# The Traveling Salesman Problem and P vs. NP: Some 1960s Theoretical Work at NIST On the Complexity of Mathematical Algorithms.

Jack Edmonds <jack.n2m2m6@gmail.com>

Friday, October 10, 2014 14:00-16:00, Building 101, Lecture Room D, Gaithersburg
Friday, October 10, 2014 12:00-14:00, Room 1-4058, Boulder

**Abstract:** An informal description for a general audience
of some basic mathematical theory developed at NBS, and
a bit of reminiscing about important mathematical NBS colleagues,
Alan Hoffman, Alan Goldman, and Christoph Witzgall.

The TSP is to find an optimum way for a stylus or a salesman to move through any prescribed set of points. It turns out to still be algorithmically difficult.

The most famous of unsolved mathematical questions is still whether or not the TSP will forever remain intrinsically difficult. While researching the TSP at NBS, some other seemingly difficult algorithmic problems were nicely solved.

*Contact:* B. Cloteaux

A lovely related paper by Christoph Witzgall and NIST:
http://nvlpubs.nist.gov/nistpubs/sp958-lide/140-144.pdf
(simply google: 'nist paths, trees, and flowers')

t

I graduated from McKinley Tech High School in D.C. in 1952.
I wouldn't be here without that great school. After 3 years of university
I tried to get rich so I could be a student forever. I didn't get rich.
As a student I was an investigative reporter, directed a play,
tried to write literature, studied arts and sciences, and did very little math.
I ran errands and proof read TV schedules at the Washington Post.
I was too slow to become a journalist, and so in 1958, I chose grad school in math,
the easiest subject with the longest turn-around time.
I loved math but was a poor student, disliked the ways of academia,
and dropped out in 1960 to support a wife and kids.

I lucked on to Alan Hoffman's footsteps at **NBS,**
and to Alan Goldman as my section chief, advisor, and mentor.
And so NBS is where my scholarly studies got very serious.
Reading Hoffman, Berge, plus some negatively inspiring algorithms,
I discovered P, NP, and conjectured the thrilling "**NP ∩ coNP = P".**
I presumed that **"NP ≠ NP ∩ coNP"** was easy
and at that time not thrilling to anyone.
I'll try in this talk to motivate NP.


Jiao Tong University, Shanghai

t


Turing Memorial at Jiao Tong University

t


Stone in my front yard in Kitchener, Ontario.

t


Digging it up from in front of my house.

t

Sandblasting it.

t


A tourist attraction forever.

t


For 100 yuan ($20), I hired the best calligrapher in Beijing

t

To write my favorite thing to say,

t


**“Existence is complex.”** I am hoping that the NIST Gallery will use this, And also put some profound stones on campus.

t

In case that was not enough motivation for **"NP",** let's try some theory.

I was mystified by traditional math treatments of
**The Marriage Theorem:**
Given a set of girls, a set of boys,
and the set of pairs, (i, j), such that boy, i, loves girl, j.
The traditional way to say the marriage theorem is:

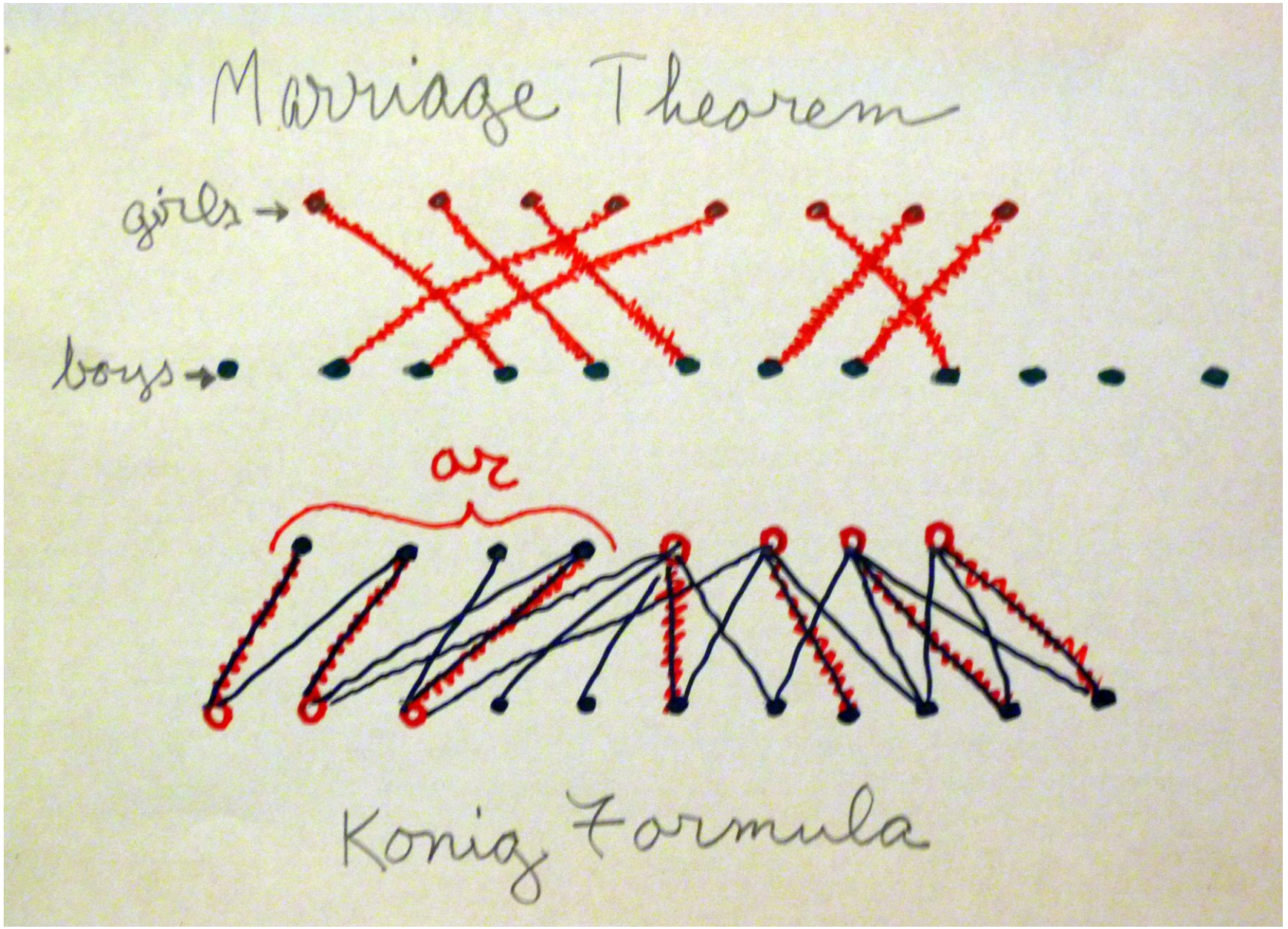
**The girls can all marry distinct boys who love them**
**If and only if, for every subset S of the girls,**
**the size of S ≤ the number of boys who love someone in S.**

Are we to see if the girls can all marry by looking at every subset of the girls?
Traditional proofs in fact seem to confirm that the question is exponentially difficult.

The **NP∩coNP** way to say the theorem begs us to prove it by a polynomial time algorithm:

***Either there is a way for all the girls to marry distinct boys***
***who love them, or else there is a subset S of the girls which is***
***bigger than the subset of boys who love someone in S. (Not both).***

t




t

In fact, a simple easy algorithm proves the more general “Konig Formula”,

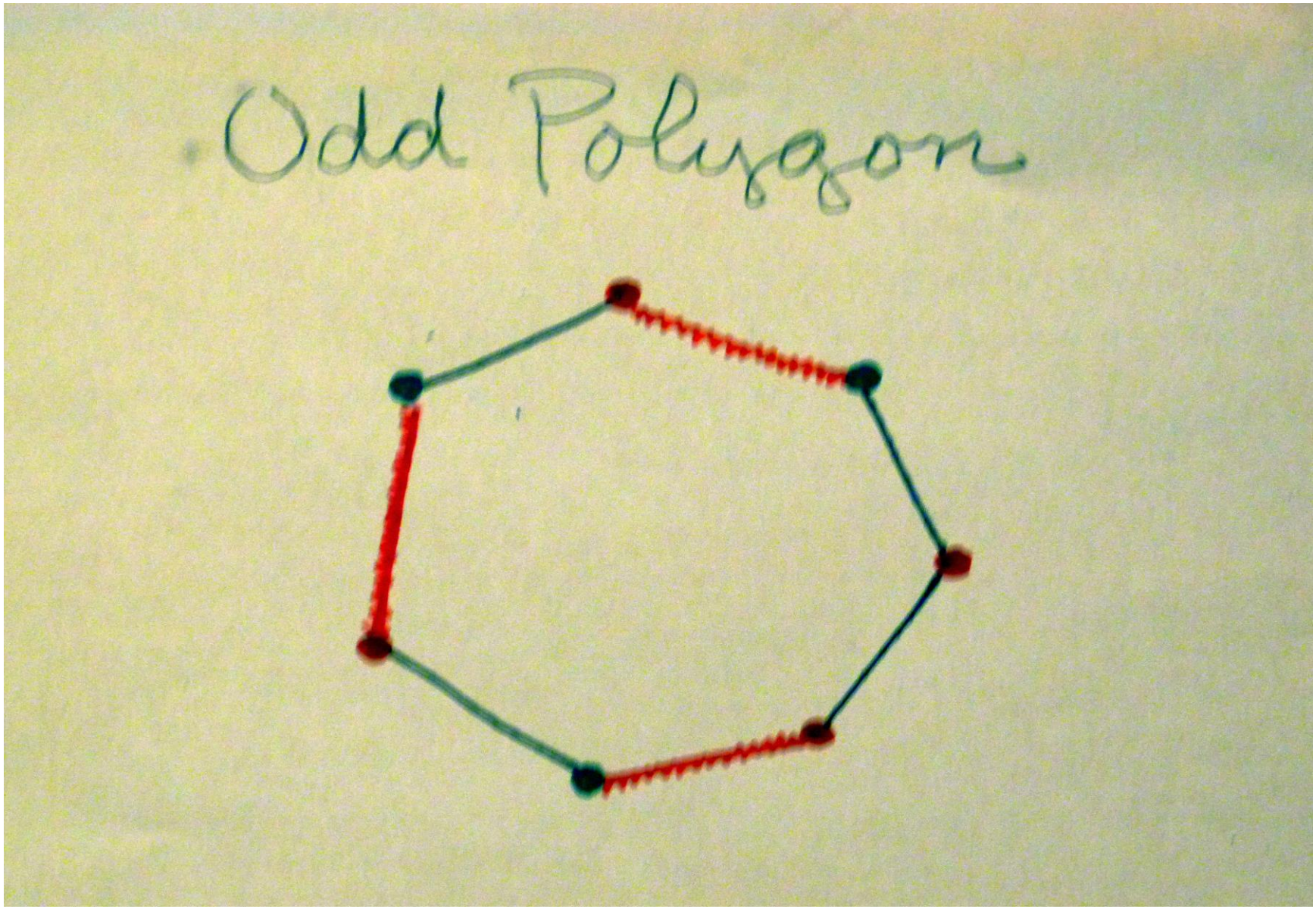
***Max size of a matching in a bipartite graph G***
***= Min size of a set of the nodes which ‘touch’ all the edges of G.***

A graph G is a set V of nodes and a set E of edges
such that each edge ‘touches’ 2 nodes.
A matching M in G is a subset of the edges such that each node of G
touches at most one edge in M.
‘G is bipartite’ means that G has 2 kinds of nodes, say boys and girls,
such that each edge of G touches 1 boy and 1 girl (who might marry).

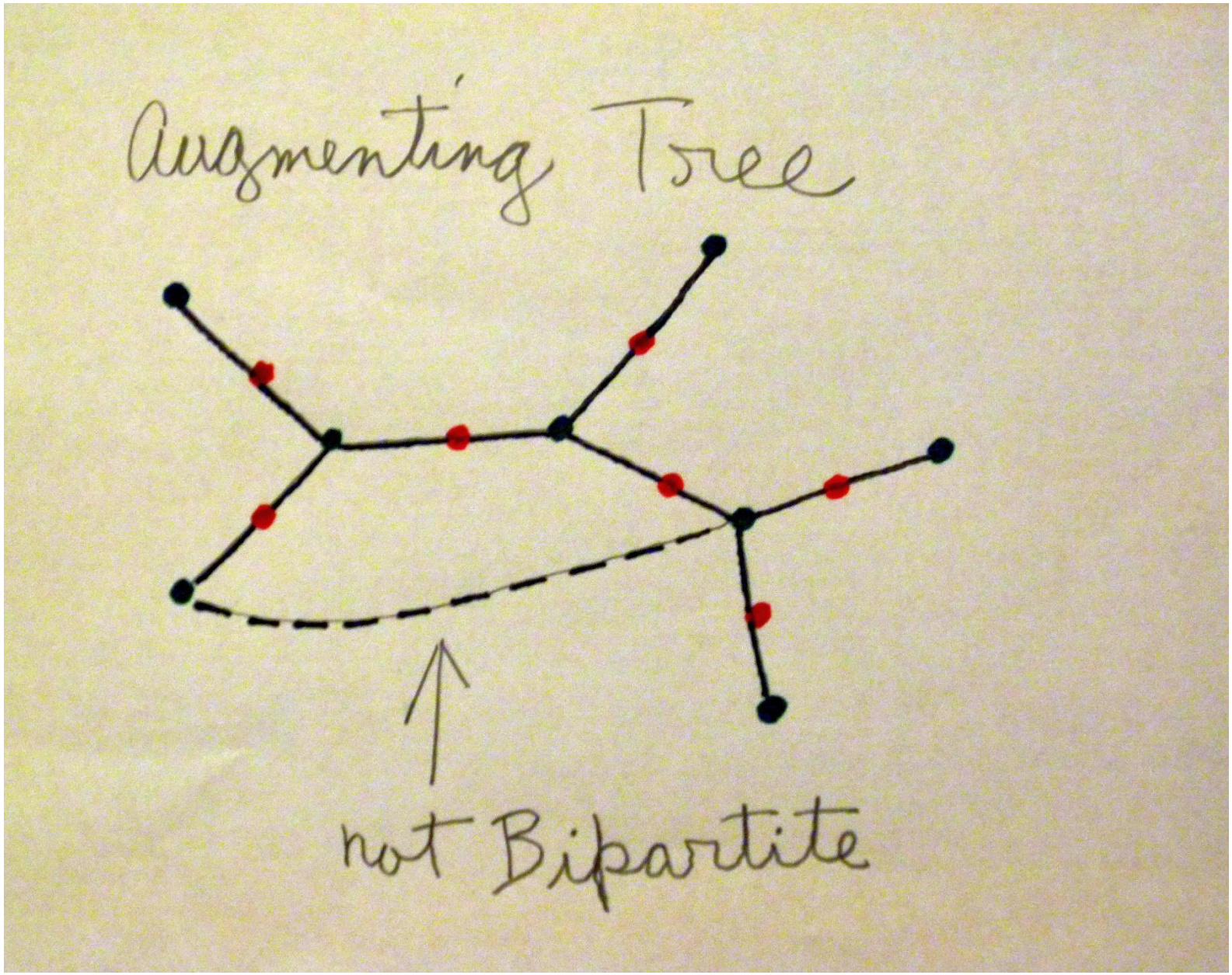
Though Max ≤ Min for any G, notice that the Konig Formula is not true where G is a simple polygon with 2k+1 nodes and 2k+1 edges. In this case we need k+1 nodes to touch all the edges but any matching has at most k edges.

It’s easy to prove by an easy algorithm that any graph G is either bipartite (i.e., has a way to partition its nodes into boys and girls) or else contains a subgraph which is an ‘odd’ polygon: find a ‘spanning forest’ F of G with nodes alternately labeled ‘boy’ and ‘girl’. If any edge, e, touches 2 boys or 2 girls
then e and the path in F joining them is an odd polygon.

t




t




t

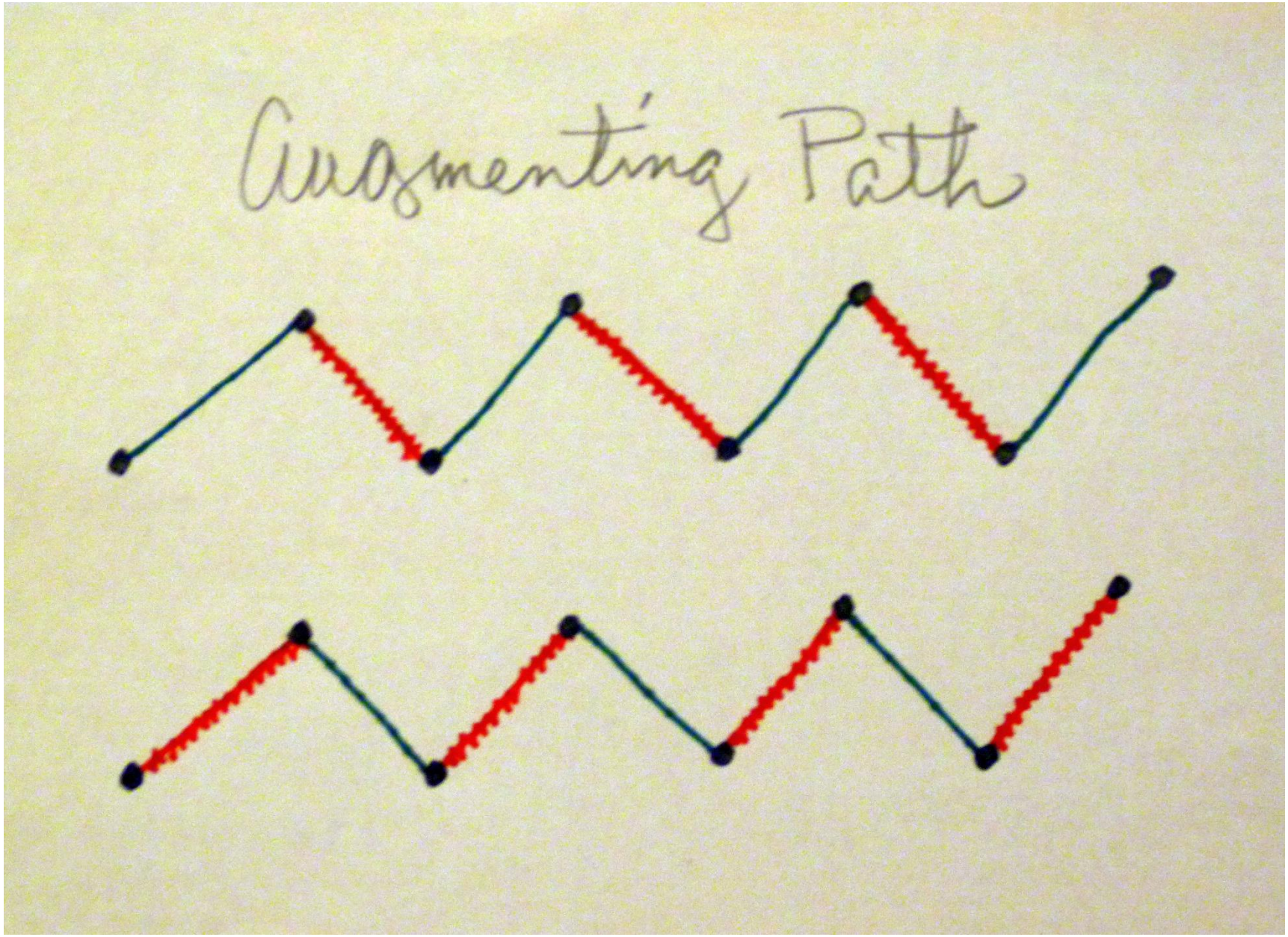
The ‘Augmenting Path Theorem’ says that:
For any matching M in any graph G either M is largest size or
there is a path p in G such that changing M by interchanging the roles of edges in p gives a matching larger than M.

A set S of nodes in a graph G is called stable
when no edge of G touches 2 nodes of S.

My first published math paper presented the following not very interesting ‘Augmenting Tree Theorem’: For any stable set S of nodes in any graph G either S is a largest size stable set in G or there is a tree T in G with nodes alternately labeled ‘inner’ and ‘outer’ such that each inner node of T touches exactly 2 edges of T and such that changing S by interchanging the roles of nodes in T gives a stable set larger than S.

I realized that the Augmenting Tree Theorem does not provide a good (i.e., NP∩coNP) characterization of a max size stable set in G
Whereas, for bipartite G, Konig’s Formula does.
A max size stable set is merely the complement of a
min size set of nodes which touches all the edges of G, and so for bipartite graphs Konig’s Formula gives a ‘good characterization’ of max size stable sets.

t




t

It's about time I say how this classic graph theory motivated the idea of NP.

,

A ***predicate*** f(x,y) means a statement which is true for some inputs, (x,y), and false for the other possible inputs, (x,y).

A predicate f(x,y) is **said to be in P** when there 8is a deterministic algorithm (i.e., Turing machine) for deciding whether or not it is true which is bounded in running time by a  polynomial function of the bit-size of the input (x,y).

A predicate g(x) **is called NP** if it can be expressed in the form
g(x) = [There is a y, not too big, such that f(x,y) is true]
where there is an algorithm in P which decides, for any input x and y, whether f(x,y) is true or false,
and where *'y, not too big'* means the bit-size of y is bounded by a polynomial in the bit-size of x.

Informally, predicate g(x) is called NP if
there is easy way to certify (prove) that g(x) is true whenever it is true.

A predicate g(x) **is called coNP** if the predicate [not g(x)] is NP.
In other words, there is an easy way to certify that g(x) is not true whenever it is not true.

t

For example suppose for predicate g(x) the input x is a bipartite graph with girl nodes and boy nodes. Suppose g(x) = [there is a y such that f(x,y)]
where f(x,y) = [y is a matching in graph x which touches all the girl nodes]
= [y is a way for all the girls to marry boys who love them].
Clearly g(x) is in NP.

Suppose g*(x) = [there is a y such that f*(x,y)] where
f*(x,y) = [y is a subset of the girls which is bigger than the set of boys who love a girl in y]. Clearly g*(x) is in NP.

The Marriage Theorem says: **For any x, either g(x) or g*(x). Not both.**
In other words, **g(x) = not g*(x).** And so **g(x) is in NP∩coNP.**

There is a good algorithm for deciding, for any x, whether or not g(x) is true.
In other words, g(x) is in P.

Figuring all this out at NBS prompted me to conjecture the thrilling **possibility that NP∩coNP = P**.

t

Suppose that x is a bipartite graph G **and** a matching M in G.
Suppose f(x,y) = [y is a matching in G which is larger than M].
Suppose g(x) = [there is a y such that f(x,y)].
Clearly g(x) is in NP.
Suppose f*(x,y) = [y is a set of nodes which touches all the edges of G and is the same size as M].
Suppose g*(x) = [there is a y such that f*(x,y)].

The Konig Formula says g(x) or g*(x), and not both.
And so g(x) is in NP∩coNP.

The 'Augmenting Path Theorem' says there is a larger matching if and only if there is a larger matching obtained in a certain way.
That is, it says that one NP predicate equals another NP predicate.
That does not tell us that g(x) is in NP∩coNP.

The theorem that any graph either is bipartite or contains an odd polygon, not both, tells us that the predicate g(x) = [x is bipartite] is in NP∩coNP.

By the way, a person quickly gets fairly able to feel when a theorem is *a good characterization,* i.e., puts some predicate into NP∩coNP, without using the formality of the definition.

t

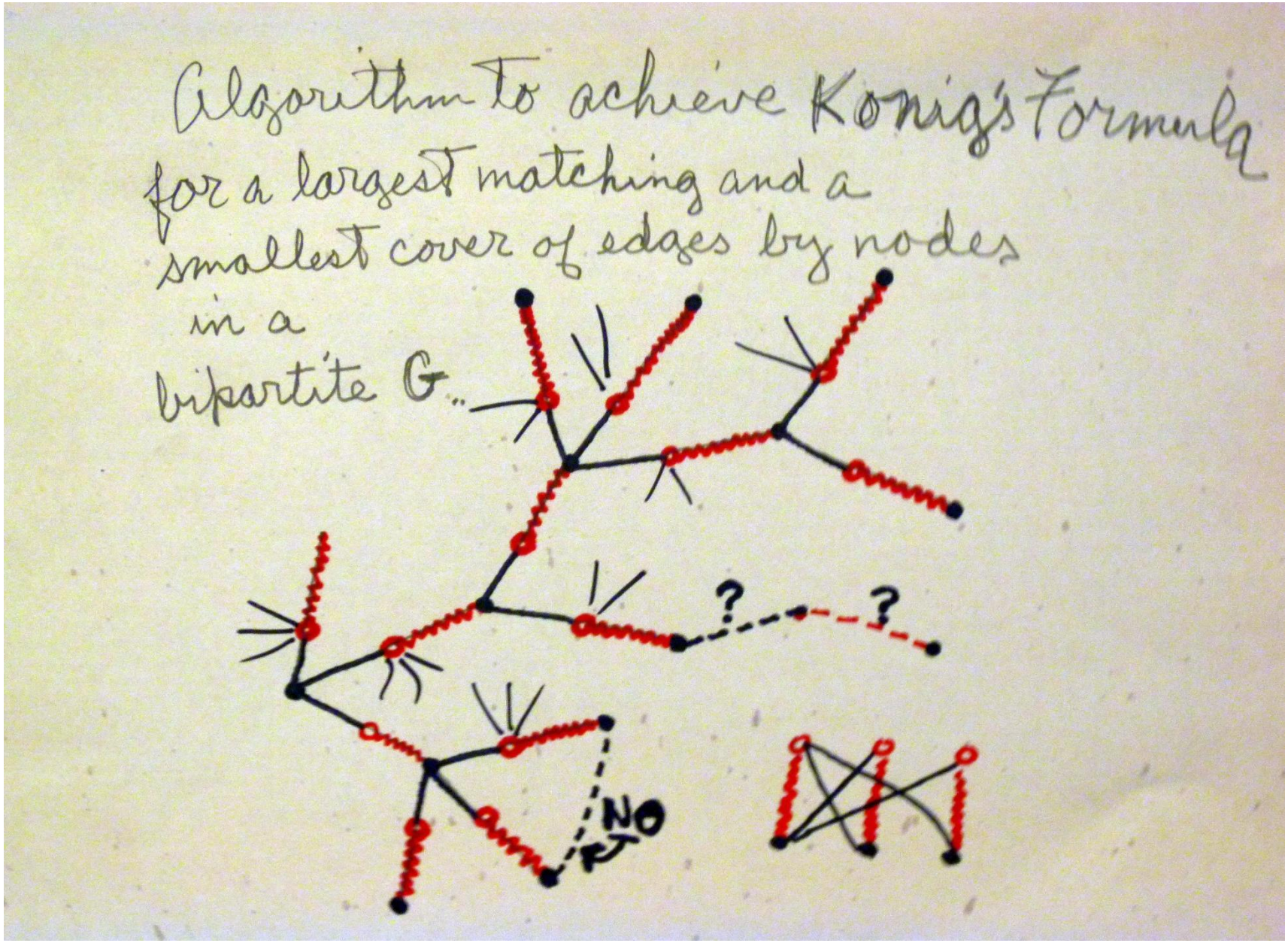
My first big NIST math was a polynomial time algorithm for finding a max size matching in any (not necessarily bipartite) graph. There is still not known, and possibly there does not exist, a polynomial time algorithm for finding a max size stable set in any graph.

Here is the polynomial time algorithm for finding a max size matching in a bipartite graph and for proving Konig's formula for a bipartite graph:

At the general step of the algorithm you have some matching M in G.
If some node v isn't hit by M, you grow from v an 'augmenting tree' T.
That leads either to an M augmenting path or else to a covering by the 'inner nodes' of T of all the edges which touch any node in T.
The number of inner nodes of T equals the number of M edges in T.
Delete T and edges touching any node in T.
Repeat on the smaller graph.

This does not work if G is not bipartite because
there might be an edge of G which touches 2 non-inner nodes of T.

t


Algorithm to achieve Konig's Formula
for a largest matching and a
smallest cover of edges by nodes
in a
bipartite G..
?
?
NO

t

Egervary's Theorem (1931) :
Given a bipartite G and a non-negative integer value $c_j$ for each edge j of G.

For any non-negative integer valuation of the nodes G,
say $y = \{y_v : v \text{ is a node of } G\}$, such that for each edge j ,
the sum of the $y_v$ on the two nodes which touch edge j is $\geq c_j$ ,
it is obvious that the total value of any matching in G is ≤
the total value of y.

Not so obviously, there is a matching M in G and
a valuation y, such the total value of M = the total value of y.
Hence max total value of a matching in G = min total value of a y.

The theorem is proved by the famous Hungarian Method for the Optimum Assignment Problem, some version taught in every Intro to Operations Research.

(The OR world was recently shocked to learn that
the method was first presented in Latin by Jacobi who died in 1851.)

Egervary did not conceive of the proof of his theorem as an algorithm.
Jacobi did not conceive of his algorithm as proving a theorem.

t

Kuhn rediscovered, and named, the Hungarian algorithm while studying Egervary's work while supported by a research grant from NBS in 1953.

Maybe it is fair to say that Egervary's Theorem is
the first published instance of the linear programming duality theorem.

I learned how Egervary's Theorem is
an instance of the linear programming duality theorem
by studying the work Alan Hoffman
published at NBS three years before I arrived there.

Hoffman's writings (1956) taught me the idea of representing
a set of 'combinatorially interesting substructures' by
a set V of 0,1 vectors, and then using a set L of linear inequalities
such that x(L), the solution set of L, is conv(V), the convex hull of V.

Together with the linear programming duality theory,
this gives combinatorial optimization and feasibility results.

t

Using total unimodularity (tu), Hoffman and Kruskal studied structures where the inequality system L can be explicitly listed.
A matrix is called totally unimodular (tu) when
the determinant of every square submatrix = 0, 1, or -1.
In particular, since the incidence matrix of a bipartite graph is tu, every vertex of the polytope defined by the L of Egervary's problem is integer valued,
and so Egervary's Thm is the LP duality theorem applied to L.

There are many theorems giving coNP descriptions of totally unimodular matrices (regular matroids), besides the definition.

My favorite problem was to get an NP description of tu.
I was thrilled when post-doc Paul Seymour did that.

t

A tour (Hamilton cycle) in a graph G is a polygon (simple cycle) in G which spans all the nodes of G. The traveling salesman problem (TSP) is given a graph G and given costs c = $\{c_j : j \text{ an edge of } G\}$
minimize the linear function cx by a point x in the set V of points,
where x is a point in V when x is the incidence vector of a tour in G.

A heuristic algorithm for tsp finds a tour which is probably at least close to being min cost.
However the TSP problem means find a tour x and **be sure** that it is min cost.
Finding the tour is usually the easier part
and being sure is the harder part of a TSP algorithm.

The TSP problem is regarded as being in NP in the sense that
there is an easy proof that a given tour x is not min cost,
namely showing some tour x' to be cheaper than x.
So far there is no known polynomial time (i.e., "easy") way,
such as an Egervary-type theorem,
to prove in general that there is no tour x' which is cheaper than x.

t

If there is no polynomial time (i.e., “easy”) way,
as I conjectured in the NBS Journal of Research in 1967,
then NP includes more predicates than does NP∩coNP,
and so then NP ≠ P since P obviously contains NP∩coNP.

There has been so much profound work by many researchers on NP versus P,
that it is plausible that NP ≠ P is true without there being any proof.

Kurt Godel, Alan Turing, Martin Davis, and others, have shown starting in the 1930s
that there are many true mathematical statements which have no proof.

In any case since the work of Steve Cook, Leonid Levin, and Dick Karp,
which I will discuss later,
which shows that TSP and apparently most other NP predicates
are as hard as any NP predicate, i.e., *NP complete.*
the conjecture that TSP is hard has become a useful axiom.

“Axiom” does not mean the mathematically obvious –
it means the unproved useful.

t

Even if the predicate g(x,G) = [x is a min cost tour in G] is not in NP,
And hence the TSP is not in P,
the situation may not be as dire as it seems because instances of the TSP which are intrinsically hard might be rather rare.

After 60 years of serious practical work by many researchers on solving large instances of the TSP problem, William Cook and his colleagues have been extraordinarily successful in doing that. Cook has also written a lot on successful practical methods, including a popular book called
**In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation, Princeton University Press, 2014.**

Any of several books called *Combinatorial Optimization* contains a lot more on the general subject we talk about here.

t

Dantzig, Fulkerson, and Johnson way back in 1956) solved an instance of the traveling salesman problem with 48 nodes which were U.S. cities,
by regarding V as the set of possible tours in a graph and by
taking L to be $\mathbf{0} \leq x \leq \mathbf{1}$ and
**subtour elimination inequalities:** For every proper subset S of nodes (i.e., cities), the sum of the variables indexed by edges leaving S is $\geq 2$.
In general, this L describes a polytope whose vertex-set includes the tours,
but also includes fractional vertices.

I confess that, unlike Bill Cook and the many others who actually try to solve instances of TSP, I never got around to reading the details of how Dantzig et al used this linear system L.

However in 1961, after studying Alan Hoffman where systems L can be explicitly listed, I was struck by the fact that the system L of subtour elimination inequalities is exponentially large compared to the size of the explicit input graph
and edge-costs. And yet L is in NP, and V, the set of tour vectors, is NP.

The only difficulty is that the polyhedron determined by this L happens to have besides V, some vertices which are fractional.

---

t

My new idea in 1961: if a set V of points has a nice (i.e., NP) description and a set L of linear inequalities has a nice (i.e., NP) description, **and** the solution-set x(L) of L is the convex hull, conv(V), of V, then using the lp duality theorem, for any linear objective, cx, we have a **"good (i.e., NP ∩ coNP) characterization"** of existence and optimality.

It seems reasonable that if V is NP then there ought to be an L that is NP and describes the hull of V. If we can discover such an L we ought to be able to prove it by a good algorithm.

I became obsessed without knowing any interesting examples. I hoped that V as the set of TSP tours would be an example, thus hopefully solving the Traveling Salesman Problem.

t

I would like to stress that NP∩coNP theorems –
that is, good characterizations,
and more generally Existentially Polytime (EP) theorems
which I'll explain in a while,
are in themselves more important to mathematics than P is.

They are not merely of interest as evidence for
the existence of polynomial time algorithms.

In fact I believe that EP, including NP∩coNP,
is a formalization of what mathematicians
most often informally regard as beautiful.

t

My NBS chief, Alan Goldman, persuaded
his PhD chief, Prof. Al Tucker at Princeton,
to invite me to be a novice participant
in a summer-long workshop on combinatorics
at the RAND Corporation in Santa Monica, 1961.

Every combinatorial big shot was there,
including a bunch of my heroes -
in particular Alan Hoffman, George Dantzig,
Ray Fulkerson, Bill Tutte,
and Claude Berge.

Higher ups at NBS determined
that taking leave from NBS
to work at the government contractor, RAND,
was against the rules.
So I quit NBS and was hired back with a raise a few months later.

t

The day before it was my turn to lecture at the RAND workshop,
I still didn't have an example of my NP∩coNP philosophy
though I had settled on the "b-matchings" in a graph as V

and, as the L, hopefully inequalities
(1) $x \geq 0$;
(2) for every node, u, [ the sum of x indexed by edges hitting u ] $\leq b_u$ ;
and
(3) for every subset S of nodes such that $\Sigma(b_u : u \text{ in } S)$ is odd,
$\sum [x_j : \text{edge j has both ends in S }] \leq [-1 + \Sigma(b_u : u \text{ in } S)] / 2$.

All I actually had was the inadequate "augmenting tree theorem"
and a speech about **NP∩coNP**.

Suddenly my officemates, Balinski and Witzgall, heard me shout
something like **"Eureka! You shrink!".**

t

With b = all 1s, and the linear objective function cx with c = all 1s,
(and assuming "total dual integrality"),
LP duality applied to these inequalities gives a
**"Konig-type min-max formula"**
for the max cardinality of a matching in a non-bipartite graph, G.
It is more transparent then the one already given by Berge.

Say that a single node v covers with weight 1 all the edges which hit v,
and that any set S of nodes of size 2k+1 covers with weight k
those edges with both ends in S.

Then
**Max size of a matching in G = Min weight of a covering of all edges of G**.

"Eureka! You Shrink!" meant that I had just figured out 'the blossom algorithm'
which proves that formula, and so presumably the same idea proves that
the preceding system L of linear inequalities
does give the convex hull of the b-matchings. (It does.)

t

My talk next morning to the high priests was a sensation.
There was some heckling. Alan Hoffman defended me.

Some question prompted me to say
“Perhaps only Prof.Tutte and God know”.
Alan said “Could one of the cited authorities comment?”

Christoph Witzgall, my office mate at the RAND workshop, and subsequently my office mate at NBS, is one of the many who helped developed the algorithmics of matchings in graphs. He had me worried when he found an alternative algorithm, published in the NBS Journal, which looked better than the one I found. Indirectly my most useful achievement was persuading him to come work at NBS.

t

I feel simply extraordinarily lucky to be cited for introducing P.
The book *Complexity Theory* by Arora and Barak even gives a long quote of my proselytizing in *Paths, Trees, and Flowers*, 1965.
Here are some parts of it:

I am claiming, as a mathematical result, the existence of a *good* algorithm for finding a maximum cardinality matching in a graph.

There is an obvious finite algorithm, but that algorithm increases in difficulty exponentially with the size of the graph. It is by no means obvious whether *or not* there exists an algorithm whose difficulty increases only algebraically with the size of the graph.

we may use something like Church's thesis in logic. Then, it is possible to ask: Does there or does there not exist an algorithm of given order of difficulty for a given class of problems?

One can find many classes of problems, besides maximum matching and its generalizations, which have algorithms of exponential order but seemingly none better. An example known to organic chemists is that of deciding whether two given graphs are isomorphic. For practical purposes the difference between algebraic and exponential order is often more crucial than the difference between finite and non-finite.

t

There is an extensive combinatorial-linear theory related on the one hand to matchings in bipartite graphs and on the other hand to linear programming. It is surveyed, from different viewpoints, by Ford and Fulkerson in **(5)** and by A. J. Hoffman in **(6)**. They mention the problem of extending this relationship to non-bipartite graphs. Section 5 does this, or at least begins to do it. There, the König theorem is generalized to a matching-duality theorem for arbitrary graphs. This theorem immediately suggests a polyhedron which in a subsequent paper **(4)** is shown to be the convex hull of the vectors associated with the matchings in a graph.

Maximum matching in non-bipartite graphs is at present unusual among combinatorial extremum problems in that it is very tractable and yet not of the "unimodular" type described in **(5** and **6)**.

In paper **(4)**, the algorithm is extended from maximizing the cardinality of a matching to maximizing for matchings the sum of weights attached to the edges. At another time, the algorithm will be extended from a capacity of one edge at each vertex to a capacity of $d_i$ edges at vertex $v_i$.

This paper is based on investigations begun with G. B. Dantzig while at the RAND Combinatorial Symposium during the summer of 1961. I am indebted to many people, at the Symposium and at the National Bureau of Standards, who have taken an interest in the matching problem. There has been much animated discussion on possible versions of an algorithm.

t

My proselytizing about NP ∩ coNP is more interesting:

We seek a good characterization of the minimum number of independent sets into which the columns of a matrix of Π can be partitioned. As the criterion of "good" for the characterization we apply the "principle of the absolute supervisor." The good characterization will describe certain information about the matrix which the supervisor can require his assistant to search out along with a minimum partition and which the supervisor can then use "with ease" to verify with mathematical certainty that the partition is indeed minimum. Having a good characterization does not mean necessarily that there is a good algorithm. The assistant might have to kill himself with work to find the information and the partition.

Theorem 1 on partitioning matroids provides the good characterization in the case of matrices of Π. The proof of the theorem provides a good algorithm in the case of matrices of Π. (We will not elaborate on how.) The theorem and the algorithm apply as well to all matroids via the matroid axioms. However, the "goodness" depends on having a good algorithm for recognizing independence.

From *Minimum Partition of a Matroid into Independent Sets,* ***J. Res. NBS, 1965.***

(I think this is the first published discussion of "NP".)

t

Let us mean by a ***good polyhedron characterization*** **(GP):**
an NP∩coNP characterization which is based on LP duality
applied to an NP set V of points and an NP set L of linear inequalities
such that all of V satisfies L and 'the vertices' of L are all in V.

After the RAND debut I did manage to find some more nice classes of GPs,
in particular based on matroids and submodularity. However in a serious search
for a GP where V is the set of vectors of tours (Hamiltonian cycles) in a graph, I failed.
E.g., adding to subtour elimination, 0,1 b-matching inequalities, or tree inequalities,
still produces fractional vertices.

In frustration I conjectured P ≠ NP,
in an obviously equivalent form which anyone can more easily appreciate:

***I conjecture that there is no good algorithm for the traveling salesman problem.***
***My reasons are the same as for any mathematical conjecture:***
***(1) It is a legitimate mathematical possibility, and (2) I do not know.***
(Optimum Branchings, **J.Res.NBS, 1967**)p

I hope that discovery of classes of GP has not been completely buried by
NP completeness. Surely there are more GPs out there.

t

I am afraid that I eventually slid a slippery slope into academia.
I went on leave without pay from NBS to be a professor.
I stayed in touch. Though I was not costing anything
I had to be riffed as part of Reagan's austerity.

At many places, I've taught the stuff I learned from working at NBS – Canada, Belgium, Germany, Denmark, France, Princeton, Cornell, Stanford, Univ. of Maryland, and in 1982, as well as a month ago, in Beijing and Shanghai.

I had great students including Peyton Young, Bill Pulleyblank, Vasek Chvatel, Bill Cook, Gilberto Calvillo, Rick Giles, Ephraim Korach, Komei Fukuda, Anna Lubiw, Kathie Cameron, Arnaldo Mandel, Julian Araoz, Jon Lee, Walter Morris, Xiaotie Deng, and many others.

t

In 1982 I gave courses to the first grad students after the Cultural Revolution. Everyone in China then wore a dark blue Mao jacket, drove a black bicycle, and was part of a one-room family. The pollution then was from the coal heating. Jazz and rock were forbidden –
I explained how these were even better than linear programming.
The students were using my slowness to improve their English.
Almost all of them then studied abroad.

The infrastructure I saw in Beijing and Shanghai a month ago is better than what I see in the U.S., though of course some people are richer here.

In Shanghai now there are neighborhoods of new million dollar mansions with absentee owners or not sold. I went last month with Guan Meigu (the Chinese Postman) to see again his childhood home which in 1982 had been split into 10 different family units including one for his ancient mother. We passed more high-end designer boutiques than I've seen in Paris.

t

I didn't learn of NP completeness until Knuth conducted a poll to name it,
and even then I didn't make sense of ***non-deterministic*** Turing machines.
I was heartbroken not to be included in the 1972 IBM workshop
on *Complexity of Computer Computations*.

Eventually I saw that the Cook-Levin NP-Completeness Theorem
is easy to prove by using

(1) the definition I knew of an NP predicate as
g(x) = **[** there is a polysize y such that f(x,y)] , f is in P;
(2) that a Turing machine for fixed size input (x,y)
is a polynomial size Boolean circuit; and
(3) that Boolean circuit satisfiability reduces to cnf formula satisfiability.

B reduces to A means there is way to get a polytime algorithm for B
by using a polytime algorithm for A. We can also then say that A is B hard.
Of course many problems are solved by reductions.

The Chinese Postman's problem can be reduced to the shortest problem and
the 1-matching problem. The b-matching problem is polytime solved by reducing it to
the optimum flow problem and the 1-matching problem.

t

There has been great success in using the conjecture NP∩coNP = P as a template for special cases. The conjecture prompted my GP math, as well as famous successes like linear programming, and PRIME (deciding whether a number is prime). Still there are some NP ∩ coNP theorems for which good deterministic algorithms are not known.

Anne Condon wrote in 1992: "Although many number theoretic problems not known to be in P lie in the class NP∩coNP, combinatorial problems that lie between P and NP∩coNP are rare." Then she goes on to describe a good candidate: Simple Stochastic Games:
*The complexity of stochastic games.* Information and Computation, 96:203-224, 1992.

Daniel Andersson and Peter Bro Miltersen.
*The complexity of solving stochastic games on graphs*.
20th International Symposium, ISAAC 2009, December 16-18, Hawaii,
Proceedings, Volume 5878 of Lecture Notes in Computer Science.

The simplest example of a problem in NP∩coNP conjectured not to be in P might be Integer Factorization.
(Cryptographers would hate Integer Factorization being in P, even though it already is with quantum computing.)

t

Define the following decision version of Integer Factorization:

Given two natural numbers n and k,
decide whether there is a prime factor p of n with $p \leq k$.

If this task was in P, then one could use binary search
to find the smallest prime factor of n efficiently;
thus factoring n in polynomial time.

On the other hand, the above problem is in $NP \cap coNP$:
The witness for both Yes-instances and No-instances
is just the integer factorization of n itself.

Indeed, given numbers $p_1,\ldots,p_t$, it is easy to check that

(i) Their product is n, and
(ii) They are all prime, since Primes is in P.
(iii) Finally, one can check whether $\min \{ p_1,\ldots,p_t \} \leq k$.

By the same token, if a computational task always has a unique solution and a solution is efficiently verifiable, then one can cook up a NP∩coNP problem by simply asking for the $k^{th}$ bit of the solution.
In this way, every bijective one-way function
would give a decision problem in (NP∩coNP) – P.

---

Maybe I should hedge my bet by
weakening the conjecture NP∩coNP = P to
**"GP, i.e., good polyhedral characterization, is in P",**
since these have been my only successes
(and we don't want to threaten national security).

Could we show that NP∩coNP reduces to GP?
Might we more modestly show that Integer Factorization reduces to GP?
That would be a lovely GP, regardless of whether it is in P.

t

An existentially polytime (EP) theorem means a theorem of the form
**"For any x, there is a polynomial size y such that f(x,y)",**
where f(x,y) is in P,
Most of the revered theorems of combinatorics are EP.

Example. Dirac's Thm: In any graph G, there is a Hamilton cycle in G
or a node joined to fewer than half the other nodes.

Of course NP∩coNP theorems are EP.
Most EP theorems seem to have polytime algorithms for finding a y,
such as "derandomization" for EP theorems
which are proved by inequality-type counting.

However Papadimitriou has famously identified 2 classes of beautiful EP
theorems based on parity which have beautiful algorithmic proofs
(that there is a y) but seem not to have polytime algorithms:
PPA theorems and PPAD theorems.

Chen and Deng have famously shown that 2NASH,
i.e., finding a 2-person Nash equilibrium, is PPAD complete.

t

By reducing from 2NASH, Vlad Gurvich and I have shown that
Polytopal Sperner is PPAD complete.

"Manifold Sperner" Theorem. For any simplicial (pseudo) d-manifold M,
and any coloring of the vertices of M with d+1 colors,
and any chosen one of the colors,
there is a natural pairing of the rainbow rooms.

A simplicial (pseudo) d-manifold M means a finite set V of vertices
and a set R of size d+1 subsets of V, called the rooms,
such that each size d subset of a room, called a wall, is a wall of exactly 2 rooms.

A room is rainbow means that it has one vertex of each color.
The associated search problem is to find the brother of a given rainbow room.

Polytopal Sperner is where the d-manifold M is the simplicial polytope boundary
of the convex hull of a set V of points in general position in d+1 space.

There are many beautiful, and apparently algorithmically exponential, PPA theorems,
but currently no known PPA search problem
which does not use abstract Boolean circuits.

Zeying Xu, Xiaotie Deng, and I, are working
on showing a non-oriented geometrical Sperner to be PPA complete.

t

Here is a nice polyhedral EP theorem:
**Any graph G has either**
**an induced odd hole, an induced odd anti-hole,**
**or a stable set S and a set C of cliques the same size as S**
**such that C covers all the nodes of G.**

An odd hole is a polygon with no chords and
the number of its nodes odd and greater than 3.
An odd anti-hole is the complementary graph of an odd hole.

There should be a good direct algorithm which proves the theorem
by finding in any graph an instance of what is said to exist.

The problem at least has a very long indirect solution
by putting together four different long works.

Thanks for listening.

t