Raport asupra proiectului ATCO, "Technici Avansate in Optimizarea Combinatorie şi Complexitatea Computaţionalã"

Perioada de raportare: **Iulie- Noiembrie 2017**

Gabriel Istrate
6 decembrie 2017

1 INTRODUCERE. DEMARAREA PROIECTULUI

Am fost informaţi despre demararea proiectului pe 13 iulie. Am realizat contractarea, precum şi redimensionarea bugetului/replanificarea activitãţilor pentru a rãspunde procesului de contractare credite de angajament.

Am realizat pagina web a proiectului, disponibilã la adresa `http://tcs.ieat.ro/grants/idei-2016`.

Am procedat la completarea echipei cu poziţiile (vacante) de masterand/doctorand. Echipa a fost completatã cu

- Teodora Selea (doctorand)
- Diana Diniş (masterand)

2 SUMARUL REZULTATELOR

Fiind vorba de perioada de debut a proiectului, cuprinzând doar circa patru luni de lucru efectiv, nu putem raporta deocamdatã rezultate publicate. Pe de altã parte, cercetãrile noastre au condus pânã acum la patru lucrãri, aflate in stadii diferite de lucru:

- Una deja trimisă spre publicare la *Discrete Applied Mathematics.*
- Alte două intr-un stadiu avansat, posibil definitivate până la 31 Decembrie 2017.
- O a patra in curs, cu rezultate (experimentale) solide, dar care necesită definitivarea lor + redactarea lucrării.

Avem, de asemenea, un număr de rezultate preliminare, neacoperite de lucrările raportate mai sus, dar care pot constitui baza unor lucrări in perioada ulterioară de raportare.

3 CERCETĂRI ASUPRA HEAPABILITĂŢII

Cele mai multe rezultate pe care le-am obţinut s-au concentrat asupra problematicii heapabilităţii. Astfel, am obţinut două lucrări aflate intr-un stadiu avansat, pe care preconizăm să le definitivăm/trimitem până la 31 Decembrie 2017.

1. János Balogh, Cosmin Bonchiş, Diana Diniş, Gabriel Istrate, Ioan Todincă *"On the heapability of partial orders ".* Preconizăm să o trimitem spre publicare la *SIAM Journal on Discrete Mathematics.*
2. Cosmin Bonchiş, Gabriel Istrate *"The langage and series of Hammersley?s process".* O vom trimite spre publicare la *Discrete Mathematics and Theoretical Computer Science*, sau la o conferinţă.

O a treia lucrare este in lucru (dar nu este momentan in stadiul avansat al primelor două)

3. Cosmin Bonchiş, Diana Diniş, Gabriel Istrate (titlu provizoriu) *"The Ulam-Hammersley Problem for Heapable Sequences".* O vom trimite spre publicare (când o vom definitiva) la *Experimental Mathematics.*

Rezultatele din aceste lucrări sunt raportate mai jos.

3.1 HEAPABILITATEA ORDINILOR PARŢIALE

Intr-o primă lucrare aflată in curs de definitivare/trimitere, am investigat impreună cu membrii echipei Diana Diniş şi Cosmin Bonchiş, cu János Balogh (Szeged) şi Ioan Todincă (Orleans) câteva rezultate legate de heapabilitatea ordinilor parţiale. Motivaţia acestui studiu este faptul că heapabilitatea "obişnuită" se obţine considerând aşa-numitele *permutation orders*, ordini parţiale induse de non-inversiunile unei permutări.

Am introdus un parametru pentru o ordine parţială numit k-*width*, care generalizează noţiunea clasică de "width" al unei ordini parţiale, definit drept numărul minim de clase intr-o partiţie a mulţimii elementelor ordinii parţiale in subşiruri "k-heapable" (adică arbori k-ari care respectă invariantul unui min-heap)

Am obţinut următoarele rezultate:

1. Pentru ordini de tip interval un algoritm (Figura 8.1) pentru calculul celei mai lungi subsecvenţe heapable. Pentru permutări (echivalent ordinile permutare) complexitatea acestui parametru este o problemă deschisă din lucrarea iniţială a lui Byers et al. asupra heapabilităţii.

2. O teoremă generală care oferă o metodă de calcul pentru parametru k-width bazată pe fluxuri in reţele de transport. Interesul acestui rezultat este faptul că metoda de calcul este o generalizare a teoremei lui Dilworth, arătând legătura acestui rezultat cu conceptul de heapabilitate.

3. Dacă pentru ordini parţiale (in general) algoritmul de calculare a parametrului k-width nu este de tip GREEDY, el fiind bazat pe alte metode (fluxuri in reţele), arătăm faptul că pentru ordini trapezoidale (o clasă de ordini care sunt o generalizare "multidimensională" a ordinilor interval) un algoritm simplu de tip greedy calculează parametrul k-width.

4. Am arătat că pentru **mulţimi de intervale** putem reduce problema la cea pentru şiruri considerând ordonarea intervalelor in ordinea crescătoare a capătului lor din dreapta.

5. Am obţinut o generalizare a procesului Hammersley pentru şiruri de intervale. Acest lucru ne-a favorizat realizarea simulărilor menţionate la punctul următor. Am implementat o simulare a procesului Hammersley pentru intervale. **Programul realizat este disponibil pe (pagina proiectului şi pe) bitbucket.org.**

6. In sfârşit, am obţinut o conjectură (bazată pe date experimentale) privind comportamentul asimptotic al parametrului k-width pentru mulţimi/şiruri de intervale aleatoare. Cumva surprinzător **in ambele cazuri**, comportamentul asimptotic conjecturat este

$$\lim_{n\to\infty} \frac{E[k-\text{width}(R_n)]}{n} = \frac{1}{k+1}. \tag{3.1}$$

Dorim să scriem ulterior o lucrare experimentală extinsă dedicată acestei conjecturi.

3.2 Limbaje şi seriile formale asociate variantelor procesului Hammersley

O problemă legată de heapabilitate este analiza variantelor aşa-numitului proces Hammersley. Acest proces a fost definit in literatură in legătură cu problema Ulam-Hammersley şi extins de noi pentru heapabilitatea permutărilor şi a mulţimilor de intervale. Comportamentul asimptotic al procesului Hammersley determină natura scaling-ului parametrului k-width.

O metodă pentru determinarea matematică a acestui comportament asimptotic este bazat pe folosirea unor metode de combinatorică analitică. Speficic, asociem o serie de puteri (formală) unei stări a procesului Hammersley. Codificăm astfel, cu ajutorul unei serii de puteri, familia probabilităţilor de apariţie a fiecărei stări. Speranţa este ca printr-o analiză a acestei serii de puteri să determinăm "cele mai probabile" stări ale procesului.

In lucrarea menţionatã am demarat acest proces, atât pentru procesului Hammersley arbori (corespunzând heapabilităţii permutãrilor), cât şi pentru cazul intervalelor.

In primul caz starea sistemului la momentul n este codificatã printr-un cuvânt de lungime n peste alfabetul $\Sigma_k = \{0, 1, \ldots, k\}$. **Nu toate aceste cuvinte sunt posibile.**

O primã contribuţie este legatã de caracterizarea tuturor cuvintelor posibile, cu alte cuvinte a *limbajului procesului Hammersley*, suportul seriei formale considerate. **Am identificat complet acest limbaj.** Astfel:

- Pentru cazul $k = 1$ (cazul clasic) limbajul procesului Hammersley este limbajul (regulat) $L_H^1 = 1\Sigma_1^*$.
- Pentru cazul heapabilităţii ($k \geq 2$) limbajul procesului Hammersley este limbajul (context-free dar neregulat)

$$L_H^k = \{w \in \Sigma_k^* | \text{ pt. orice prefix } y \text{ al lui } w, |y|_k - \sum_{i=0}^{k-2} (k-i-1)|w|_i > 0\} \tag{3.2}$$

Pentru cazul procesului Hammersley intervale problema este mai subtilã/complicatã. Astfel, existã nu unul ci douã maniere de a defini limbajul asociat:

- O manierã adecvatã scopului final urmãrit, acela de analizã a comportamentului asimptotic al procesului Hammersley. Ca şi in cazul permutãrilor, starea sistemului este codificatã peste alfabetul $\{0, 1, \ldots, k\}$. Am numit acest limbaj **limbajul Hammersley efectiv** asociat heapabilităţii intervalelor.
- Pe de altã parte, numele de limbaj Hammersley pentru intervale a fost rezervat unei variate care permite obţinerea unei relaţii de recurenţã pentru seria de puteri.

Am demonstrat urmãtoarele rezultate:

- Pentru fiecare $k \geq 1$ limbajul Hammersley efectiv asociat heapabilităţii intervalelor **coincide cu limbajul analog pentru permutãri** L_H^k (determinat anterior).
- Pe de altã parte, $\forall k \geq 1$ **limbajul Hammersley pentru intervale <u>nu este context-free</u>.**

Am obţinut, de asemenea, **algoritmi pentru calculul coeficienţilor seriilor de puteri** asociate diverselor variante ale sistemelor Hammersley. Unul din aceşti algoritmi, ComputeMultiplicity, destinat cazului permutãrilor, este listat in Apendix.

Pe baza acestor algoritmi am tabulat coeficienţii seriilor de puteri pentru un segment iniţial al lui Σ_2. Din pãcate, din inspecţia vizualã a acestor coeficienţi nu se intrevede vreo metodã de a identifica/analiza natura acestor serii de puteri.

Pe de altã parte in cazul $k = 1$ algoritmul ComputeMultiplicity se reduce la o recurenţã. Nu am putut rezolva aceastã recurenţã. Am arãtat insã cã o variantã a acestei recurenţe are drept soluţie coeficienţii multinomiali - credem cã soluţia recurenţei are proprietăţi similare.

Tabularea acestor coeficienţi ne-a permis explorarea unei probleme legatã de natura (conjecturatã) a constantei de scaling in cazul heapabilităţii permutãrilor cu $k = 2$. Conform conjecturilor noastre anterioare, aceastã constantã ar fi egalã cu numãrul de aur.

$$\lambda_2 = \frac{1 + \sqrt{5}}{2}$$

Parametrul introdus in lucrare poartă numele de *increment*. Identificarea distribuţiei lui (când $n \to \infty$) poate valida formula de mai sus.

Folosind coeficienţii seriilor de puteri calculate cu algoritmul ComputeMultiplicity am calculat **distribuţia exactă a parametrului increment** pentru valori mici $n \leq 15$ ale parametrului n. Concluziile sunt următoarele

- Distribuţiile exacte calculate conduc la valori pentru constanta λ_2 destul de departe de numărul da aur.
- Pe de altă parte, acest lucru se datorează faptului că **distribuţia incrementelor converge foarte incet către limita sa**: simulări cu $n = 10000000$ au arătat că la această valoare distribuţia incrementelor este aproximativ o distribuţie geometrică cu media $\frac{\sqrt{5}-1}{2}$.

Cu această prognoză, **valoarea constantei λ_2 este intr-adevăr numărul de aur.**

3.3 Identificarea experimentală a constantei de scaling pentru heapabilitatea permutărilor

Am demarat cercetări experimentale legate de identificarea precisă a constantei de scaling λ_k care caracterizează heapabilitatea permutărilor, pentru $k \geq 2$. Rezultatele obţinute vor fi completate, rezultând, intr-o lucrare trimisă spre publicare in jurnalul *Experimental Mathematics*.

Baza rezultarelor experimentale o constituie identificarea limitei procesului Hammersley cu k vieţi pentru $k \geq 2$. Starea acestui proces la momentul n poate fi descrisă drept un cuvânt de lungime n peste alfabetul $\Sigma_k = \{0, 1, \ldots, k\}$. Când $n \to \infty$, un cuvânt tipic de lungime n va avea aproximativ $c_0 n$ cifre de 0, aproximativ $c_1 n$ cifre de 1,..., aproximativ $c_k n$ cifre de k, pentru nişte constante $c_0, \ldots, c_k > 0$.

Valorile constantelor $c_0, \ldots, c_k$ pot fi prezise folosind argumente matematice neriguroase, de tipul argumentelor "mean-field" folosite de fizicieni. **Experimentele făcute confirmă remarcabil de bine aceste predicţii teoretice.** Spre exemplu, in cazul $k = 2$ predicţiile indică

$$c_0 = c_2 = \frac{3-\sqrt{5}}{2} \sim 0.381, c_1 = \sqrt{5} - 2 \sim 0.236.$$

Figura 8.2 (care conţine estimări preliminare ale constantelor pentru $k = 2, 3, \ldots$) confirmă aceste calcule.

Pe baza acestor calcule/experimente **conjecturăm faptul că λ_k este cea mai mică rădăcină in intervalul** $(1, 2)$ **a ecuaţiei**

$$x^k \cdot (2 - x) = 1.$$

4 Probleme exploratorii in Optimizarea Combinatorie

Rezultatele obţinute in cadrul acestei activităţi au condus la o lucrare trimisă spre publicare

1. Flavius Turcu, Cosmin Bonchiş,Mohamed Najim. *"Vector partitions, multi-dimensional Faá di Bruno formulae and generating algorithms"*. Manuscris, trimis spre publicare in *Discrete Applied Mathematics.*

precum şi la mai multe rezultate necuprinse incã in vreun manuscris.

In lucrarea amintitã se studiazã o problemã combinatorialã motivatã de aplicaţii practice, cea a calculului in serie pentru compunerea a douã funcţii multivariate. Se obţine o formulã, generalizare a formulei Faá di Bruno utilã in aceastã problemã. Se mai aratã utilitatea conceptului de *partiţie a unor vectori* in problema de mai sus. Oferim şi un algoritm pentru calculul acestui tip de partiţii.

Pe lângã rezultatele deja trimise spre publicare am mai obţinut şi alte rezultate preliminare cu potenţial de a constitui lucrãri publicabile, de exemplu in domeniul algoritmilor pentru tabele Young "cu multiplicitãţi", tabele care generalizeazã tabelele Young şi (in parte) procesul Robinson-Schensted.

5 COMPLEXITATEA CÃUT RII ŞI DEMONSTRAŢIILOR PROPOZIŢIONALE

Am inceput cercerãri legate de complexitatea demonstraţiilor in douã scenarii: unul legat de cazul stabil (datorat lui Schrijver) al teoremei lui Kneser-Lovász; celãlalt legat de un formalism introdus de Armin Biere (Linz) intr-o lucrare din conferinţa CAV'2017, care s-a dovedit remarcabil de bun in demonstrarea formulelor (dificile anterior) care exprimã principiul cutiei.

Am obţinut unele rezultate preliminare legate de primul cadru (cel al variantei stabile a teoremei Kneser-Lovász). Rezultatele fac legãtura intre un concept din complexitatea parametrizatã (cunoscut sub numele de *kernelizare*) şi existanţa unor demonstraţii propoziţionale eficiente de tip Frege/extended Frege. Motivul pentru care teorema Kneser-Lovász avea demonstraţii eficiente era faptul cã clasa grafurilor Kneser are o kernelizare formalizabilã logic pentru problema de (ne)colorabilitate.

Rezultatele parţiale obţinute se leagã de existenţa unei astfel de kernelizãri pentru o altã familie de grafuri, cea a grafurilor Kneser **stabile**. La acest moment insã, apreciem, ele nu sunt suficiente pentru a constitui o lucrare publicabilã in literatura de specialitate. Intenţionãm sã le continuãm/completãm in perioadele urmãtoare de raportare.

6 METODE EXPLORATORII IN COMPLEXITATEA COMPUTAŢIONALÃ

Am inceput lucrul in perioada menţionatã la o problemã din domeniul Complexitãţii computaţionale: este vorba de problema separãrii puternice (cu imunitate) a ierarhiei polinomiale relativ la un oracol aleator.

Cercetãrile işi propun sã extindã (cu mijloace moderne) unele rezultate clasice ale anilor '80 din domeniul complexitãţii: Bennett şi Gill au demonstrat in anul 1981 cã P şi NP pot fi separate cu imunitate relativ la un oracol aleator. Pe de altã parte separãri cu imunitate ale ierarhiei polinomiale relativ la *un oracol* sunt cunoscute (Bruschi, Ko), dar nu relativ la un oracol aleator.

Ne propunem sã utilizãm tehnici recente (bazate pe analiza circuitelor booleene) ale lui Rossman, Servedio şi Tang (care in 2015 au separat ierarhia polinomialã relativ la un oracol aleator, o problemã celebrã in domeniul complexitãţii computaţionale).

Aceasta este, din punct de vedere tehnic, cea mai dificilã, dar şi cea cu impactul cel mai mare din problemele studiate. Nu am avut succes, pentru moment, cu tentativa de a adapta tehnicile noi la problema noastrã. Intenţionãm, insã, sã continuãm cercetãrile in aceastã direcţie.

7 Cooperare ştiinţificã şi diseminare

Am prezentat o comunicare pe subiecte legate de tematica cercetãrii noastre la conferinţa SWORDS'2017 care a avut loc la Szeged, şi a avut parte de prezenţa unor cercetãtori de renume precum Miklos Szimonovits (membru al Academiei Ungare, Budapesta), Gyorgy Turan (Univ. Illinois at Chicago, S.U.A.), Rolf Niedermeier (Technical University of Berlin, Germania), Gerhard Reinelt (Heidelberg University, Germania), sau Balasz Szorenyi (Yahoo Research, New York, S.U.A.).

In mãsura disponibilitãţii (de timp cât şi de mijloace tehnice), ne propunem sã contribuim (in anul 2018) la diseminarea rezutatelor proiectului prin realizarea unor videoclipuri cu prezentãri ştiinţifice ale unora din rezultatele obţinute, videoclip-uri pe care ne propunem sã le incãrcãm pe Youtube.

8 Perioada urmãtoare

Pentru perioada imediat urmãtoare ne propunem definitivarea lucrãrilor incepute, trimiterea lor spre publicare, precum şi realizarea unor lucrãri publicabile in proceedings-urile unor conferinţe cu deadline-ul la inceputul lui 2018.

APPENDIX: UNELE DETALII TEHNICE

In această secţiune sunt incluse unele detalii tehnice (algoritmi, imagini) obţinute in cursul proiectului. Cum textul articolelor este in limba engleză şi multe din noţiunile folosite nu au corespondent in limba română, listarea de mai jos reproduce textul (in limba engleză) din lucrare/manuscrise.

Input: A set of intervals I.
Output: A k-chain $J \subseteq I$ of maximum cardinality.
Sort the intervals w.r.t. $\sqsubseteq$: $I = (I_1, \ldots, I_n)$.
For $i := 1$ to n do:

    If $I_i = [a_i, b_i]$ can be inserted into some empty slot

        then

            $J = J \cup \{i\}$.

            insert I_i in the highest-valued compatible slot.

        else

            reject I_i from J.

Figura 8.1: The greedy best-fit algorithm for sets of intervals.

										c_0.396	0.207	0.396	k=2
									0.185	0.126	0.191	0.496	k=3
								0.092	0.072	0.117	0.177	0.539	k=4
							0.047	0.040	0.068	0.112	0.170	0.560	k=5
						0.024	0.021	0.039	0.067	0.110	0.166	0.571	k=6
					0.0124	0.011	0.021	0.038	0.066	0.108	0.163	0.576	k=7
				0.0063	0.0060	0.011	0.021	0.038	0.066	0.108	0.162	0.579	k=8
			0.0032	0.0031	0.0060	0.011	0.021	0.038	0.066	0.107	0.162	0.580	k=9
		0.00163	0.0015	0.0031	0.0059	0.011	0.021	0.038	0.066	0.107	0.161	0.581	k=10
	0.00082	0.00080	0.0015	0.0031	0.0059	0.011	0.021	0.038	0.066	0.107	0.161	0.581	k=11
0.00041	0.00040	0.00080	0.0015	0.0031	0.0059	0.011	0.021	0.038	0.065	0.107	0.161	0.581	k=12

Figura 8.2: Valorile coeficienţilor $c_0, \ldots, c_k$ pentru $k = 2, 3, \ldots$

Algorithm 1: ComputeMultiplicity(k,w)

Input: $k \geq 1, w \in \Sigma_k^*$

Output: $F_k(w)$

```
S:= 0
Let w = w_1 w_2 ... w_n
if w ∉ L_H^k then
    return 0
if w == "k" then
    return 1
for i in 1:n-1 do
    if w_i = k and w_{i+1} ≠ k then
        let r = min{l ≥ 1 : w_{i+l} ≠ 0 or i + l = n + 1}
        for j in 1:r-1 do
            let z = w_1 ... w_{i-1} w_{i+1} ... w_{i+j-1} 1 w_{i+j+1} ... w_{i+r} ... w_n
            S := S + ComputeMultiplicity(k,z)
        if i + r ≠ n + 1 and w_{i+r} ≠ k then
            let z = w_1 ... w_{i-1} w_{i+1} ... w_{i+r-1} (w_{i+r} + 1) w_{i+r+1} ... w_n
            S := S + ComputeMultiplicity(k,z)
if w_n = k then
    let Z = w_1 ...... w_{n-1}
    S := S + ComputeMultiplicity(k,z)
return S
```