SYNASC 2013: Timișoara

J.H. Davenport — J.H.Davenport@bath.ac.uk

23–26 September 2013

Contents

Chapter 1

23 September 2013

1.1 Opening

Rates 49 papers accepted out of 99 submissions, across the five tracks.

Local All main talks in A11 (**not** A02), workshops etc. in the rooms as advertised.

Welcome This evening 18:30

1.2 Model-Driven Decision Procedures for Arithmetic: Leonardo di Moura

We wish to use "logic engines as a service". These are satisfiability checkers, e.g.

$$x^2 + y^2 < 1 \wedge xy > 0.1; \text{ satisfied } x = \frac{1}{8}; y = \frac{7}{8}$$

but if we write $xy > 1$ it is not satisfiable, and we really need a proof.

We have seen the RISE of Model-driver techniques.

- saturation — proof-finding
- search — model finding

Let's take a simpler example — SAT: CNF is a conjunction of clauses, each a disjunction of literals.

Resolution $C \vee l$, $D \vee \neg l \Rightarrow C \vee D$. Expoenetial time **and** space in worst case.

DPLL Split S into S, p and $S, \neg p$. Essentially a backtracking search

CDCL Conflict-Driven Clause Learning combines the two.

Note that in linear arithmetic, we have Fourier–Motzkin, which is saturation, and Simplex, which is Model-finding. Fourier–Motzkin example: $t_1 \leq ax$, $bx \leq t_2 \Rightarrow bt_1 \leq bax$, $abx \leq at_2$ so $bt_1 \leq at_2$.

Now consider polynomials sytems, but $\exists\mathbf{R}$. Look at Cylindrical Algebraic Decomposition. We can regard projection as saturation w.r.t x_n. Lifting (existential) can be regarded as a search technique. Partial CAD $\Leftrightarrow$ backtracking. The point about CAD is that projection guarantees that the sample point is uniform across the cell. **But** projection is very expensive.

NLSAT — Model-based search. This is dynamic, and essentially optimistic search. There are two kinds of decisons in NLSAT.

Boolean

Model constructon of a sample point.

when we fail to find a sample point, this gives us a contradiction beween **two** polynomials, which is the **useful** contribution from the projection phase.

In NLSAT, Lemma learning prevents a conflict from happening again. Some conflicts can cause *non-chronological* backtracking (i.e. jumping back several layers).

Note that we need a lot of the CAD machinery: polynomials etc.

Data — among previous methods, note that CAD-like ones far outperform the heuristic-based ones in terms of completeness. Mathematics is by far the best of the CAD ones. Our procedure does nearly as well as Mathematica, but much faster.

We can view Gaussian elimination as saturation! Also congruence closure.

MCSAT — Model-driven SAT. Trying to lift ideas from CDCL to SAT. Given $x^2 + y^2 \leq 1$ we can project to $-1 \leq x \leq 1$. This can also be thought of as incremental Fourier–Motzkin. This general paradigm works if every "finite" theory has a finite basis. Always true for finite theories. This is in Z3, only 7KLOC, but outperforms many others. `http://z3.codeplex.com`.

Q How important are algebraic subroutines?

A Vital— when Mathematica outperforms us, it's because of better algebra!

1.3 Davenport

See `http://staff.bath.ac.uk/masjhd/Slides/SYNASC-2013v2.pdf`.

1.4 Watt

[SMW used JHD's laptop, so JHD couldn't take notes]

He had 60K symbol examples, assigned manually into 388 classes. For each class, he assigned determining points by hand to one symbol. Then used 30-step homotopy on arc length of the symbol to assign these to the others on the same

class,followed by manual eye-balling (0.45% failed, being "bad examples"). Then regard this as "ground truth", and ask how many steps are needed to recover this. Details in the paper, but typically 2–4 steps suffice.

1.5 Sasaki

Physical Laws plus Constraints give differential equations to which we apply differntial elimination, arrange to be sequentially soluble, then solve. Note that we convert ill-conditioned DAEs into ODEs. But the model is large, so we get cancellation erro (this talk), rounding error and loss of trailing digits.

Gröbner methods (if we restrict to leading-term elimination) have several error-avoiding methods. For differential elimination, we propose a new method. These systems have many numeric equations, and these are very sparse. We propose a numeric=symbolic method with two error-eliminating methods, to combine with pivoting.

Consider $C\mathbf{x} = \mathbf{b}$, where $C \in \mathbf{F}[\mathbf{p}]^{m\times m}$ is a matrix over the parameters $\mathbf{p}$. In we partition the matrix into (numbers only — parameters) then elimination in the numeric part does not increase the parametric order. In general we note (one-step) Bareiss–Dodgson elimination, but if $C^{(k-2)}_{k-1,k-1}$ is small, we get large errors. We can also look at the two-step method, but this produces error-blowup later on.

- separate-pivot method: replace $C^{(k)}_{k-1,k-1}$ by $\mathbf{q}\cdot C^{(k)}_{k-1,k-1}$
- local 2-step which lets us exchange two rows to get a larger pivot.

Example had 10^4-level cancellation, but was able to eliminate most of this, by either of these two methods, the second being slightly better.

1.6 Knot theory of regular polygons — Ida

Computer-assisted construction and verification.

Rigor is a pre-requisite for security and safety. Related to, but not the same as, precision. However in geometry, we often appeal to intuition. Non-degeneracy conditions therefore tend to be missed. Today, we are not looking enough at the constructive aspects of algebaric geometry. We cannt prove that a given shape is a regular pentagon by inspecton, or by measuring, evne though we can improve precision with better tools.

Question: how does one knot an "Obi" — broad sash. Folding consistes in determiing the foldline, folding along it, and making a crease.The determination in origami is does by superposing points or lines on eahc otehr, which is *not* what goes on with knot-folds. Hence Huzita's fold principles.

O1 Fold along the line passing through P and Q where $P \neq Q$;

O2 ...

O3 ...

O4 ...

O5 ...

O6 ...

Note that the origami constructable numbers are **Q**(sqrt, cube root). [Cox, Galois Theory]. Showed 13-step Mathematica construction of regular pentagon. But where's the proof? 4.5 seconds via Gröbner bases.

Q The script has distances in it, how are these determined?

A These are just used for the drawing not in the proof.

A more interesting construction is the "double isosceles lemma" — took 0.3 seconds. Showed 3-D visualisations of the knot. Had a 31-item Gröbner basis after elimination (by hand)of some redundancies.

Can also construct the heptagon, and $(2n+1)$-gons in genral, similarly. Aslo squares, but the hexagon cann't be one in a space-filling way. The Dodecagon is not perfect — not wholly covered (hole is very small, but present)

Q-JHD How much computation does going through the "double isosceles lemma" save?

A It's really for my benefit: doesn't really help in computing time terms.

1.7 McGrail

> Implementation of the Solution to the Conjugacy Problem in Thompson's Group F.

Piecewise linear homeomorphisms $[0,1] \to [0,1]$, every slope is a power of 2, and eery breakpoint is a dyadic rational. Small generating set. Can draw tree diagrams of splits in codomain and domain, turn codomain upside-down and marry them, hence a strand diagram. Shows for generatord and inverses. To invert a strand diagram, invert every arrow, and turn diagram upside-down. Concatenation of strand diagrams is the composition operator in F. Unifying the roots of the two trees gives us the annular strand diagram (ASD). There's a confluent rewriting system on the ASDs

Two ASDs are isotopic is one can be obtained form the otehr by a continuous map of the annulus to itself.

Theorem 1 (coauthors) *Two elements of F are equivalent if the reduced ADSs are isotopic.*

Theorem 2 *The connected reduced ASDs can be encoded into planar graphs which are isomorphic iff the ASDs were isotopic.*

Hence an algorithm.

1. Produce strand diagrams (linear)
2. Annularise (linear)
3. Reduce (each reduction takes constant time and there are $O(n)$ of these, so linear).
4. label connected components — breadth-first search.
5. Encode to planar graphs — there are nine different classes of edge.
6. check isomorphism — [HW74], but this has never been implemented, so used brute-force breadth-frst search which is actually $O(n^2)$.

1.8 Filuppi

On ladder operators for little q-Jacobi polynoials and theor generalizations

Orthogonal polynomials satisfy a three-term recurrence. Consider $D(\sigma w) = \tau w$, where for classical weights, $\deg \sigma \leq 2$, $\deg \tau = 1$. "Ladder operators" as terminology comes form quantum mechanics. From $DP_n = A_n P_{n-1} - B_n P_n$ where A and B are given by integrals fo the P_n.

We do not have a complete algorithm, but need clever grouping of terms

1.9 Zając

Classification of Signed graphs

'+' on an edge means "friend" and "−" means 'enemy'. We consider loop-free bigraphs, and don't allow two connections with different signs between A and$B$. This gives us an upper-triangular Gram matrix $\check{G}$, with the signed number of edges in the entry (1 on diagonal). The Coxeter matrix is $\check{G} \cdot \check{G}^{-1^T}$.

Can construct complete bigraphs from these (how?). Her constructions produces graphs of corank $n - 1$. There are 1.8M such bigraphs on six nodes, so hard to analyse.

1.10 Gąsiorek

Efficient Computation of the Isotropy Group of a Finite Graph.

What properties of a graph can be read from the spectrum? Graph isomorphism is about permutation matrices between adjacency matrices: what about other matrices? A bigraph has two kinds of edges, with a non-symmetric Gram matrix

G as before. Symmetric matrix $G_\Delta = \frac{1}{2}(G+G^T)$. These related to simply-laced Dynkin diagrams.

A, C are $\mathbf{Z}$-congruent if there is a $\mathbf{Z}$-invertible B with $C = B^T AB$. Concept of matrix morsification. If we have two positive bigraphs with the same Coxeter polynomial, what can we say?

The Isotropy group is the set of $\mathbf{Z}$-invertible B such that $B^T \cdot G_D \cdot B = G_D$.

1.11 Jeffrey

> Recent results concerning exact row reduction of matrices.
> What should I do with these results?

Fraction-free methods [Bar68] "using results known to Sylvester".

1.11.1 New Material

Exact Integer or Polynomial matrices. [GCL92] — "common knowledge" that you should always choose the largest pivot, but for exact should always use the smallest. No backing provided. $A = PLD^{-1}U$ (P permutation, L lower triangular, D diagonal and in the ring, not quotients and hence postponing the divisions, U upper triangular).

Take matrices of random integers ≤ 11, or polynomials of degree ≤ 3. Pivot with largest U^l and smallest U^s. There is a strong numerical confirmation of the [GCL92] remark, both for integer and polynomial matrices.

1.11.2 Paper Material

Do rows have a common factor? Naïve answer is "no". small example shows that they do exist. Let D_1 be what we get after dividing out as much as possible. $||D||$ and $||D_1||$ grow quadratically with n, but $||D|| - ||D_1||$ grows linearly.

Q–JHD See [Dod66]. I am surprised as well, and you should publish pivoting reuslts in CCA

A Thanks. Also what about "pivoting for sparsity"?

1.12 Felisiak

> On Cexeter-type classification of loop-free edge-bipartite graphs and matrix morsifications

So we have A_n. D_n and E_6, E_7, E_8. $\Delta = (\Delta_0, \Delta_1)$. Symmetric Gram matrix G_Δ as before. $\mathrm{Cox}_\Delta = -\check{G}_\Delta \check{G}_\Delta^T$.

Used Maple, and Eigen and GSL – Gnu Scientific Library.

Proved that two matrices with the same Coxeter polynomial and determinant with $n \leq 7$ (JHD thinks this is the correct statement). For example, E_7 has five bigraphs.

1.13 Fortiș

> Dependence of the oscillatory movements on an unmanned aerial vehicle on the forward velocity

There are aircraft models as a system of DE with initial and boundary conditions. We use GAM (Generic Aerodata Model) 12 DoF models originally from Saab, now in Modelica. There is ALFLEX (Automatic Landing Flight Experiment) — Japanese. ADMIRE (aero-data Model in a Research Environment) simplifies GAM to 6 DoF.

We need a safe interval $[\underline{\delta_\epsilon}, \overline{\delta_\epsilon}]$. Previous research has only covered a constant forward velocity (84.5 m/sec). We are looking at [79.120]. Need to track the evolution of the Euler pitch θ, also angle of attack α. Successfully fitting a quadratic to numerical data.

1.14

> Plana stretching flows with partial slip

Nonlinear differential equation $u'''(\eta) + u(\eta)u''(\eta) - (u'(\eta))^2 = 0$. Navier-Stokes still applies, but the boundary conditions are different. Use a homotopy method. We have a different approach from the usual one: regard the parameter as governing the non-linear part. This gives us (apparently) a linear system, and hence an expression for the nonlinear part.

Used 4th order Runge-Kutta (shooting) in Mathematica 6.0. We obtained the convergence parameters by Galerkin. The approximate solution approximates the numerical solution very closely.

1.15 Extended Precision in MatLab: Jeffrey

> A lot of our students will avoid C, C++ etc., and only program in MatLab.

Define a Matlab class (and hence the user doesn't need new syntax) called `ddreal`. Note this is not arbitrary precision, like GMP or ARprec, and isn't intended to replace Maple etc. either. My own use was to computer a special function W. This has a square-root singuarity at $z - 1/e \approx -0.368$. If the usr inputs u near z, we need $u - z$ to double precision. Since ths is a square root singularity, need twice as many bits. Also iterative refinement of linear systems.

Has also used this for demonstrating calculus, e.g. $\sin'(x) \approx \frac{\sim(x+h)-\sin(x)}{h}$

```
h=ddreal(1e-10)
x=ddreal(1)
der=(sin(x+h)-sin(x))/h
```

Standard "double double representation", e.g. $\pi = \pi_u + \pi_l$. In Matlab native code, we get speed, so use as far as possible.

```
s=sqrt(x.Hi)
sout=0.5*(s+x/s)    % type contagion to ddreal
```

Unfortunately, needed to inline a lot of the class uses internally — painful but gets speed, since class dispatch is relatively expensive. Notes that he uses different algorithms with more precomputation — memory is cheap. In computing exp, store $e^{p\Delta}$ with $\Delta = 2^{-8}$ so that he can use Taylor in highly convergent regions.

Chapter 2

24 September 2013

2.1 Autonomic and SLA aware Cloud management: Brandic

Use case: "smart cities", especially traffic. On the netaive side, spam. Also aging/care. But CT accounts 10.5% of Germany's energy consumption, and accounts for 2% of the worldwide CO_2 consumption. Hence a major question

> Are clouds the energy-efficient solution to Big Data Management

Physical machines/hypervisors/virtual machines (abstractions). Hence can move virtual machines "on the fly" to data centres where the energy is cheap, or where cooling is cheap. This requires *federation*. Claims that 'cloud computing' is becoming the fifth utility. However, this is all very ambitious.

1. Automate the cloud management. Note that a humanbeing's own body is an autonoic, self-managing, system. Note that, if we regard each human as such, they negotiate SLAs between themselves. Claim there's a MAPE feedback loop: Monitoring, Analysis, Planning, Execution, supported by a Knowledge base/manager. Note that one can overkill on negotiation — I negotiate over a house, but not a bottle of water. That's a commodity, and cloud computing should be like that. Quotig an example of an SLA request, with MIPS and MB of memory (also GB of storage), Mb of bandwidth (in/out) as parameters. But SLAs offered might be different. Hence we need some metric for how "close" SLAs are, and then macth them.

2. Have build a "cloud simulator" (Tu Wien + KIT Karlsrühe) to test this and measure net utility (gross utility less cost). Various clusterinng approaches all outperform naïve. This is at a high level. How do me translate hardware monitoring from the hardware level to the cloud SLA. Claims the solution is a "run-time" monitor, with access to SLAs, hardware data and mapping rules. A good example of a single rendering application, but

running three different scenes, with different time/load profiles (ranging from uniform to chaotic). However, claims "CPU time is very generic".

3. Speculative approach to SLA Management. Need to grow/shrink resources allocated to VMs.This process also needs monitoring.

Q–JHD I am not sure that this, especially MIPS being generic, works at the HPC level, where performance dpends critically on things like cache availability.

A Discuss offline.

Q Can we work at the application level, rather thanthe hypervisor level?

A Not our research direction. We want to be application-independent.

Q Querying clusters of SLAs? I would think a range of SLAs would be desirable.

A We are not limiting the SLAs, rather the set of tools for expressing the SLAs.

Q How many parameters might an SLA have? Inhigh-dimensional space, clustering is dubious.

A typically less than ten, so I don't think is a major concern. This figure is based on real SLAs from cloud priviers in the marketplace. Note that "time to decide" is also a concern.

2.2 On-line scheduing on identical machines with migration: Ehlers

Off-line scheduing is strongly NP-hard, but there are good approximations. The simple on-line schedule is 'list scheduling": $\frac{1}{m}\sum P_j + P_{\max}$ with m machines.

Given a new job j with makespan μ. Aim to add the job with minimal impact on makespan. Let $LB := \max(\frac{1}{m}\sum P_j, P_{\max})$. A job is "small' if its size if $\leq \epsilon LB$. Also round job sizes to multiples of ϵ, which increaed makespan by ??.

- Small job just add greediy.
- A large job arrives and there's room for it not increasing the makespan — do it!
- A large job arrives into an efficient schedule. Set up an Integer Linear programming problem. A contains all possible patterns with makespan $\mu' = \max(\mu, 2LB)$, b contains all large jobs in current schedule, and b' is new solution.

Theorem 3 ([Cook *et al.*]) *Given $Ax = b$ and $Ax = b'$ both having integer solutions, then $\exists y$ with ...*

Choose at most $O(1/\epsilon)$ new jobs. Solved in $2^{O\left(\frac{1}{\epsilon^2}\log^3\left(\frac{1}{\epsilon}\right)\right)}$. Actually iterate on desired makespan. This process rearranges the large jobs, then solve the small ones with bin packing.

Q Your job-size is only time. What about memory?

A Harder!

2.3 Algebraic Properties of Generalized Multisets: Ciobanu

Multisets are a representations of resources: $X \to \mathbf{N}$. We are thinking of $X \to \mathbf{Z}$. Set theories; ZF or ZF+AC, NBG (manages proper class), ZF+Agents (name) and Fraenkel–Mostowski, which manages the concept of "fresh names". Developed in 1930s to prove independence of AC. Rediscoved in 2000s by Gabbay/Pitts. Also known as "theory of nominal sets". n general, this reapces "finite" by "finitely supported".

Theorem 4 $\mathbf{Z}(\Sigma)$ *is a free abelian group.*

Direct sum of indecomposable subgroups.

Note that Loeb had a view of subsets of multisets which did not allow "passing through 0", i.e. $X \to \mathbf{Z} \setminus \{0\}$.

We have results for Riesz type, and the three Isomorphism theorems. Also Cayley-type results. Can place a total order on $\mathbf{Z}(\Sigma)$. Using the FM-setting means that we add works like "finitely-supported", and "set"→"nominal set", and we actually recover Loeb's order.

Q Is there an application to database logs, where we have deletion as well.

A

Q–SMW Applications in symbolic computation, where multiplicities can be positive or negative.

A

2.4 Algorithmic experiences in Coxeter spectral study of P-critical edge-bipartite graphs and posets: Polak

Bigraph is $(|Delta_0, \Delta_1)$ with Δ_1 partitioned into $\Delta_1^+ \cup \Delta_1^-$. Gram matrix and symmetric Gram matrix. No loops, but diagonal entries. Δ is positive iff G_Δ

is positive definite. Δ is P-critical if it is not positive but each $\Delta^{(a)}$ (remove vertex a) is positive.

w is a root of Δ if $wG_\Delta w = 1$. We can construct P-critical bigraphs as one-point extensions of positive bigraphs. There are questions of equivalence. Have a theorem for $3 \leq n \leq 6$ showing that there are suitabe conditions to show that $\Delta \approx \Delta'$.

2.5 Bounded Bi-ideals and Linear Recurrence: Bērzina

A is an alphabet, with λ the empty word over it. $|u|_v$ is the number of occurrences of word v in u. $v_0, \ldots$ is a bi-ideal sequence if there is a sequence of workds $u_0, dlots$ with $v_0 = u_0$, $v_{i+1} = v_i u_i v_i$. Hence bi-ideals and FG bi-ideals

2.6 Token-free bounded delay codes and hash iteration: Dițu

Want to use token-free bounded delay codes to improve cryptographic hash functions. Noted Pre and variants ePre, aPre; Sec and variants; Coll. See [RS04].

If we use token-free codes in pre-processing, can respect only two of the seven. We can now do Coll, Pre, aPre and ePre. This beats HAIFA, and indeed everything except ROX, which only works in the random oracle model.

Use $\in$ to means "is a subword of". a code is m-token free if there is a word of length m which is never emitted. For practical reasons, we want bounded-delay codes. If C is m-token free with witness μ, replace m by $\mu||C(m)$. This process is actually iterated (JHD lost the details).

Q Do you have to read all the input, as the presentation implied?

A No.

2.7 A Systematic Approach to Computations on Decomposable Graphs: Ravve

Example — Map of Timișoara with river and ()for simplicity) two bridges.$G_i = (V_i, E_i, P_i, Q_i)$ where the last two are one-place relations (colourings) of the graphs. Define $G := G_1 \leftrightarrow G_2$ to have vertices $V_1 \cup V_2$ (disjoint) and coloured rules for uniting edges (too complicated to wite down). G has a cycle if either half as, or if there's a suitable one across the union.

How do we generalise this to multiple components and colours? Use Monadic Second Order Logic. In particular, can we do this is a distributed parallel way?

If we actually start from G, does this decomposition help? There is (I dodn't see quite hwere) a proble of reconstructing the solution from the components.

Can make gains by pre-/re-computation if we make small changes to components and or addition/deletion of components.

2.8 Specify and Verify Your Language using K: Rosu

We want a formal language definition from which all tools can be derived. So consider a programming language L.

- Formal semantics — probably skipped.

  ```
  int main()
  { int x=0;
      return (x=1)+(x=2);
  }
  ```

  GCC3, icc etc. give 3, GCC4 gives 4, some formal tools "prove" 4, C standard says "undefined"

- Formal syntax — more likely, but how useful.

- If you need a PhD to define a langauge, we have failed!

Semantics of K are best understood in terms of graph rewriting. "Double pushout gives true concurrency in the presence of configuration sharing".

Two SDF-based parsers generated, one for programs (concrete syntax) and one for semantics (abstract and concrete). Backends: Maude+Z3 (most features), LaTeX, Java+Z3 (prototype). We have defined Java 1.4, Java 7, Verilog, Python etc.

We have a semantics for C. Configuration is a Heap with 75 cells, and c. 1200 rules. Tested on thousnds of examples from IOCCC to GCC torture test, passing 99.2% so far. GGC 4.1.2 passed 99%, Clang 98.3%

We propose to build semantics starting from (trusted) operational semantics. We take these as axioms, and derive reachability rules etc. This is sound and relatively complete.

Use "Matching Logic" for Static Properties. Configuration terms with variables ar allowed to be used as predicated, which are patterns, and satsifaction is matching. Claims that this produces Hoare triples.

See `http://www.kframework.org`, which has various examples. The "configuration" would contain the 'mathematical domain', e.g. list, reverse etc.

The conclusion is that one can define the range of tools required form the single axiomatic definition.

2.9 Local Rank Distance: Ionescu

We anotate patches, rather than pixels. Note that NPL uses n-grams, but DNA uses m-mers. Define a Local Rank Distance bounded by m of two strings. Two experiments

1. Used Mitochondrial DNA from 22 mammals from 7 orders. Used 2-mers ... 8-mers. Only 8-mers gives the "correct" answer, categorising humans with other primates.
2. TOEFL11 corpus for non-native English Used $k(s_1, s_2) = e--LDR(s_1, s_2)/\sigma^2$ as the similarity measure. LRD only gives 42%, but normalised LDR is 70+%, whether we use 4, 6 or 8-grams.

Still want to understand the theory of this metric, and also to use Hamming distance rather than exact equality.

2.10 From Plagiarism to Malware Detection: Oprișa

Students borrow code, but

- Comments and formatting
- α-conversion
- swapping parts
- deleting of high-scoring parts.

Malware also changes:

- Functional changes
- Obfuscation changes.

Student work is source code, but malware is binary. Let Σ be the alphabet of opcodes, and regard only the opcodes in the binary.

Shannon's entropy $E(S) = -\sum p_s(s) \cdot \log p_s(s)$

Kolmogorov uncomputable, but compressibility is a proxy

so $1 - \frac{\cdots}{\cdots}$

Common n-gram similarity. In fact should weight by frequency.

946 homeworks from 100 students. Precision P, Recall R, and $F = \frac{(\beta^2+1)PR}{\beta^2 P+R}$. Weighted n-grams did best, beating Moss.

Q Did you try learning the weights?

A No. Hand-tuning.

2.11 Clustering Data streams using Mass Estimation: Sorin

One problem is that data streams evolve. Mass estimation is an alternative to density estimation. Can do this accurately (too espensive) or via random approximations. $\overline{gmass}(x) = \sum_i h : dtree(mass(d_i))$.

2.12 Malware detection

Antivirus vendors used generic detection techniques, but as malware increaed when to machine-learning. But current malware is site-polymorphic (self-updating). We have a one-sided model. 34353 prevalent malicious examples. 1934296 genuine samples. 14985 features collected per file. We selected 250 for our model. Best to train over two months data (and repeat every month). 35 False positives (0.15%). Note that false positives are extremely expensive here.

2.13 Self-organising maps for fraud prediction at online auction sites: Almendra

Note this is rare $< 1\%$, so many detectors are not suitable. Outlier detection is not easy, since we are dealing with (malicious) fraud. Also can't afford too many false positives. We use self-organising maps, but a 3×3 SOM really did not do enough discrimination by JHD's eyes, but he could produce a subset of the clusters which kept false positives under 25% (but still much more likely than true positives).

Would like to test his methodology on other unbalanced data bases. Also to do some dynamic splitting of classes.

2.14 Reducing the numner of useless reductions performed by constraint solvers based on AC-3

[Presented by a colleague.]

Many real-life applications of constraint solvers. A "solution" is a complete consistent assignment of values to the variables. We are developing OmniCS. Follows JSR 331 (Constraint Solving API) guidelines. x is arc-consistentcy if, for all values a of x, there is a value of y such that $x = a, y = b$ is consistent. b is the *support* for $x = a$. We wnat to maintain arc-consistency, hence the AC-3 algorithm. A key to this is the `Revise` pass. The authors proposed a better version [ASB09], and here we have a generalisation. For example, if the operator is $x \neq y$, we only revise if y has a singleton domain. Experimental results show a 30% reduction in useles revisons, and a massive one in the case to $\neq$.

2.15 Should we beware of exceptions: Eclipse project

Recounted story of how a ne user was confused by "out of disc" exception. Tried to correlate "exceptions", "structural metrics" (including number of catches, number of throws, etc.) and "defects" (pre- and post-release) in Eclipse. Measured this on a per-class basis.

Prior work has shows that classes that use exceptions are more defect-prone. [Mar, IWPSE 2011]. Three research questions.

1. Do classes that use exceptions are more complex than other classes? Spearman's rank correlation shows high correlation with LOC etc.

2. Are the metrics based on exceptions better or worse than those on complexity? Numbers looked identical.

3. Are the classes that handle exceptions improperly better/worse than those that handle them properly? 31% of Eclipse classes do not handle exceptions properly! The "odds ratio" for proper/improper handling shows consistently (across three releases of Eclipse) 2.1 for pre-release defects for all releases., but 2.4 ... 1.6 for post-release defects.

Q–JHD Is the post-release issue a question of the fact that older releases have been in the field longer?

A I took post-release defects 6 months after release for consistency

***** An interesting point. One could still wonder about relative speed of takeup etc., but that was a nice answer!

2.16 Parallel Data Acquisition for Visualisation of Very Large Sparse Matrices— Langr

"Very large" implies processed on distributed memory systems. But the user does not know when his matrix will be being processed by the cluster. The memory is too large to ship to the user's workstation. We could store it on a cluster file system, but this is often the weakest link. Shows big current clusers, taking 20–100 miutes to write the matrix to disc, and look at the wasted core hours! Client-server doesn't work through asynchronous issues. Hence we can't work with the original matrix.

So work with the image only? This is straightforward, but any problems with the image cannot be rectified. Hence this involves a lot of guess-work. Hence we process the matrix into "visualisation data" and load this into the user's workstation, with which the user can interact. In detail, we partition the matrix into blocks, and use various visualisation functions (densities, average magnitude etc.). Showed contrasting images of these two. The user also

specifies how big the answer should be. We have an experimental implementatoin in C++/MPI. Use a 5440-core 5.44TB cluster QueenBee/LONI[1]. See a "Scalable Parallel Generator of Sparse Matrices". Run tme typically 10 seconds independent of number of cores. Will have a simple C/C++ (/Fortran) API.

Q Use cases

A Nuclear physics (very abstruse). They are especially interested in load balancing. But we've presented artificial, scalable, test cases.

2.17 The study of the impact of matrix processor mapping on the parallel sparse matrix-vector multiplication: Simecek

Let A be an $n \times n$ matrix with N nonzeros. Assume Compressed Sparse Row format. Then $\Theta(n \cdot N)$ sequentially. Assume P processors with $P << n$. Previous studies have only had small P, or special matrices. Divide the matrix into regions, and assume each processor only stores one region, and computes region $\times$ part of vector. Four quality metrics

q_1 load balancing

q_2 memory usage: $\frac{\max(\text{memory in processor } i)}{\text{total memory}/P} \geq 1$

q_3 time of transformation

q_4 time of operation (including all communication and reduction)

1. Fixed rows
2. Dynamic Rows according to
3. Chequerboard
4. Dynamic Chequerboard
5. adaptive
6. k-d tree based on #nonzero
7. k-d tree based on memory requirements (not the same as the previous because the vector part also has to be stored, and the result).

Paper computes upper bound (worst case scenarios). In practice, 6 has best load balancing and 7 best memory usage.

In practcie, we often do many matvecs, so the redistribution cost may not be an issue.

[1] Louisiana Optical Network Initiative.

2.18 SecuringCommnication ina peer–to–peer mesage middleware: Szilágyi

AMQP or 0MQ (scalable but insecure). We are working within a cloud, so must allow for insider attacks. However, I assume that my own nodes within the cluster are secure.

> Unintelligible transmission with a verifiable originator and unhideable tampering.

Use Diffie's Station-Station key exchange, essentially DH with keys on the signature. Suggest to expand the master key thus obtained by HKDF into 4KB of key data. Convert AES into a stream cipher, e.g. in counter mode.

But what about forwarding nodes? They don't need to decrypt the messages, so split between message and header, and forwardding nodes don't decrypt the message. Faster for large ($> 4KB$) messages, up to a factor or 5.

-
-

Chapter 3

25 September 2013

3.1 Data Mining and Compression: Simovici

Data Mining is very expensive — preparing and cleaning the data can account for 80% of the total. So we need to evaluate whether a data set is mineable. I claim that the answer is compression, since data that contains patterns is more compressible. Note the distinction between lossy (e.g. JPEG) and lossless (e.g. LZW) compression. We are only interested in lossless compression.

Compression [Keogh2004] was used in developing parameter-free data mining. Note that the more parameters you have, the more you can extract.

[CilibrasiVitanyi] defined a pseudo-distance

$$d(x,y) = \frac{C(x||y)}{C(x)+C(y)}$$

where C is the length of the compressed dataset. Least value 1/2. Note that this is **not** a metric, in particular not triangular.

Let $n_x(w)$ be the number of occurrences of string x in w. The *prevalance* of x in w is $\frac{n_x(w)|x|}{|w|}$.

Experiments as prevalence of `001` in a random string increases, the compression ration tends to 0. Consider the Thue–Morse sequence:

$$s_i = \begin{cases} 1 & \text{if } i \text{ has an odd number of 1} \\ 0 & \text{otherwise} \end{cases}$$

, also "obtained by starting with 0 and successively appending the Boolean complement of the sequence obtained thus far" [Wikipedia]. This is cube-free, i.e. no www occurs. For small k, S_{2^k} compressed very badly (1.226 at $k = 10$) but well for $k > 20$ (less that 0.01).

Generative grammar: $A_N \cup A_T$ with non-terminal and terminal symbols. The more complex the grammar, the less compressible. Let L_n be the words of length n in this grammar, in lexicographical order. Take $L_1\{ww|w \in \{0,1\}^*\}$ (context-sensitive)) and $L_w\{ww|w \in \{0,1\}^*\}$ (context-free). L_2 compresses

better. $L_{prime} = \{a^p | p \text{ prime}, a \in \{0,1\}^*\}$ needs 42 productions, but $L_{exp} = \{a^{2^i}\}$ can be done in six, and the compression ratios are very different.

Consider adjacency matrices of a graph. Let V be a 0,1-matrix-valued ranom variable. let $\mathcal{B}$ be a random variable $\begin{pmatrix} B_1 & B_2 & \dots \\ p_1 & p_2 & \dots \end{pmatrix}$, B_i a random matrix and p_i probabilities. $A \otimes B$ is the matrix each element of which is $a_{i,j}\mathcal{B}$. Applied to adjacency graphs, this lets us build graphs with structures. The Shannon entropy of the p_i is closely correlated with the compressibility of $A \otimes B$ in experiments.

Principal submatrices of A are the adjacency graphs of subgraphs. Has a formula for the "entropy of the subgraphs". This also turns out to be correlated with compressibility.

Market basket data sets are very common in data mining. Note that a supermarket may have 10^4 items on shelves. Represents as sets (N.B., not multisets). $\text{conf}(X \to Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$ where supp is the support of a set, and conf is the confidence in this correlation ("people who bought X also bought Y"). Compressibiliy indicates the number of useful rules.

Current work shows that compressibility of genomic regions is a useful tool for detecting areas where the gene replication mechanisms are disturbed. This is an indicator for certain cancer types.

Q Did you find counter-examples?

A No — compression always indicates patterns.

Q How useful is $d(X, Y)$ for graphs?

A "Frankly, it stinks". My statement is about the similarity of compression ratio of the dstribution of subgraphs of graphs (principal submatrices), not about the compressibility of the concatenation.

3.2 TiMo in Timisoara: Ciobanu

TiMo - "Timed Mobility". The formal approach comes from Pi-calculus. Location + mobility + interaction + timers. Local interaction and local clocks. Migration is non-urgent, whch models network delays. Special symbol $\textcircled{\scriptsize 9} P$ for "stalled process P". "$a^{\Delta t}! < v >$ then P else Q" will send v on channel a (if permitted).

Various results defining semantic consistemncy. They can be translated into Petri nets with time constraints, whose transition system is strongly bisimilar with the TiMo. There is a notation for access permissions, and these are preserved by the operational semantics of TiMo. Example of person going to shop, agent going to shop, and transferring the (information about?) price of a flight.

PAT is an extensible framework for domain-specific model-checkers which we have used. Operations such as "bounded liveness" can be modeled. Optimaity checking for the system is also possible, but involves far more states/time.

pTiMo (probabilistic) aks "what is the probability of reaching state s_1 before time t_1" rather than TiMo's "can we reach?".

rTiMo, using a real-valued global clock, lets us do "real time" work. A link with timed automata allows model checking by UPPAAL.

3.3 An Introduction to Modern Symbolic-Numeric Computation: Watt

Note that this is a tutorial!

Note that symbolic computation data can grow: even if we $m \times n$ matrices, and m and n don't change, the entries can grow, e.g. in $\mathbf{Q}$. One solution is homomorphic images, and another might be floating point (inside symbolic objects). **But** the usual algorithms just don't work in this setting. In particular "until $x = 0$" doesn't work, and "until $|x| < \epsilon$" isn't that much better. Note that floats are not associate, or distributive. Note that floats don't have exactness but do have nearness.

[Corlessetal] introduced condition number etc. as fundamental numerical concepts into SNC. How much the output changes as we change the input — independent of the *method* used ot compute the output. Stability, on the other hand, *is* a function of the method.

Adding or multiplyng two polynomials — perturb the inout and we perturb the output. This is *not* true of greatest common divisors. Assuming we make the polynomials monic, (f, g) has $m + n$ degrees of freedom. For $h = \gcd(f, g)$ of degree 0, there are enough cofactors. If $\deg(h) > 0$, there aren't enough, and we are in a space of positive codimension. Hence the problem is intrinsically ill-posed.

3.4

Q1 [Sch85]. Find h such that $|hf_1 - f| < \epsilon$, $|hg_1 - g| < \epsilon$ and any other quasi-gcd divides h. Forward error.

Q2 Backward error analysis. Let p^* be the vector of coefficients of p, and $x^* = [x^n, \ldots, x, 1]$ and $p = p^*.x^*$. Polynomial multiplication via Cauchy Matrix. Note the Sylvester Matrix: linear combinations of rows of $S(f, g)$ are polynomial combinations of f and g. But, of course, the gcd divides all combinations of f and g.

L_2 norm for matrices and vectors. M transforms the unit sphere to an ellipsoid. The lengths of the semi-major axes are the singular values of the matrix, $M = U\Sigma V^T$ where U and V are orthogonal.

Theorem 5 *σ_k is the 2-norm distance to the nearest matrix of rank less than k.*

So when σ_k is small enough, we have a rank deficiency big enough.

Q3

Q4 Closest polynomails with a non-trivial gcd.

Q5 Largest gcd within ϵ

3.4.1 Image Processing

[LYZ10] states that approximate gcd of bivariate polyomials corresponds to image resolution. $O(n^2 \log n)$ versus $O(n^8)$ prior art. Initial matrix P, blurring U and noise N, and the polluted matrix is $P * U + N$. Suppose we have two distorted images: $F_1 = P * U + N_1$, $F_2 = P * V + N_2$. Under the Z-transform, this becomes

$$\begin{aligned} f_1(x,y) &= p(x,y)u(x,y) + n_1(x,y) \\ f_2(x,y) &= p(x,y)u(x,y) + n_2(x,y) \end{aligned}$$

which looks like, and is, a high-degree approximate gcd.

In fact, apply this technique to R/G/B of a single photo, all of which have the same blurring but different noise.

$$\begin{aligned} f_R(x,y) &= p_R(x,y)u(x,y) + n_R(x,y) \\ f_G(x,y) &= p_G(x,y)u(x,y) + n_G(x,y) \\ f_B(x,y) &= p_B(x,y)u(x,y) + n_B(x,y) \end{aligned}$$

but now we have a low-degree gcd.

3.4.2 Also

Approxiate square-free decomposition, approximate factoring. Approximate polynomial decomposition is challenging: $(f + \Delta f)(x) = g(h(x))$. Same dimension problem as gcd, so backward error analysis is the right idea.

3.5

Arithmetic Coding Base has good space complexity, but poor time complexity.

Therefore MBT format (Minimal Binary Tree), or MQT. These are actually only mnimal if we assume a fixe dnumber of bits, and CBT (Compressed Binary Tree) is better. These formats can be applied to sparse matrices of any structure. But how do we parallelise?

Applie to matrices with $n \approx 5 \cdot 10^6$ and $N \approx 5 \cdot 10^7$. Clains its < 10

3.6 Cooperative CPU/GPU Processing

Note that the GPU can't address the same space as the CPU.

three stages to render an object. Using a GPU, run-time seemed to be independent of the number of Mandelbrot iterations, whereas on CPU, independent of the number of cores, it was almost linear in this.

1. Computed on CPU — ray cast for eah pixel
2. Computed on CPU or GPU — Apply Mandelbrot to compute the pixel colour
3. Computed on CPU — render result

This was one piece of code, the difference being the thread despatcher for the Mandelbrot code.

3.7 CT Toolbox: Bardino

From the e-science centre at Copenhagen, working, among others, on CT scanners. CT reconstruction is 3D from a variety of 2D recordings. Wanted to reconstruct a pig in 5 seconds, whereas medical scanners normally take 15–20 minutes. We (consortium) actually made our own scanner (audience surprise).

Used this as an assignment in parallelism course at the university. One project went from hours in MatLab to seconds this way. Note that `float` will do for these calculations.

```
for each 2D image
  convolute projection
  for each voxel
    calculate contribution of voxel to image
```

We built the toolbox in Python. Use NumPy for vlidation, and PyCUDA for performance on the GPU: this combination really worked for them in terms of development time. It's therefore multi-platform code, under GPLv2.

CPU compute-bound

GPU limited both by the I/O bandwidth and the problems of getting the data to the right thread in the GPU. Need to allocate memory and threads in sync. Also, have a hiccup when the images don't fit into GPU memeory and we have to transmit copies multiple times.

Q Errors?

A Numerical ones, but the same from CPU as GPU.

Chapter 4

26 September 2013

4.1 Covering Arrays: Torres-Jimenez

Suppose a software component has 10 components — possibly 2^{10} cases and 2^{10} tests. Is this necessary? Depends on what we want to do. Maybe we only need to test that on/off works — two tests. To test two combinations, six tests (JHD didn't follow this). Quoted a US analysis (NIST) which showed, for a variety of applications, 6-strength was all that was needed, and for NASA applications, 1-strength found 60% and 2-strength found 95%.

Orthogonal arrays: $OA_\lambda(N; t, k, v)$ is an $N \times k$ array of integers $0, \ldots, v-1$ such that ... (I think each t-combination arises λ-times. These are related to Latin squares. In there are η Mutually orthogonal Latin Squates, then we can construct $\eta + 2$ ....

See `math.nist.gov/coveringarrays` and [KS73]. Covering arrays: $CA(N; t, k, v)$ is an $N \times k$ array of integers $0, \ldots, v-1$ for any t distinct columns out of the k columns, each combination occurs at least once. It is comnjectured that constructing these are NP-complete (unknown), but several related problems are certainly NP–complete. How many of these:

$$CAN(t, k, n) \approx v^t \log k$$

but the search space is bounded by $v^t \leq S \leq \binom{v^k}{N}$.

Example 1 $CA(5; 2, 4, 2)$*: any two columns contains all values of combinations of the 2 symbols:*

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array}$$

$k = v = 2$ is solved, and some prime-power v also. Note that CAs have row symmetry, column symmetry and also symbol symmetry (the numbers $0, \ldots, v-$

1 are only labels). Various constructions by cyclotomic vectors, group theory etc. NIST used the "generalised fusion operator" and 1.5M CPU hours to find 349 new upper bounds, one fo which was 30% better than previously known (average gain 3.85%).

Covering arrays may have redundant elements.

4.2 Fresh Variable Automata

Web services, variable,s real-time with time as a variable. His motivating example was cart⊗search.

Fresh Variable Automata (FVA) are theleast extension to allow infinite alphabets. (finite words, $|\Sigma| = \infty$). Q is set of states, $\mathcal{X}$ is (finite) set of variables. These are closed under intersection, union, concatenation and kleene operator, but not complement. For intersection, consider $A_1 \otimes A_2$, where transitions are labelled by pairs of states — 2-FVA, which is equivalent to being an FVA.

Theorem 6 *Communication bisimulation for FVAs is decidable.*

4.3 Altergo: Iguernelala

program + Specification fed to a verification tool produces mathematical formulae. These need to be verified. These use mathematical (unbounded) integers, many theories, hundreds of quantified formulae, and sometimes nonlinear arithmetic, which is undecidable. Shows integer square root as an example. In Deductive Program Verification, we only need a small set of rules (algebraic).

Rewrite equalities via the LIA solver (rewriting modulo AC). For inequalities, we use an interval calculus.

Can be used in SPARK 2005/2014 (Ada), Frama C and Caveat (C). For SPARK 2014 benchmark we are 20% faster than prvious versio of AltErgo, and resolves slightly more.

4.4 Bound Propagation in Vampire: Dragan

Inspired by DPLL.

Level 0 No bounds

Level 1 Pick a variable/value and propagate bounds

Level 2... Continue until a contradiction, then backtrack and learn.

Unfortunately, bound propagation is non-terminating. So do limited bound propagation.

Had toadd various types to Vampire, which are useful in general. Represent numbers as three types

- Native — `long double`
- rational
- ??

What value do we choose:

- smallest
- "nicest" — looked like Farey fraction between (bottom,top)
- several others, including "random"

No one strategy solved most problems, and some "bad" strategies did solve problems that nothing else did (ouch!). Looks worse than Z3 or Yikes on the benchmarks quoted.

Q So how do you choose?

A There is no "best" answer.

4.5 Hoare logic in K: Arusoaie

See section 2.8. What should we do in practice to verify K-semantics programs? A langauge L might have its syntax defined in IMP. IMP programs have State:Var↦values. In practcie a Hoare-style verifier is language dependent. As well as Hoare triples, we also need to transform these into reachability rules. Showed example with `while`. End up passing to an SMT solver.

Q The time figures in the paper look confusing.

A There's a significant overhead of loading the K system.

4.6 Certified Univariate Taylor Models in Coq; Martin-Dorel

We want tocompute polynomial approximations to functions with provable bounds. The correctness of such bounds is part of correctness proofs for many numerical codes. Overview of interval arithmetic — dependency property (e.g. $x(1-x)$).

A Taylor model is (P, Δ) (apparently "use P with an error interval Δ). Taylor–Lagrange theorem (with remainder). Roudning errors are easily handled by interval arithmetic. $(P_1, \Delta_1) + (P_2, \Delta_2) = (P_1 + P_2, \Delta)\Delta)$ and similar rules (see paper) for multiplication and composition.

Zumkeller's technique (his thesis) producs Taylor approximants. Use SSreflect and MathComponents libraries, and Coq.Interval as te abstract interface for intervals. Note that this CoqApprox is formaly proved,unlike Sollya, and we are generally faster. Naïve Co (not Zumeller) has much greater approximation errors.

Q–JHD How does this compare with MetiTarski [PPdM12], which I think is doing the sme thing?

A Does MetiTarski do transcendental functions (JHD — yes), then it would be interesting.

-
-

4.7 An extension of the Interpreter pattern

Presentation remotely given. Term rewriting is a Turing-complete high-level system. We wish to use this in an OO-framework. Why not use existing systems? JHD didn't quite follow the answer to this rhetorical question.

"Interprter": given a language, define a representaton forthe grammar and an interpreter for that language. The original Interpreter required a lot of boilerplate. As an example we take set expressions, and use DNf as the normal form. This reuires the usual logical rules, but the classes representig operator smust overrite `simpify`, and this complicates re-use. So we us ethe Boolean algebra view of set theory (at least that's what JHD understood).

Claim that this helps re-use. hence we propose a domain-parametric extension of Interpreter. This produces a generic `normalise` operator, base don factory method `build`. The type `LogicalExpression` collects all that is needed to simplify a logical expression. haven't done a full complexity analysis, but pragmatically the cost of the new approach is $< 10\%$.

-
-

4.8 Paralel averages or noisy Monte Carlo samples

Semi-classical time-continuous functions. We choose, with $p = 1/2$ to move wither coordinates or momenta, of one particle.

```
do I=1,nblock
  do L=1,ntrack
    detect position
    #move system
    for j=1,navg
      take a sample
    compute average
    compute observable
```

Want 2D parallelism (BlueGene Q) so use nblock and navg as the two variables. Needed to modify the RNG. Columns = numbe rof MPI-threads per node. FERMI has 64, for example (16 cores with 4 hyperthreads each). Use `nblock=2048` and `navg=` 512 or 2048 (depending whether its coordinates or momenta). 64 threads gives a 54× speedup for weak scalability. Strong scaling is good — 90% at 131K cores.

4.9 Parallel Implementation of a Region Growing Segmentation Algorithm

Region-Growing is the alternative to edge detection. Two parametrs — maximum distance between merged regions, minimal size fo region at end.

Q What precision?

A float then double.

4.10 Parallel Poisson Surface Reconstruction with a GPU

Want to reconstruct the surface from a set of unorganised points. $\min_X(\nabla X - V)$ appears to be the quantity we need to compute (JHD suspects its the location of the minimum). We discretise with an adaptive octtree. Galerkin formulation transforms to a linear system. Compute the node array bottom-up, from finerto coarser. ut we may not get all thenodes this way. We scan each parent and checki for missing nodes, which CAN be done on a GPU.

4.11 (Nanotechnology)

The experimental guys can produce fine nanostrcutures, so bizarre metamaterials. Photonics is one application. Consider a 2D structure of cylinders (that's what he said — I think we're looking at a 2Dslice through a set of aligned cylinders, and perpendicular to the axis). OPTIMET — program to do *ab inition* computation of fundamental frequency. Sequential code, but embarrassingly parallel. Produced great 2D results, but real engineers want 3D computations. Use Bessel function expansions. They had 2D library code, but not 3D, which was a challenge. Had an addition theorem. Joint project with UCL, and also had had NAG support[1]. Want to scatter of irregular spheriodal particles,not just spheres.

C++. I/O support via HDF5, which is good for ineroperability with other GLS, ScaLAPack. OpenMPI. BleGen at UVT, Legion at UCL at HECToR. For 12 spheres, the scattering matrix ($n \times n$) is 20GB, so must itself be distributed.

[1] JHD: I assume via HECToR CSE scheme.

However, the interesting result is as n-vector, so once computed, can be collected and post-processed sequentially.

Future work involves looking at a hybrid OpenMP/MPI version. Also need to reduce dependence on specifics of vendor libraries and MPIs. Also managing mapping of the matrix onto the nodes needs to be managed.

4.12 Current status and H2020 Perspectives: Petcu

4.12.1 Current situation

`http://host.hpc.uvt.ro`

- Have EU-funded 2012–2014.
- Want to collaborate with EU Stakeholders: CINECA, EPCC, INRIA,PSNC, UEX.
- Have a GPU cluster: 3100 CUDA cores.
- Open-access to the UVT resources — come for two weeks – two months.

4.12.2 H2020

- FET Computer Science: Towards Exascale computing 1–4 (note topic 2:programming environments for extreme parallelism)
- Research Infrastructure challenge 2.1 Development of e-infrastructures.

* Topic 1 looks like "son of PRACE"

4.13

[JHD arrives late] 30 runs of 1000 iterations each. Search spaces of dimension 2, 4 and 10. All of DE, PSe and CMA failed for 10 dimensions, and were disappointing even in four. Loss of population diversity seems ot be a good termination criterion for DE and PSO, but maybe not for CMA. Statistical analysis is a promising tool for looking at the behaviour of evolutionary algorithms.

Bibliography

[ASB09] Marlene Arangu, Miguel A Salido, and Federico Barber. Ac3-op: An arc-consistency algorithm for arithmetic constraints. In *CCIA*, pages 293–300, 2009.

[Bar68] E.H. Bareiss. Sylvester's Identity and Multistep Integer-preserving Gaussian Elimination. *Math. Comp.*, 22:565–578, 1968.

[Dod66] C.L. Dodgson. Condensation of determinants, being a new and brief method for computing their algebraic value. *Proc. Roy. Soc. Ser. A*, 15:150–155, 1866.

[GCL92] K.O. Geddes, S.R. Czapor, and G. Labahn. Algorithms for Computer Algebra. *Kluwer*, 1992.

[HW74] John E Hopcroft and Jin-Kue Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM, 1974.

[KS73] D.J. Kleitman and J. Spencer. Families of k-independent sets. *Discrete Math.*, 6:255–262, 1973.

[LYZ10] Z. Li, Z. Yang, and L. Zhi. Blind Image Deconvolution via Fast Approximate GCD. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 155–162, 2010.

[PPdM12] G.O. Passmore, L.C. Paulson, and L. de Moura. Real Algebraic Strategies for MetiTarski Proofs. In J. Jeuring *et al.*, editor, *Proceedings CICM 2012*, pages 357–369, 2012.

[RS04] P. Rogaway and T. Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Proceedings FSE 2004*, pages 371–388, 2004.

[Sch85] A. Schönhage. Quasi-GCD Computations. *J. Complexity*, 1:118–137, 1985.