatására további tesztek futtatását tettem lehetővé, azzal hogy a sugár módszeremet C++ programnyelvi környezetbe implementáltam és vegyes egészértékű feladatokra általánosítottam. Így az IBM CPLEX professzionális solvert használva már hivatalos MIPLIB tesztfájlokon tudtam tesztelni a módszer hatékonyságát. Összehasonlítottam a CPLEX (B&B algoritmussal) futási idejét abban az esetben amikor használta a módszerem által generált induló korlátot és alap beállításon amikor nem kap semmilyen plusz külső információt. A tesztek az mutatják, hogy az eljárásom a CPLEX futási idejét átlagosan 21%-kal (a sugár módszerem futási idejét is beleértve) tudta gyorsítani nagy méretű IP és MIP problémákon. Azt mondhatjuk, hogy a sugár módszerre fordított elhanyagol-

[1]General User-friendly Linear and linear-Fractional programming package for educational purposes. Developed at the University of Debrecen. See: http://zeus.nyf.hu/ bajalinov/WinGulf/wingulf.html

ható futási idővel szignifikáns gyorsítást tudunk elérni még az egyik legelterjedtebb professzionális solver teljesítményében is.

PhD Thesis

Computational methods for optimization problems

Anett Bekéné Rácz

Supervisor: Dr. Erik Bajalinov

UNIVERSITY OF DEBRECEN

Doctoral School of Informatics

Debrecen, 2012

Chapter 1

Introduction

Optimization is a basic tool in applied research, engineering, business, and sciences. Operations research (OR) has become a key component of modern-day decision-making in such fields as aerospace, biotechnology, finance, forestry and agriculture, energy, telecommunication, transportation and logistics, the Internet, etc. A survey from 1993 showed that nearly 85 percent of the five hundred largest companies in the United States used optimization models [17].

First application area was the military operations during the II. World War. After the "heroic time"[1] came the "golden age"[1], when many researchers attempt to made linear programming the most frequently used optimization technique not only in military applications but also in science and industry. In those days developed G. B. Dantzig his simplex algorithm [4] which became the most widespread algorithm and also H. W. Kuhn published his Hungarian method [13] based on a research of Jenő Egerváry and Dénes Kőnig for solving assignment problems. Then came the "crisis"[1] period when researchers strived to develop different algorithms such like ellipsoid algorithm by L. G. Khachian [11] (the first polynomial time algorithm) and several variations of interior-point method (most of them based on the Karmarkar's algorithm [9]). More information about the history of linear programming can be found in [5], [12].

The next period in the history of OR is connected with the spread of computers. Performance leaps and continuous improvement of computer hardware might be a reason for the success of optimization algorithms (especially the simplex algorithm). The most important challenge is to use the well-known algorithms efficiently in computational environment by developing powerful codes. Although LP softwares and computers have became much faster, LP models have increased in size. More efficient algorithms and improved implementation techniques are therefore still very important. Large scale problems require enormous computational effort, and because of floating point arithmetic numerical inaccuracies can occur and also accumulate. These numerical errors can affect

[1] Historical classification from [12].

not only the final solution but also the number of iterations which can dramatically rise at any time if a small value, which is actually a numerical garbage, was wrongly chosen as a pivot element. These phenomenon has motivated several research on algorithmic techniques that are numerically more stable and also efficient.

Also, we were motivated by the fact that large scale problems to be solved correctly and efficiently usually require such special additional computational techniques as preprocessing and scaling. These techniques very often lead to considerable performance improvement of used solvers.

The main goal of my dissertation is to present the methods developed by myself during my PhD studies for preprocessing optimization problems (especially for Linear-Fractional (LFP) and Integer Programming (IP) models) in order to obtain a more stable and faster solution process.

Chapter 2

Experiences on applications of optimization techniques

Nonlinear optimization methods with and without constraints have been frequently used in sciences and engineering for a long time. I took part in several research in field of Physics and Linear-Fractional programming during the past few years.

Cooperating with T. Vertse and R. Id Betan we used unconstrained minimization problem when expanding the complex energy shell model (CXSM) calculation for complex potentials. ([8])

Result The application developed and performed for several IARs confirmed that the CXSM method is able to reproduce the position and the full width of the IAR even in the complex potential case.

In another work of us with P. Salamon and T. Vertse I used unconstrained optimization with a simple non-linear real function of three parameters to make SV potential, which approaches zero smoothly without artificial cut-off, resemble most to the Woods-Saxon form, which had to be cut at or before a particular point. ([16])

Result The shapes of the pole trajectories were quite different for the two potential forms, which has an important consequences in nuclear structure calculations. As a modification of the previous optimization I introduced a multi-objective minimization, which forced also the derivatives of the shapes of the two potential forms to be similar as much as possible. The results of this optimization will be used in a coming publication of us [6].

During these researches I experienced that the development of computational power made it possible to handle complicated problems including large matrices, where the number of floating point operations increased dramatically and also that large or badly scaled problems require special computational techniques.

Experiences on these topics motivated my investigations on preprocessing techniques.

Chapter 3

Preprocessing

Preprocessing is very important for solving large optimization problems irrespectively of using interior point or simplex algorithm. Most of the professionally developed solvers (like IBM CPLEX, LINGO, GUROBI, etc.) automatically use preprocessing techniques to maintain numerical stability and improve performance. Even though computers have become even faster, the real-world models have increased in size. The reason can be the growing complexity requirements and the new sophisticated model generators too. In section 2.2 of the dissertation I consider some preprocessing techniques (presolve and scaling), and adapt them to Linear-Fractional problems.

3.1 Presolve in LFP

Presolve is a special technique developed especially for model reduction and is based on the use of such different operations like fixing variables, finding and removing redundant constraints, and detecting unbounded or infeasible problems. When performing these manipulations some data from the dual problem may be lost or distorted. To restore such data some special postsolve operations are required.

Result In section 2.2.1 I describe the main results of my investigations connected with preprocessing techniques in LFP. My investigations are based on well-known preprocessing techniques used in LP. I show that well known LP techniques in the case of LFP can not be applied in original form and must be adapted in the corresponding way. I developed the appropriate adaptation of these techniques to LFP problems. In certain cases this adaptation is not trivial. Not only preprocessing but also postsolve techniques are different in the case of linear fractional programming problems. I introduce some preprocessing techniques with the corresponding postsolve operations based on [1], [15] and present how I adapted them to LFP.

I discuss the main differences between the presolve and postsolve operations in LP and LFP and I also show the advantages of using my LFP presolve adaptation instead of transforming the LFP model into LA form by Charnes-Cooper transformation (CCT) and using LP presolve on LA model. I proved that if we do not use my LFP presolve on the original model before CCT, in LA model will not be any chance for it. The reason is that the CCT modifies the structure of the model so that the simplification possibilities are damaged.

3.2 Scaling in LFP

Solving a large-scale problem requires hundreds of thousands to millions of floating-point arithmetic operations. Because of the finite precision inherent in computer arithmetic small numerical errors occur during these calculations. These errors typically have a cumulative effect, which often leads to a numerically unstable problem and possibly large errors in the "solution" obtained. To avoid such problems all well-made industrial solvers include special sophisticated techniques that dramatically reduce the cumulative effect of rounding and often lead to considerable improvement in the solvers' performance. One of the most simple, relatively effective and widespread techniques of this type is scaling. It's main goal is to transform the model into a numerically more stable form. Scaling also needs some postsolve ("re-scale") operation, like presolve, to obtain the optimal solution of the original problem from the optimal solution of the scaled, and therefore numerically more stable problem. In most cases scaling improves the numerical characteristics of a problem, i.e. reduces the $\sigma(A)$ value of the model-matrix (see: section 2.1 in the dissertation) and sometimes it can also reduces the number of iterations needed during the simplex method.
Result In the second part of chapter 2.2 of the dissertation I describe the LFP adaptation of general scaling rules. During my investigations I got acquainted with different variations of scaling methods ([7], [14], [2]), which motivated me to develop more effective scaling algorithms. I developed my new scaling rule ([3]) and compared with two other scaling method, Mean rule and Geometric rule ([2], [14]) which have been implemented in the linear programming codes developed at Edinburgh University, Department of Mathematics and Statistics, Scotland.

Computational test on official test files from NETLIB library were conducted. These tests show that my Min-Max method is more efficient than the Geometric rule, because it can reach the predefined $\sigma(A)$ for every test problem, while geometric rule does not. The required time was almost the same in case of our Min-Max method and Mean rule, but there are several test problem (for instance STOCFOR2, SHIP12L) on which the one or the other method was a bit faster.

I proposed an easy to implement scaling method which is more efficient than Geometric rule. Compering it with the Mean rule we can see that both rules are quite similar from the point of view of the run time and improvement achieved in $\sigma(A)$, but there are some special cases when a little difference appears.

Chapter 4

Ray-method

The third part (Chapter 3) of my dissertation concerns with other type of optimization problems, namely the Integer Programming (IP) problems. The goal was the same: To improve computational efficiency of a well-known optimization algorithm, the branch and bound (B&B) method which is the most widely implemented algorithm by professional solvers for solving integer programming (IP) or mixed integer programming (MIP) problems. In section 3.2 I give a detailed description of my new method for calculating initial bound which can improve the efficiency of B&B algorithm. The idea is based on the "ray-method" by V. R. Khachaturov and N. A. Mirzoyan ([10]), which was developed for solving IP problems. This theoretical method in the original form has difficulties concerning computational implementation. So I used the theoretical background and the main idea of Khachaturov's ray-method to get an integer feasible solution for B&B method. Using the objective value of this feasible solution I provide a bound value for B&B method when starting. The requirement for this kind of procedure is to be fast and provide a feasible solution as close to the optimal point as possible.

Result I tested my ray-method on random generated IP problems using WINGULF[1] at first. Encouraged by the results on these test files I conducted further tests now using professional solver CPLEX and official test files form MIPLIB. So I adapted the method for MIP problems and developed a test environment using IBM CPLEX Callable Library for comparing the run-time of B&B algorithm in CPLEX with the initial bound found by ray-method or without any initial bound. So I had the opportunity to test my method on large problems from MIPLIB official test collection library. According to these test results my new method can improve the efficiency of the professional CPLEX solver by about 21% on large IP and MIP problems. We can say that spending a very little time to calculating an initial bound using my method a significant

[1]General User-friendly Linear and linear-Fractional programming package for educational purposes. Developed at the University of Debrecen. See: http://zeus.nyf.hu/ bajalinov/WinGulf/wingulf.html

improvement can be reached on solving IP problems even in case of the most advanced solver IBM CPLEX.

Irodalomjegyzék / Bibliography

[1] Andersen, E. D., Andersen K. D., *"Presolving in linear programming"*, Mathematical Programming, Vol. 71, pp. 221-245, 1995.

[2] Bajalinov, E. B., *"Linear-Fractional Programming: Theory, Methods, Applications and Software"*, Kluwer Academic Publishers, 2003.

[3] Bajalinov, E., Rácz, A., *"Scaling problems in linear-fractional programming"*, Acta Mathematica Academiae Paedagogicae Nyiregyhaziensis, Vol. 25, no. 2, pp. 283-301, 2009.

[4] Dantzig, G. B., *"Linear Programming and Extensions"*, Princeton University Press, 1963.

[5] Dantzig, G. B., *"Linear programming"*, Operations Research 50: 42-47, INFORMS, 2002.

[6] Darai, J., Rácz, A., Salamon, P., Lovas, R. G., *Antibound poles in cutoff Woods-Saxon and in Salamon-Vertse potentials*, *submitted*

[7] Fulkerson, D., Wolfe, P., *"An algorithm for scaling matrices"*, SIAM Review, 4, pp. 142–146, 1962.

[8] Id Betan, R., Rácz, A., Vertse, T., *"Calculation of the Isobaric Analogue Resonance Using Shell Model in the Complex Energy Plane"*, International Journal of Theoretical Physics, Vol. 50, no. 7, pp. 2222-2226, 2011.

[9] Karmarkar, N., *"A New Polynomial Time Algorithm for Linear Programming"*, Combinatorica, vol. 4, no. 4, pp. 373-395, 1984.

[10] Khachaturov, V.R., Mirzoyan, N.A., *"Solving problems of integer programming with ray-method."*, Notes on applied mathematics, Computer Center of Soviet Academy of Science, 1987.

[11] Khachian, L. G., *"A polynomial algorithm in linear programming"*, Soviet Mathematics Doklady, vol. 20, pp. 1093-1096, 1979.

[12] Kirby, M. W., *"Operational research in war and peace - The british Experience from the 1930s to 1970"*, Imperial College Press and the Operation Research Society, 2003.

[13] Kuhn, H. W., *"The Hungarian method for the assignment problem"*, Naval Research Logistics Quarterly, vol. 2, pp. 83-87, 1955.

[14] Maros, I., *"Computational Techniques of the Simplex Method"*, Kluwer Academic Publishers, 2003.

[15] Mészáros, Cs., Suhl, U. H., *"Advanced preprocessing techniques for linear and quadratic programming"*, OR Spectrum 25: 575-595, Springer Berlin, 2003.

[16] Rácz, A., Salamon, P., Vertse, T., *"Trajectories of S-matrix poles in a new finite-range potential"*, Physical Review C 84 (3), art. no. 037602, 2011.

[17] Shu-Cherng Fang, Sarat Puthenpura *"Linear optimization and extensions: theory and algorithms"*, Prentice Hall, 1993.

A szerző publikációi / List of publications

[1] Rácz, A., Salamon, P., Vertse, T., *"Trajectories of S-matrix poles in a new finite-range potential"*, Physical Review C 84 (3), art. no. 037602, 2011 (IF: 3.416) *(Scopus, Web of knowledge)*

[2] Id Betan, R., Rácz, A., Vertse, T., *"Calculation of the Isobaric Analogue Resonance Using Shell Model in the Complex Energy Plane"*, International Journal of Theoretical Physics, Vol. 50, no. 7, pp. 2222-2226, 2011 (IF: 0.670) *(Scopus, Web of knowledge)*

[3] Bajalinov, E., Rácz, A., *"Scaling problems in linear-fractional programming"*, Acta Mathematica Academiae Paedagogicae Nyiregyhaziensis, Vol. 25, no. 2, pp. 283-301, 2009 *(Scopus, MathSciNet)*

[4] Rácz, A., *"Determining initial bound by "ray-method" in branch and bound procedure"*, Acta Cybernetica, Vol. 19, no. 1, pp. 135-146, 2009 *(Scopus, MathSciNet)*

[5] Bajalinov, E., Rácz, A., *"Scaling problems in linear-fractional programming"*, Proceedings of the International Conference on Information Technology Interfaces, ITI , art. no. 1708531, pp. 495-499, 2006 *(Scopus, Web of knowledge, IEEE Xplore)*

Közlésre benyújtott cikkek:

[1] J. Darai, A. Rácz, P. Salamon, and R. G. Lovas, *Antibound poles in cutoff Woods-Saxon and in Salamon-Vertse potentials*, Physical Review C

Oktatási segédanyag:

[1] Bajalinov Erik, Rácz Anett, *Operációkutatás I.*, TÁMOP-4.1.2-08/1/A-2009-0046

[2] Bajalinov Erik, Rácz Anett, *Operációkutatás II.*, TÁMOP-4.1.2-08/1/A-2009-0046

A szerző konferencia előadásai / Conference Talks

[1] Anett Bekéné Rácz *"Stabilitás- és teljesítményjavító eljárások optimalizációs feladatokban"*, XXIX. Magyar Operációkutatási Konferencia, Balatonőszöd, 2011. 09. 28.

[2] Zoltán Megyesi, Anett Bekéné Rácz *"Optimalizációs feladatok megoldása GPU segítségével"*, XXIX. Magyar Operációkutatási Konferencia, Balatonőszöd, 2011. 09. 30.

[3] Veronika Mátyus, Zoltán Megyesi, Anett Bekéné Rácz *"MPS processing for optimization problems"*, (conference poster) CSMA 2011, Debrecen

[4] Anett Rácz *"Methods for increasing computational efficiency and stability in solving optimization problems"*, 7th Conference on Applied Mathematics and Scientific Computing - ApplMath11, Trogir, 2011.06.17.

[5] Anett Rácz *"Preprocessing in Linear-Fractional Programming"*, 24th European Conference on Operational Research - EURO XXIV 2010, Lisbon, 2010.07.14.

[6] Anett Rácz *"Preprocessing in Linear-Fractional Programming"*, Conference of PhD Students in Computer Science 2010 - CSCS, Szeged, 2010.07.01.

[7] Anett Rácz *"Adapting LP preprocessing techniques to LFP problems"*, 8th International Conference on Applied Informatics - ICAI 2010, Eger, 2010.01.29.

[8] Erik Bajalinov, Anett Rácz *"On the Ray-based procedure for determining initial bound in branch and bound method"*, Veszprém Optimization Conference: Advanced Algorithms - VOCAL, 2008.12.16.

[9] Anett Rácz *"Determining Initial Bound by "Ray-method" in Branch and Bound Procedure"*, The Sixth Conference of PhD Students in Computer Science - CSCS, Szeged, 2008.07.03.

[10] Erik Bajalinov, Anett Rácz *"A sugár módszer lineáris és hiperbolikus programozásban"*, 9. Gyires Béla Informatikai Nap, Debreceni Egyetem, 2007.11.23.

[11] Anett Rácz *"Skálázási eljárások értékelése"*, XXVIII. Országos Tudományos Diákköri Konferencia - OTDK, Miskolc, 2007. 04. 26.

[12] Anett Rácz *"Scaling Problems in Linear-Fractional Programing"*, 28th International Conference on Information Technology Interfaces - ITI, Cavtat/Dubrovnik, 2006.06.20.