Retrospective sketch of Warren D. Smith's research 1990-2001

Warren D. Smith
wds@research.NJ.NEC.COM

date = April 17, 2002

Abstract — **Nonexhaustive description of some highlights of my research. Cites of the form [1] refer to the bibliography at the end. Cites of the form (#34) refer to my list of online-available papers** http://www.neci.nj.nec.com/homepages/wds/works.html[1].

Much of my research 1990-2001, can be grouped into 5 themes: Inscribable graphs, Physics meets computer science, Game theory, Voting theory, Geometrical algorithms and graphs.

Contents

1 Inscribable graphs

Soon after I arrived at NECI, I realized that a very old open geometrical problem, first posed by Jacob Steiner in 1832, ought to be solvable by using results I'd found in the PhD thesis of Igor Rivin.

The inscribability problem is: "which polyhedral types are convexly inscribable in a sphere?[2]"

It had remained unsolved since it was posed by Jacob Steiner in 1832. The related easier problem of characterizing the convex polyhedral types was solved by E.Steinitz (this is described in Grünbaum's book [10]). The answer is

Theorem 1 (Steinitz's theorem) *A graph is the one-skeleton of a convex 3D polyhedron if and only if it is a 3-connected planar graph*[3].

Igor Rivin and I found the first polynomial-time algorithm for deciding the inscribability (or circumscribability – this is where each face of the polyhedron is tangent to the sphere) of any particular polyhedral type. Algorithmically, we can actually *find* an inscription, accurate to D decimal places, in time polynomial in the number of bits in the input and output (if one exists).

Unfortunately the best writeup of the proof resides in a 90-page unpublished manuscript by me, but Rivin has published some papers outlining it. My main contribution was to recognize that the Steiner problem was solvable using Rivin's thesis, correct a major error in his thesis, realize that one could get a polynomial time algorithm as a result of the Ellipsoid algorithm technology in [9], organize the

[1] Many of these are published elsewhere, but you'd have to examine the separate Journal and Conference publication lists to deduce that; see my web page http://www.neci.nj.nec.com/homepages/wds/.

[2] A polyhedron is "inscribed" in a sphere it all its vertices lie on the sphere's surface. It is "convex" if, whenever two points A and B are inside the polyhedron, then so is the point exactly midway between them. A polyhedral "type" is the *combinatorial* specification of which faces connect to which, which vertices are on which faces, etc., as opposed to the *geometrical* specification of actual coordinates for all the vertices.

[3] A "graph" is a set of "vertices" and "edges;" it is "planar" if the vertices may be embedded as points, and the edges as *non-crossing* Jordan curves in the plane joining those points. A graph is "k-connected" if you need to delete at least k of its vertices to disconnect it.

project, and write it up while serving as chief devil's advocate. Craig D. Hodgson (Mathematics Department, University of Melbourne, Parkville, Victoria, Australia) also played a role.

Our characterization is a simple to state linear program, but the proof is difficult and involves hyperbolic (nonEuclidean!) geometry and Brouwer's "invariance of domain" argument from topology – two things one might not expect to have anything to do with it. Here is the theorem[4]:

Theorem 2 *Let P be a polyhedral graph with a real weight $w(e)$ assigned to each edge e. Let P^* be the planar dual of P, where the edge e^* dual to e is assigned the dual weight $w^*(e^*) = \pi - w(e)$. Then P can be realized as a convex polyhedron in $\mathbf{H}^3$ with all vertices on the sphere at infinity and with dihedral angle $w(e)$ at every edge e if and only if the following conditions hold:*

1. *$0 < w^*(e^*) < \pi$ for all edges e.*
2. *The sum of dual weights of edges $e_1^*, e_2^*, \ldots, e_k^*$ bounding a face in P^* is equal to 2π.*
3. *The sum of dual weights of edges $e_1^*, e_2^*, \ldots, e_k^*$ forming a circuit that does not bound a face in P^* is strictly greater than 2π.*

There have been several followup papers on consequences of our characterization (#2). The enumeration of inscribable & circumscribable polyhedral types was discussed in (#7). Four-connected polyhedra are both inscribable and circumscribable (#5) [5]. Other consequences: Polynomial time algorithms for "circle packings" and numerical conformal mapping (#6), a fast algorithm for trivalent inscribability [4], proof of the NP-completeness of finding Hamiltonian paths in Delaunay triangulations [3], and an understanding of the numerical stability properties of the 2D Delaunay triangulation [6]. More recently E.Jucovic has used our work [E.Jucovic, ≥ 1 papers].

2 Physics meets computer science

What do computer science and information theory have to say about the laws of physics? How do the laws of physics impose fundamental limits on computer performance – or allow greater performance than might be expected? These questions have always fascinated me. They have also been a central theme in the careers of Rolf Landauer and Charles Bennett, two physicists at IBM, and more recently of Peter Shor (AT&T labs).

I taught a course on this subject at Princeton University in fall 1998. There is some possibility a book will arise from it; I think 1 or 2 more iterations of teaching that course should provide enough lecture notes and organization to make a book. This course is described in `www.neci.nj.nec.com/homepages/wds/pu-course.html`.

[4]You have to know about the "conformal and projective models" of hyperbolic 3-space inside a Euclidean ball, to understand it – suffice it to say that P is inscribable in Euclidean space if and only if it is realizable as a hyperbolic convex polyhedron with all vertices on the sphere at infinity.

2.1 *Information flux, memory capacity, and the laws of physics*

Consider the following problem. You have 1 kilowatt of power. You want to communicate information to me through a square window 1 centimeter on a side. What is the maximum number of bits per second you can transmit to me?

I claim that the laws of physics impose a limit on this of 3.82×10^{22} bits per second, and in fact there is a sense in which this number is achievable.

The way to get such a bound on information flux, and that the bound is tight, occurred to me in 1992 (#10). However, the argument depended on some nonrigorous "physicist's arguments," or anyhow on making some physically plausible, but mathematically rather annoying, assumptions.

Roughly speaking, it is this[5]. I claim the maximum possible information flux, in bits, corresponds to the $\log_2$ of the number of distinguishable states of a electromagnetic radiation field passing through the window and having total power 1 kilowatt. This is the same as the (known) "entropy flux of blackbody radiation" with that power flux. The assumption that more information flux that this is possible, may be shown to lead to a contradiction with the second law of thermodynamics and the law of conservation of energy: Consider shining your magic beam with power P and allegedly more information flux than a blackbody, into a large cavity forever. Eventually thermal steady state is reached – the cavity will be radiating out the same power you are radiating in. But the cavity (of course) will be emitting blackbody radiation. Therefore, the total process will be decreasing the net entropy of the universe, a contradiction. This argument certainly sounds convincing at first, but it is not a mathematical proof. For one thing it depends on the false assumption that you must use photons. This assumption is asymptotically essentially true, in the limit of high power and high information fluxes, according to some additional "physicist's arguments" which also are not a mathematical proof but are pretty convincing.

To show the bound is tight, you consider shining "blackbody radiation" through the window onto a huge hemisphere coated with spectro-analyzing, polarization analyzing single photon detectors. Shannon-like arguments show that the presence of thermal noise can be corrected for.

My bound is summarized by the following formula ($\dot{I}$ is information flux in bits per second, A is window area, P is available power, and κ_1 is a fundamental physical constant expressible in terms of the usual constants such as $\hbar$.)

$$\frac{\dot{I}}{A} \lesssim \kappa_1 \, (\frac{P}{A})^{3/4}, \quad \kappa_1 \approx 2.149 \times 10^{21} \, \frac{\sec^{5/4}\mathrm{bit}}{\mathrm{Kg}^{3/4}\,\mathrm{meter}^2}. \qquad (1)$$

In 1998, I finally saw how to use operator analysis to make this result, and a similar tight bound I have on the number of bits storable in a memory of given mass and volume, fully

[5]I omit a tremendous amount of discussion on "ways to cheat" and when and if they work. This same nonrigorous argument was also invented by Lachmann, Newman, Moore [12] independently.

rigorous. The bounds presently (#43) follow solely on the assumption of Schrödinger's equation – or certain alternative equations that are caricatures of relativistic quantum mechanics (either will do) – and are valid in the asymptotic limit of high power density (or high mass density in the case of the memory bound).

Incidentally, R.Landauer has published several papers indicating communication is possible [11] while consuming *zero* power. The contradiction between his and my claims is because of a difference in our definitions of the word "communicate." (Actually, Landauer does not provide a definition at all, but this is the key issue.) For example, suppose we are sitting in the vacuum of space, at opposite ends of a long rigid rod. A tennis ball orbits us both, sometimes passing near me, sometimes passing near you. Of course, the molecules inside the tennis ball encode information. Maybe it is not interesting information, but that does not matter. Now we are each expending zero power, but the ball keeps orbiting forever "transmitting" information from my vicinity to yours.

Unfortunately, it is the same information over and over, and you never actually measure all the atoms in the tennis ball so you in some sense never receive the information.

I claim that Landauer's zero power scheme, when reduced to its bare essence, is precisely the tennis ball scenario above. If conditions are imposed that the information stream has to continue forever and keep being "genuinely new" information (e.g. if I transmit independent random bits) and you genuinely have to "receive" it, then Landauer's construction won't work and my bound becomes valid.

Another remarkable consequence of my information flux bound, was the impossibility, *on information theoretic grounds*, of building a "cloak of invisibility" of the sort found in various works of science fiction (#26).

More recently I have been able to solve one of the main open questions I'd been unsatisfied with (and had in fact treated incorrectly) in #10: the attempt to get a law of "information destruction", i.e. saying that stored information must decay at at least some rate, or ("quantitative version of 2nd law of thermodynamics") saying that entropy must increase at at least some rate. The resolution of this puzzle (which I am going to write up in a revision of my old paper...) is as follows. I had had a big list of cross section formulae for various interactions between various kinds of particles. However, they were the wrong formulae in the sense that what is wanted is the "asymptotically dominant" formulae representing the most important interaction-type asymptotically in the "extreme relativistic limit." For example, "double pair production" is asymptotically more dominant than single pair production for photon-photon interaction. The dominant interactions are (at least conjecturally) known, as I recently found out, from work which was done in the late 1970s but largely forgotten. The resulting cross sections are *independent* of the energy of the particles in the high-energy limit (or get unboundedly large, in which case, we should avoid those kinds of particles if we want to prevent interactions). In that case, it *does not matter* how you store information – thermal statistics are the best in the sense you store the most information with the least mass-energy, but all statistics are the same as far as decay rate is concerned, provided you are in the high-info-density asymptotic limit. Then the rate of decay of that information (i.e. #particle interactions per second) will be bounded below by an explicit formula. Thus I *can* get a pretty plausible law of information-decay, or quantitative version of the second law, *provided* we are in the high info-density limit, and the resulting bound is in fact tight in the sense it is actually met by a "photon-gas" memory. Furthermore, the law seems to have some predictive power in the sense that, the reason white dwarf and neutron star matter are the density they are, is, that they could not store the same info stably if they were denser, roughly speaking. (These are both relativistic Fermi gases and come reasonably close to my bound.) Details will appear in the revised #10.

Two other reports I wrote relevant to the subject of physical limitations on the performance of computers are #44 (how much time is required to perform elementary operations?) and #46 (concerning "reversible computation" consuming zero power).

2.2 Church's thesis I: classical mechanics

One of the foundation stones of computer science (and of physics, although many physicists are blissfully unaware of it) is "Church's thesis" which states that any physical system is simulable by a Turing machine, and conversely, no physical system can compute something a Turing machine cannot.

Church's "*strong* thesis" states that furthermore, this simulation may be performed with only "polynomial slowdown."

This is a "thesis" rather than a theorem since it depends on some rather undefined concepts ("compute"? "simulate"?) depends on (yet unknown) laws of physics, and is unproven.

Unfortunately, it looks as though Church's strong thesis is likely to be *false* in linear quantum mechanics, since Shor [15] showed how to build a plausible quantum system which would factor N-digit integers in polynomial(N) expected time.

In (#21) I point out that Church's thesis does not need to be a thesis. With appropriate definitions and selection of some set S of laws of physics, it becomes a mathematical question subject to proof or disproof for that S.

I was able to *disprove* Church's thesis in classical mechanics by showing that with Newton's laws of gravity and motion, N point masses in the Euclidean plane could describe uncountably many topologically distinct trajectories in 1 minute, whereas, in any finite time, a Turing machine could only do a finite number of different things. This demonstration leaned heavily on work of J.Gerver [8].

On the other hand, I was able to *prove* Church's thesis in a slightly more realistic set of physical laws (Newton modified in two ways to make it more closely approximate the demands of special and general relativity). This was because of a powerful new result on simulating the time evolution of a wide class of differential equations with only "polynomial slowdown."

2.3 Church's thesis II: quantum mechanics

In 1998, I studied the same problem – N point masses, interacting via inverse-square (or a wide class of other) force laws – but now employing Schrödinger's equation of quantum mechanics rather than Newton's laws of classical mechanics[6]. The grand conclusion (#49) is that, in these laws of physics, Church's thesis is true; quantum mechanics is algorithmically simulable, to arbitrary accuracy. Indeed, if the simulator is a "quantum Turing machine" rather than a conventional one, then the simulation has "polynomial slowdown" – i.e. the *strong* thesis is true.

I view Church's thesis, in this paper, as a statement about a competition between a simulator (a Turing machine with access to a random bit generator) and a genuine physical experiment. You run both the simulator and the experiment, and then you measure (or ask the simulator to simulate measuring) some quantity. For example, you could measure the boolean quantity "is there a proton in this box (yes/no)?" Both the simulator and the experiment return some answer, which is a random variable (in this case, an appropriately biased coin flip). If the two coin-bias probabilities p_{simul} and p_{expt} differ, then it would be possible, after making a large enough number R of re-runs of both the experiment and simulation, to become highly confident that $p_{\text{simul}} \neq p_{\text{expt}}$, i.e. that the simulation was wrong. However, *my* simulation algorithm has the property that it *tries harder* on the $(k+1)$th re-run than on the kth (k is an input to the algorithm), i.e. striving for *more* accuracy. I prove that it does this in such a way that for *any* measurement, no matter how many re-runs are made, one can *never* acquire high statistical confidence that the simulation is wrong.

I also mention that, if the point-particles are regarded as having *magnetic moments*, then I pointed out also in #49 that quantum mechanics *breaks down*, goes singular, and has no solution. This easy fact has not previously been pointed out. It may have a healthful mental shaking-up effect on any smug feelings you may have that quantum mechanics takes care of restoring algorithmicity, so physics is ok again. Presumably this indicates the essential need for *relativistic* quantum mechanics to explain the stability of matter.

2.4 Church's thesis III: hydrodynamics

In #58, I showed that the usual Navier-Stokes equations of "hydrodynamics" are nonalgorithmic – or, certain widely believed assumptions about hydrodynamics are wrong. The proof is the most dramatic possible: I show how to build a fluid mechanical machine that solves undecidable problems, such as Turing's halting problem, in 1 minute. (The 1-bit output can be of the form: either > 2 liters of fluid flows into a basin, or < 1 liter.) It looks as though this machine still will work even if the initial conditions are slightly randomly perturbed in certain ways; i.e. its workings are "self correcting."

I consider this to be a very dramatic conclusion signifying the "failure of hydrodynamics." Let me make an analogy. Great effort has been devoted, and will continue to be devoted, to the task of trying to decide whether computer programs are correct. (Also: for the task of simulating fluids.) But these efforts have never been very successful. It is true *some* programs have been proven correct, and *some* fluid scenarios have been proved to have a unique known solution, but these are a tiny minority. Nobody has ever managed to make an algorithm to decide program-correctness. Nobody has ever managed to make an algorithm to simulate hydrodynamics to arbitrary (user-specified, guaranteed) accuracy. There is a reason for that: these feats are both *impossible*.

2.5 Church's thesis IV: general relativity and QED?

The question of whether general relativity is simulable seems very difficult, since existence and uniqueness is still open for GR, and it is known that, for example, gravitational waves in a vacuum can self-focus to create black holes out of nothing.

Quantum field theories such as QED are also difficult to address, again for reasons of lack of rigorous mathematical foundations, but also because these theories, at least as usually employed, do *not* claim to predict *any* physical quantity with arbitrary accuracy.

But I believe that in future I will be able to make progress extending the algorithmicity program into both these areas.

2.6 Physics meets topology? Monopoles; 3-dimensionality

Why do all electrons have the same electric charge? Why do all protons have the same (exactly opposite) charge? In 1931 and 1948, the great physicist P.A.M.Dirac proposed a remarkable possible answer[7]. Namely, he showed that the existence of even just one "magnetic monopole" somewhere in the universe would make the quantization of charge logically inescapable. Dirac also related the quanta of magnetic and electric charge; each determines the other. Various "grand unified theories" have since been proposed that feature, and predict the existence of, monopoles. Monopoles would make Maxwell's equations more symmetrical and hence more "beautiful." But: despite numerous searches, no monopoles have ever been detected.

In #55 I point out that there is another, perhaps even more remarkable, possible explanation for charge quantization. It is this. Suppose the universe is a compact boundaryless 3-manifold with some nontrivial topology[8]. In that

[6] This is probably the most difficult paper I've written. It is both conceptually and technically difficult; it employs rigorous quantum mechanics, a difficult subject; and it steals key ideas from some great mathematicians/physicists (R.P.Feynman, E.H.Lieb, T.Kato, M.Suzuki) as well as introducing several new ones.

[7] Which goes rather beyond "because they just do."

[8] "Compact" means finite. It is easier to imagine a 2-manifold, i.e. 2-dimensional, universe. For example, imagine the surface of a

case there could be (and presumably would be) some "topologically trapped magnetic field" in the universe[9]. In that case (and my proof assumes, for simplicity, that universe has a *fixed* shape) it turns out that the quantization of electric charge also is logically inescapable, for *every* compact boundaryless orientable 3-topology – *except*[10] for the surface of a 4D ball, for which my argument apparently does not work. In this, the entire universe plays the role of Dirac's monopole. The quantum of charge turns out to be a "topological invariant," unaffected by distortions of either the exact geometry of the universe, or its exact magnetic field. The charge quantum is expressible in terms of the size of the universe and the strength of its magnetic field; a magnetic field far far below present detection limits suffices to explain the actual value of the electron's charge.

This all allows us to abolish magnetic monopoles and hence perhaps to simplify future "grand unified theories" considerably. I also show in #55 that monopoles would (if they existed) have some very peculiar and nasty properties, which makes it even less likely that they exist.

Another fundamental question: why is the universe 3-dimensional? #55 surveys various proposed "explanations" of this – and introduces a new, topological one. It relies on the following topological characterization of the number "3." Consider a d-dimensional universe. Consider combing d different colors of hairs in that universe, so that, at each point of the universe, the hairs of different colors are orthogonally oriented[11]. If that is possible, the universe has a coordinate system which "locally looks like orthogonal Cartesian coordinates in flat d-space" everywhere.

Theorem 3 **(d-combability)** *For $d \geq 2$: all orientable d-manifolds are d-combable if and only if $d = 3$. (If $d = 2$, the only orientable 2-combable manifold is the torus and the only nonorientable one is the Klein bottle. If $d \geq 4$ all d-manifolds are generically not d-combable, i.e. even if one is d-combable, a slight distortion will suffice to make it not.)*

2.7 DNA computers

L.Adleman started a small industry in a paper [1] demonstrating the solution of a small computational problem (finding a Hamiltonian path[12] among 6 "cities" with a given set of "roads" interconnecting them) by DNA molecules in a test tube. Actually, in my opinion there remains room for doubt that Adleman's DNA actually did solve the 6-city problem he thought it solved[13]. But anyway, there now can be no doubt that methods like Adleman's can solve computational problems using molecules. The specification of which problem to solve, and the sequence of steps to solve it, are controlled by a chemist by performing sequences of chemical and physical operations, and then the molecules in the test tube solve the problem.

In (#29) (with Al Schweitzer[14]), we demonstrated[15] that well known chemical and physical operations on test tubes containing DNA could be used to make an ultra-parallel *Turing universal computer*[16]. By contrast, Adleman's methods were not universal and were specially designed to solve Hamiltonian path problems.

This was the first demonstration that realistic chemical techniques were capable of universal computation, and in my opinion, it is still the most realistic available proposal, with the most analysis.

But our analysis of engineering and actual numbers, suggests that neither our scheme, nor anybody else's schemes so far, will be competitive with conventional computers in the forseeable future.

Although there is enormous parallelism inherent in the fact that you can get $> 10^{13}$ molecules of DNA in a test tube[17], a serious problem is that chemical yields (which are below 100%) multiply exponentially, and undesired side reactions introduce similar exponential effects. We have analysed known yield numbers and side reaction rates, and no scheme proposed so far can run for very many steps nor seems capable of rivaling conventional computers[18].

doughnut ("toroidal topology") or of a two-holed doughnut (a different topology). Or: imagine the surface of a ball. All of these are compact 2D topologies. There are a great number of possible 3D topologies our universe could be, and at present it is totally unknown which.

[9]For example, imagine "combing the hairs" (symbolizing the magnetic field) on a doughnut. Such a (constant) magnetic field would be a stationary solution of Maxwell's equations, and never would go away. But it is topologically impossible to comb the hairs on a ball or 2-holed torus. Those mental examples were 2D. I proved that in *all* compact orientable *3D* topologies, it *is* possible to "comb the hairs" and find a nonzero magnetic field in that universe which is a stationary solution of Maxwell's equations and can never go away.

[10]Strictly speaking: I needed to assume the widely believed "Poincare conjecture." Incidentally, many of the topological theorems in #55 are really due to great topologists, not me – I reformulated and combined their theorems in various, comparatively trivial, ways to get my theorems.

[11]For example, if $d = 2$: it is easy to see how to comb red hairs on a torus going "around it" while simultaneously combing blue hairs going "around it in the perpendicular direction."

[12]A "Hamiltonian" path is one that visits every city exactly once and follows the roads.

[13]I would have been more convinced if he had sequenced his final DNA to actually "read the output" and hence tell what the path was, instead of deducing indirectly that the solution must have been found.

[14]Schweitzer asked to be removed as an "author" from the final published version. That version was also published with the key figures omitted and printed doublespaced, as the result of editor errors. Both of these developments were unfortunate, but are fixed in my online version of the paper.

[15]Our demonstrations were on paper only and were never performed in a lab (although all the individual steps we need, have been performed many times in labs).

[16]Essentially, a computational device is "Turing universal" if it may be programmed to solve *any* problem solvable by any other kind of computational device.

[17]Our computer may be programmed to make nondeterministic "two-way branch" steps at selected moments, allowing access to that parallelism; indeed we can solve NP-complete problems in polynomial time – up to the point where you run out of DNA! Speed numbers: Each chemical cycle to perform a Turing machine step takes maybe 1 hour in the lab. This is either pathetically slow (1 step per hour) or extremely fast ($> 10^{13}$ steps per hour) depending on how much you like parallelism. My desktop computer is a 90MHz machine and hence performs 90 million steps per sec $= 3.24 \times 10^{10}$ steps per hour. (Each such step is really equivalent to quite a few Turing steps though, since my computer has a more powerful set of basic instructions than a Turing machine. For example, the "add" instruction on my computer might require 100 Turing machine steps to emulate it.). Hence, the testtube in some sense is comparably fast, and for some tasks even faster, than my desktop computer.

[18]Although we showed how to incorporate error correcting codes

Ultimately, future biomolecular technology might reach the point where chemists could achieve similar reliability and self-correction capabilities to those enjoyed by life itself, indeed to the point where one could easily make full-custom lifeforms. At that point DNA computers probably would be practical, but that might then pale in significance! For more on this, see (#30).

It therefore seemed to Schweitzer and me that, *since molecular biology does not seem to make any great practical contribution to computers* (although they are connected in a theoretical sense) *it is a better idea to go in the other direction and see what computer science can say about biology.* Our paper's second half contains a survey of biological processes that look highly computational, together with some speculation.

Two of those speculations are as follows. First, we have a likely explanation for the origin and nature of the "junk DNA" that constitutes 98% of human chromosomes. Second, "RNA editing" processes that happen in trypanosomes might provide the basis for a future kind of DNA (actually RNA) computer possibly ≈ 1000 times faster and more reliable than all the designs so far. In my opinion, this idea is the most promising direction proposed so for "DNA computer engineering." In fact, I think it is the only idea that has been advanced so far with any chance of competing with conventional universal computers.

2.8 Chemistry

Suppose you would like to build a macromolecule shaped like the statue of liberty – or any other shape you care to specify. I have in mind (for example) the problem of manufacturing parts for future "nanomachines" or "nanocomputers."

In (#34) I sketch out all the essential ideas (mathematical, algorithmic, and chemical) you need to build a robot capable of synthesizing macromolecules of any desired shape. You tell it the shape; it finds the synthesis to stick elementary "molecular tinkertoy" molecules together to build it, it moves its robot arms to pour the right test tubes, turn on the right heaters, etc... and at the end of the day, it hands you a flask of the stuff you wanted.

1. I defined mathematical notions of "universal assembling power" for sets of assembling primitive operations that stick elementary components together in various ways. It is essential not to consider all chemistry and all molecules[19], but instead to restrict attention to a specific set of reactions applied to a specific set of molecules which are subsets of a latticework made out of carefully designed molecular components I call "tinkertoys."

2. I discussed "optimal" syntheses. I can prove both asymptotic optimality statements for my class of syntheses, and provide polynomial time algorithms capable of finding the exact optimal sequence of synthetic steps, e.g. producing the most product with the least cost and elapsed time, given a yield model for the individual steps.

3. I presented actual preliminary molecular designs and chemical reactions to implement the (small number of) necessary abstract primitive operations using known organic chemistry. These designs are certainly not intended to be the last word, but instead are merely there to provide enough details, and to get close enough to feasibility, to make it clear to chemists that the scheme I am proposing is no joke.

4. I described the basic techniques needed to analyse geometric rigidity and discussed the likely major limiting factors on the whole plan.

5. I suggest several experimental "sanity tests" of the whole plan, which could be done much more cheaply and simply than embarking immediately on the full scale project.

Of course, my paper is only the beginning of what would need to be a large research program to develop such a machine and the requisite chemistry[20]. However, the technological implications of having such machines available in the future are so vast, that in my opinion it is essential to investigate this further.

3 Game theory

3.1 The usual approach

Charles Babbage and Ada Lovelace in their correspondence ($\approx$ 1840) had considered making a computer play chess. Claude Shannon in 1950 [14] considered the problem further. By the 1970s [16] essentially all the ideas were known that were eventually needed for IBM's "Deep Blue" to defeat Gary Kasparov. These ideas essentially consisted of the following.

1. The search in chess is a tree whose vertices are chess positions and whose edges are moves.

2. The value of a tree leaf is assigned by a heuristic (or occasionally exact) "evaluation function," providing a numerical estimate of "how good" the position is for the player on move.

3. The value of a non-leaf is the the max of the negated values of its children[21]. This is called the "negamax rule."

4. Perform a brute force search generating the full tree down to some depth, assign negamax values to all the

into our scheme, they can only correct some kinds of errors, and not other kinds – such as DNA hydrolysis causing your entire computer to dissolve gradually and get flushed down the drain!

[19] Questions of synthesizability would then be undecidable (#35).

[20] It took many years of work by many chemists to reach the point where today, you can buy DNA synthesis machines to synthesize essentially any 100-long DNA sequence. But the machine I am proposing would synthesize a much larger class of molecules.

[21] Because a good player will select the move which minimizes the value, for the opponent [i.e. negated] of the successor position.

nodes, and pick the move (child of the root) with the smallest value (for your opponent).

5. The approach above may be speeded up with "alpha beta pruning" (with "move ordering heuristics" such as maximizing capture value and the "history heuristic [13];" pruning enables examining exponentially fewer tree nodes while provably getting the same final result) and "lazy evaluations" (which for some leaves use imprecise evaluation numbers, but provably these techniques will not affect the final result of the search) "transposition tables," and "iterative deepening," (these techniques usually don't affect the final result of the search and speed things up) while also performing heuristic "search extensions" and "retractions" intended to search more interesting and more important moves more deeply than usual, and "quiescence search" intended to refuse to evaluate especially "noisy" leaf positions.

6. The "null move heuristic," which is predicated on the assumption that doing nothing is never better than the best move, can enable additional pruning and can speed up your chess program tremendously, but can cause blunders in "zugzwang" positions in which this assumption is false.

3.2 *What is wrong with the usual approach*

Although these techniques (combined with tremendous hardware power) are very effective in chess, mathematically they are subject to criticism.

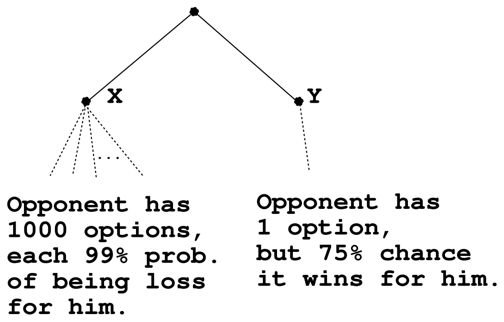

Figure 1: Should you make move X or move Y?

For example (fig. 1): suppose you can choose between two moves X and Y. X gives your opponent 1000 options, each of which has (according to your evaluation function) 99% chance of being a loss for him. Y gives your opponent only 1 option, but it is a 75% chance of being a win for him. According to the negamax rule, our best move is X. But if all the probabilities are statistically independent (?), then the probability that X is a win for your opponent is $.99^{1000} \approx .000043$, so Y is the best move!

But this calculation was assuming that, next move, your opponent will be able to figure out one of the 10 move choices he has, among his 1000 options, that wins for him. Maybe he will be able to figure this out, because next move, he will be able to search deeper than you can now search, and hence will have access to more information than you do. Or maybe your opponent is omniscient. In either of these cases, your best move is Y. On the other hand, maybe he is not omniscient and won't gain enough extra information to gain 75% confidence in the winningness of any particular one of his 1000 options. (Right now, you certainly have no idea which are the 10 wins.) In that case the best move would be X.

This example shows that in chess play between players one or both of whom have *finite* computational resources, the best move is *not* necessarily the one given by the negamax rule. Indeed, the best move choice can depend very much on some rather subtle estimates such as the estimate of how much extra knowledge your opponent will be able to gain in his deeper search next move – as compared to the knowledge available right now in the shallower search this move.

Also, in this example, different assumptions about the dependencies among those 1001 probabilities can lead to drastically different numbers.

3.3 *New BPIP approach*

So: what we really want is some procedure in which we explicitly acknowledge the fact, that at the leaves of our search tree we only have access to *probabilistic* information[22]. Given such information, the question is: what is the best (in a statistical sense) procedure for using that information to choose the best move?

I invented a procedure I called BPIP (Best Play for Imperfect Players) in which the evaluation function returns, not a number, but rather a *probability distribution* indicating the probabilities as a function of Δ that, on deeper search of that position, our opinion of its value (win probability in further BPIP play) will change by Δ.

Remarkably, in this model, it turns out to be possible to compute the exact Bayesian optimal move, and the algorithm that does it is, in practice, not greatly slower than the usual negamaxing procedure. Notice also that the independence assumption BPIP makes is that the *errors in our estimates* are independent. It does not make the assumption that the *estimates themselves* are independent, which would be a much stronger assumption, subject to drastic numerical inaccuracy.

It seems clear a priori that (at least if runtime is ignored) BPIP must be a better statistical move choice procedure than either negamax or win-probability multiplication. One reason is that negamax arises as a degenerate special case of BPIP under the assumption that there is zero probability of any opinion changes; and win-probability multiplication arises in the opposite limit (by assuming the opinion will always change as far as possible – to certainty). It should be obvious that the general case has to be at least as good as these (rather ludicrous) special cases. To verify this, I, together with Rico Tudor and Charlie Garrett, performed extensive computational experiments on game trees from 5

[22] If exact information is available – i.e. we can see to the end of the game – then nobody disputes that negamaxing finds the best move.

different games (Pearl's "P-game," a "connect-4" game, a "kalah" game, "warri," and "Othello"). All tournaments in all games at all depths between negamax, BPIP and win probability multiplication (all players using the same game trees and same evaluator) were won by BPIP[23].

More important for practical purposes is the question of whether BPIP can be made to win in realistic tournament conditions in which the players have bounded thinking time.

Remarkably, it turns out that one can use the extra probabilistic information available in BPIP search to compute, for example, the *exact* Bayesian utility (in our model of opinion changes induced by deeper search) of investigating any particular leaf – or all the leaves – more deeply! This "utility" is essentially the probability that this investigation will change our top level move choice, times how serious a difference (in game win probability) that move choice change will cause.

Unfortunately I do not know an efficient way to compute the utility of investigating an *arbitrary* subset of the leaves more deeply, and this, for large subsets, turns out to be the key issue.

Here Eric Baum joined the investigation. Baum, with some help from C.W.Gear, was able to prove an approximation theorem showing that a particular utility estimator we call "ESS" – which we could (by a remarkable algorithm) evaluate efficiently – would always approximate another utility estimator valid for large leaf sets... and to cut a long story short[24], we ended up with the ability to compute, efficiently, "utility estimates" for the usefulness of investigating any particular leaf in the tree more deeply.

3.4 BPIP game-playing algorithm and its theoretical and experimental properties

So the resulting chess playing algorithm is roughly as follows. Search a tree. Compute utility estimates for all its leaves and find the BPIP best move. If the utility of expanding all the leaves is small and we've already consumed too much thinking time[25] then stop searching and move. Otherwise, adjoin the children of the top 10% (or some constant fraction) of the leaves (with the largest ESS) to the search tree and continue on.

This type of utility-guided search will automatically grow the tree in the most important directions, as opposed to the naive approach of simply searching every line to depth 6. It automatically induces effects like "alpha beta pruning," which are known to save tremendous amounts of time, but now at a whole new level of statistical sophistication. Previous workers had no choice in the matter since their searches simply did not have access to the probabilistic information and utility estimation techniques available to BPIP. Provably (under realistic assumptions), this algorithm will run in time of the same order as conventional (iteratively deepened alpha beta) searches.

This kind of algorithm – BPIP utility-guided search – seemed to us to have the potential of being a superior game player *in practice*, than the conventional approach of iteratively deepened alphabeta with strong move ordering heuristics ("AB").

Again the programming team (Garrett, Smith, Tudor) went to work to perform experiments. In tournaments between utility-guided BPIP versus AB (both sides using the same move generator and the same evaluator – built largely by automated machine learning techniques)[26] in some games AB had the advantage, in others BPIP. Roughly speaking:

- In games in which only rather bad heuristic evaluators are available, or in which exhaustive searches are often feasible, AB has the advantage because it is faster.
- In games in which good heuristic evaluators are possible but exhaustive searches are difficult, BPIP has an advantage versus AB because it is smarter. This advantage grows if more thinking time is available to both players.

This was the first and only time that a non-negamaxing game algorithm had been shown superior to alphabeta in realistic tournament conditions. The superiority can be considerable. In time odds tournaments in Othello, our BPIP player, consuming on average 36.5 seconds of thinking time per game, could play even with AB consuming ≈ 5 times more thinking time. With the BPIP player consuming 250 seconds of thinking time per game (remember, BPIP's advantage empirically grows when both players have more time to think) it appeared to play even with AB consuming ≈ 95 times more thinking time! For contrast: in US Othello association rules for human tournament play, each player is granted 1800 seconds of thinking time per game.

3.5 Other game papers by me

In #8, I analysed a new special function arising in the analysis of the performance of alpha beta with random move ordering.

In #20, I analysed schemes for assigning numerical ratings to gameplayers in a league. Such systems are employed by the US Chess Federation, the Association of Tennis Professionals, US table tennis association, etc. This analysis is, in my opinion, the best available, and the only one discussing (1) rating noise and how to optimize the system to minimize it, (2) rating "inflation," and (3) the distribution of ratings in the population and its relation to models of "learning."

In #25 (with Kevin Lang), we employed a computerized game solver to design optimal combinatorial algorithms, which I believe is the first time this was done. Our computer

[23] Except for depth-2 othello, where negamax had a statistically insignificant edge.

[24] Actually it is not totally clear ESS is the best utility estimator to use, but it is pretty good.

[25] I was able to work out an optimal utility/time tradeoff in a certain simplified model. The parameters in this tradeoff can be adjusted semi-empirically. The point is that we have a natural, non ad-hoc, condition for termination of search.

[26] Our programs were also validated in tournament play versus other computer and strong human opponents. For example, our Othello and Warri players (both BPIP and AB) appeared to exceed Human world championship level.

found new merging algorithms requiring fewer comparisons, as well as new bounds for such algorithms.

4 Voting theory and simulation

In #56, I consider the following very simple voting scheme, which I call "range voting." In a c-candidate election, you select a vector of c real numbers, each of absolute value ≤ 1, as your vote. E.g. you could vote $(+1, -1, +.3, -.9, +1)$ in a 5-candidate election. The vote-vectors are summed to get a vector $\vec{x}$ and the winner is the i such that x_i is maximum.

I prove theorems showing that this is the unique voting system satisfying certain axioms[27]. But, more importantly, I devise a way to compare voting systems experimentally, via computer simulation of elections. There is a mathematical value, called "Bayesian regret," associated with any voting system, quantifying how much society will suffer if that voting system is used in elections. I carried out a computer study of range voting and about 30 other voting systems proposed by other people. This was by far the largest such study ever, it compared the largest number of voting systems ever, and it incorporated a vast number of possible settings[28]. The conclusion of this study was remarkably clear and unmuddy. In *every* experiment tried, range voting had the minimum Bayesian regret among *all* voting systems tried. This is a breakthrough; no previous study has ever been able to show a clear advantage for any one voting system over all others. That is probably because no previous study has incorporated range voting as a contender. I strongly recommend that democratic governments everywhere adopt range voting. The societal gains that will result can be estimated numerically (in terms of units such as "dollars" or "lives") and will be tremendous.

In #59 I devise an interesting "universal counterexample" that simultaneously makes every voting system so far proposed, elect the "wrong" winner. Many voting paradox scenarios have been devised before, but this is the first universal paradox. It relies on my invention of a very interesting nonintuitive object, which might be called "supremely nontransitive dice."

It was discovered, apparently by Trybula in 1965, that "nontransitive dice" exist. For example, suppose you have 4 dice which are *R*ed, *B*lue, *G*reen, and *W*hite. Label their 6 faces as follows[29]

$$R: 004444, \quad B: 333333, \quad G: 222266, \quad W: 111555 \quad (2)$$

Then, amazingly, with probability 2/3 each die "beats" (will produce a larger roll than) the die following it in cyclic order: $R \succ B \succ G \succ W \succ R$.

My invention is 3 dice with an even more surprising intransitivity property – concerning what happens when all *three* of them are rolled simultaneously (as opposed to just two).

5 Geometrical graphs and algorithms

5.1 Steiner ratio – the point sets with "the most shortcuts"

The Steiner ratio conjecture[30] of Gilbert and Pollak (1967) stated that the "minimum spanning tree" of N points in d-dimensional space could be no more than a factor of ρ_d longer than their "Steiner minimal tree" where ρ_d (the "d-dimensional Steiner ratio") was conjecturally achieved when $N = d + 1$ and the points were the vertices of a regular d-simplex[31].

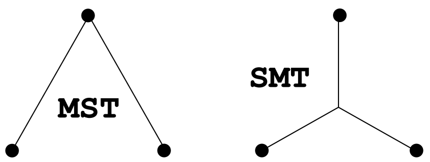

Figure 2: Minimum Spanning Tree and Steiner Minimal Tree of 3 sites in plane forming an equilateral triangle. The MST is a factor of $2/\sqrt{3}$ longer, which, by a theorem of Du and Hwang, is the biggest possible ratio.

In a breakthrough, this conjecture was proven by Du and Hwang [7] when $d = 2$.

In (#12) I gave two proofs it was false for each $d \geq 3$, thus settling the conjecture. D.Z.Du also independently found a disproof, and we ultimately published a joint paper titled "3 disproofs...".

The question then became: what is the right conjecture? In (#13 – with J.M.Smith – no relation) we proposed a new point set called the "d-sausage," which – extensive computer work suggests – is the real ρ-maximizing point set when $d = 3$. It leads to the new conjecture that

$$\rho_3 = \sqrt{\frac{283}{700} - \frac{3\sqrt{21}}{700} + \frac{9\sqrt{11 - \sqrt{21}}\sqrt{2}}{140}} \approx 0.784190373377122. \quad (3)$$

The d-sausage's conjectured Steiner ratio ρ_d is the current record holder in every dimension d, although I suspect it is going to stop being optimal in sufficiently high dimensions.

5.2 Simplexity

In another piece of geometrical investigation (#41), I found the best known lower bound on the "simplexity of the d-cube." Obviously, a square may be divided into 2 triangles. Less obviously, a cube may be divided into 5 tetrahedra. These are both minimum possible. What about in

[27] Called the COAF axioms for "*C*ompact," "*A*dditive," and "*F*air."

[28] E.g.: "honest" voters, "strategic" [but perhaps dishonest] voters, controllable "voter ignorance," different numbers of voters and/or candidates, and different utility generators for what voters think about the candidates.

[29] Or any real numbers having the same order-relations as 0,1,2,3,4,5,6 may be used instead; I recommend mysterious-sounding values such as π, e, $\sqrt{40}$ if you plan to build dice for use in bar bets.

[30] The Steiner minimal tree is the shortest network of wires that will electrically interconnect all the sites. The minimum spanning tree is the shortest network in which only site-to-site line segments are permitted.

[31] I.e., when all pairs of points were at equal distances. When $d = 2$ this means the $N = 3$ points, whose MST is the most longer than its SMT, form an equilateral triangle.

4 dimensions? The answer is almost certainly 16, but nobody has proven it! My bound proves it is either 15 or 16. What about in d dimensions – what is the minimum number of d simplices you need, to triangulate the d-cube? My bound is based on a very simple idea: use hyperbolic (nonEuclidean) geometry – you just compute the volume of a hyperbolic d-cube and divide it by the maximum possible volume of a hyperbolic d-simplex. This (I proved) yields a lower bound stronger, by an exponentially growing factor, than all known Euclidean geometry methods.

5.3 Low loss gears

Another piece of pure geometry was my derivation (#23) of optimal shapes for gear teeth – "optimal" meaning minimizing frictional losses (with the usual linear model of friction force).

The usual kind of gears ("involute gears") have the properties that they minimize vibration in the low friction limit, and still are perfectly meshed even if the gears are slightly mis-spaced. Also, involute gears can be manufactured in interchangeable ways, so that, e.g. you can mesh involute gears A and B, or instead mesh A and C, or B and C, where A, B and C have differing radii, and in all cases you still get a perfect match. Although my minimal friction gears do not have these properties, in the modern age of computer controlled precision machining, that may not be a fatal disadvantage. In applications such as generators, automobile gearboxes, and clocks, where efficiency matters a lot, it may be worth the cost of machining matched sets of (non-interchangeable) gears and spacing them precisely.

5.4 Traveling salesmen, separators, banyans

In (#38), with Satish Rao, we improved groundbreaking ideas of Sanjeev Arora [2] to their ultimate limit, by showing that in any fixed dimension, with any fixed $\epsilon > 0$, a traveling salesman tour (TST)[32] among N point sites could be found in $O(N \log N)$ steps, which was provably at most a factor $1 + \epsilon$ longer than the optimal TST. This runtime is best possible in "algebraic decision tree models of computation." Our techniques also work for finding $(1 + \epsilon)$-optimal geometrical graphs of other kinds, e.g. Steiner trees.

Another contribution in (#38) is the invention of "banyans."

A "t-spanner" is a graph on N sites such that the distance for "cars" driving on the "roads" of the spanner is no more than a factor t longer than the "airplane" distance. In other words, a spanner is a *good* set of roads. The "t-banyan" generalizes the t-spanner notion. Specifically, a t-banyan is a graph (set of line segments) that t-approximates interconnection costs (Steiner minimal tree lengths) for site subsets of *any* cardinality, not just 2.

Theorem 4 (Banyan theorem) *For any fixed $d \geq 2$ and $t > 1$, we show a t-banyan of N sites in d-space exists, has $O(N)$ vertices and $O(N)$ edges, is only a constant factor longer than the Minimum spanning tree, and is computable in $O(N \log N)$ steps.*

In (#39, #40) with Nicholas C. Wormald, I developed a variety of best-possible "geometric separator theorems." The simplest example is

Theorem 5 (Squares separator theorem) *Given: N disjoint squares in the plane oriented parallel to the coordinate axes. Then: there exists a rectangle (also oriented parallel to the coordinate axes) R such that $\leq 2N/3$ squares are outside R; $\leq 2N/3$ squares are inside R; and $\leq (4+o(1))\sqrt{N}$ squares are partly inside and partly outside.*

These theorems yield a vast number of algorithmic consequences. One such consequence is the algorithm with the best known theoretical runtime bound ($N^{O(N^{1-1/d})}$) for finding the optimum TST among N point sites in d-dimensional space, for any fixed d. The techniques also apply to many other kinds of optimal geometrical graphs.

This closes the circle: the best known algorithms for *approximately* optimal TSTs in fixed dimension are in Rao-Smith (#38); the best known algorithms for *exact* optimal TST in fixed dimension are in Smith-Wormald (#39, #40).

[32] A "traveling salesman tour" is a path visiting every point site exactly once. The "optimal" tour is the shortest possible such tour.

References

[1] Leonard M. Adleman: *Molecular Computation of Solutions to Combinatorial Problems*, Science 266 (Nov. 1994) 1021-1024.

[2] S. Arora: *Nearly linear time approximation schemes for Euclidean TSP and other geometric problems*, Proc. of the 38th Annual Sympos. on Foundations of Computer Science (1997). Fuller version to be published in J.ACM.

[3] M.B.Dillencourt: *Finding Hamiltonian cycles in Delaunay triangulations is NP-complete*, Discr. Applied. Math. 64,3 (1996) 207-217.

[4] M.B.Dillencourt & W.D.Smith: *A Linear-Time Algorithm for Testing the Inscribability of Trivalent Polyhedra*, Int'l J. Computational Geometry and Applications 5, 1-2 (1995) 21-36.

[5] M.B. Dillencourt & W. D. Smith: *Graph-theoretical conditions for inscribability and Delaunay realizability*, Discrete Mathematics, 161, 1-3 (December 1996) 63-77.

[6] M. B. Dillencourt & W. D. Smith: *A Simple Method for Completing Degenerate Delaunay Tessellations*, Computational Geometry: Theory & Applications. [see also Proc. 20th Int'l Colloq. Automata Languages Prog. (ICALP 1993) 177-188.]

[7] Ding-Zhu Du, F. K. Hwang: *A Proof of the Gilbert-Pollak Conjecture on the Steiner Ratio*, Algorithmica 7(2&3) (1992) 121-135.

[8] Joseph L. Gerver. *The existence of pseudocollisions in the plane*, J. Differential Equations 89 (1991) 1-68.

[9] M. Grotschel, L. Lovasz, and A. Schrijver: *Geometric algorithms and combinatorial optimization*, Springer 1988.

[10] Branko Grunbaum: *Convex Polytopes*, Interscience 1967.

[11] R.Landauer: *Minimal energy requirements in communication*, Science 272 (28 June 1996) 1914-1918.

[12] Cris Moore, Michael Lachmann, M.E.J. Newman: The Physical Limits of Communication. Manuscript available on Moore's web page `http://www.santafe.edu/ moore/`.

[13] Jonathan Schaeffer: *The History Heuristic and the Performance of Alpha-Beta Enhancements*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 11,11 (1989) 1203-1212.

[14] C.E.Shannon: *Programming a computer for playing chess*, Philo.Magazine 41,7 (1950) 256-275.

[15] Peter W. Shor: *Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Computing 26 (1997) 1484-1509.

[16] D.J. Slate & L.R.Atkin: *Chess 4.5: The Northwestern University chess program*, in P.Frey (ed.) *Chess skill in man and machine*, Springer-Verlag 1983