Some books on Combinatorial Optimization:

- J. Lee, *A First Course in Combinatorial Optimization*, Cambridge University Press, 2004.
-
- W. Cook, W. Cunningham, W. Pulleyblank and A. Schrijver, *Combinatorial Optimization*.
-
- C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.
-
- E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.
-
- G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.
-
- B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Algorithms and Combinatorics 21 Springer, Berlin Heidelberg New York, 2012.
-
- 3-volume book by A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency* , Springer-Verlag, 2003.

I will talk informally about a 60 year old philosophy
behind the mathematics of these books,
which the books themselves do not address very much.

A humanities professor is said to “read a paper” at a humanities conference.

A President gives a speech from “cue cards”,
the text already available to the media.

I’ll will show you the cue cards and read them with you.
You can read without listening or listen without reading.

They will be available on the internet
from my wonderful hosts, Professors Alexandre Francisco,
José Rui Figueira, and Luis Russo.

Origins of NP and P

Jack Edmonds

Lisbon, February, 2019

NP and P have origins in “the marriage theorem”:
A matchmaker has as clients the parents of some boys
and some girls where some boy-girl pairs love each other.
The matchmaker must find a marriage
of all the girls to distinct boys they love
or else prove to the parents that it is not possible.

The input to this marriage problem
is usually imagined as a bipartite graph G
with boy nodes, girl nodes,
and edges between them representing love.

A possible legal marriage of <u>some</u> of the girls to <u>some</u> of the boys
is represented by a subset M of the edges of G, called a matching.
The matchmaker's problem is to find a matching
which hits all the girl nodes
Or else prove to the parents that there is none.

Let G be a *bipartite graph* with boy nodes and girl nodes.
Each boy is hit by the edges to the girls who love him.
Each girl is hit by the edges to the boys she loves.

A *matching* M in G is a subset of its edges such that no node of G is hit by more than one edge of M.

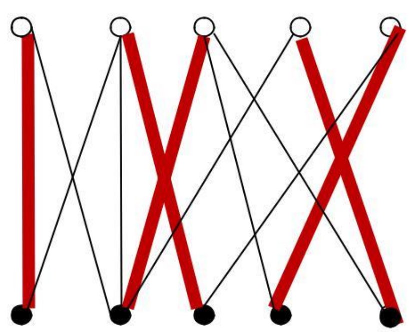
The Marriage Theorem (implicit in work of Frobenius with many variants since then) :

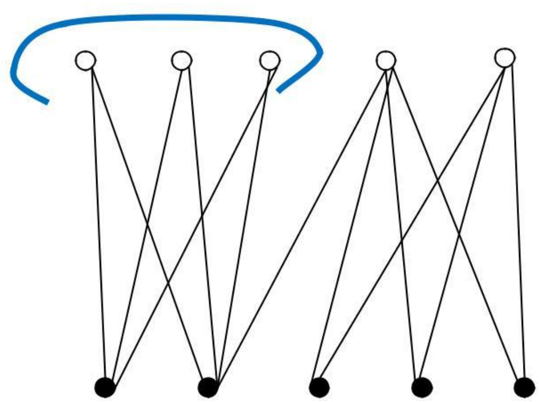
$A_1(G)$ = [The girls can marry distinct boys they love,
i.e., there is a matching M in G which hits all the boys],

or $A_2(G)$ = [there is a subset S of the girls which is bigger than the set of boys who someone in S loves,
i.e., bigger than the set of boys which are joined by an edge of G to a member of S.]
Not both.

The Marriage Theorem: $A_1(G)$ = not $A_2(G)$.

And so **$A_1(G)$ ϵ NP∩coNP**, since both $A_1(G)$ and $A_2(G)$ are clearly in NP.





From Matt Baker: https://mattbaker.blog/2014/06/25/the-mathematics-of-marriage/

Another gorgeous application of the marriage theorem:

Consider the following game of solitaire:
you deal a deck of cards into 13 piles of 4 cards each,
and your goal is to select one card from each pile
so that no value (Ace through King) is repeated.
It is a beautiful mathematical fact
that this can always been done,
no matter how the cards were originally dealt!
Matt deduces this from the **Marriage Theorem** which says

Either the girls can marry distinct boys they love
or there is a subset S of the girls such that
the number of boys which someone in S loves is smaller than S.
Not both.

Mathies like strange proofs,
such as the proof Matt attributes to me using linear algebra.
The proofs which mathies love usually convey an impression
that there is no easy way to actually get what the theorem says exists.

The best proof of the marriage theorem is a simple algorithm,
polynomial time relative to the number of girls and boys,
which always finds either marriages of all the girls to distinct boys they love
or else a subset S of the girls which together do not love enough boys.

In fact it provides an easy algorithm to select one card from
each of the 13 piles of 4 so that no value (Ace through King) is repeated.
(Maybe not so easy to do in your head, but with mental aids
far easier than trying 4^{13} possibilities in a back tracking way)

Algorithmic Proof of the Marriage Theorem: Let M_1 be any matching in G.

If there is some girl node r who is not hit by M_1
grow a tree T such that every path in T starting at node r
alternates between edges not in M_1 and edges in M_1.

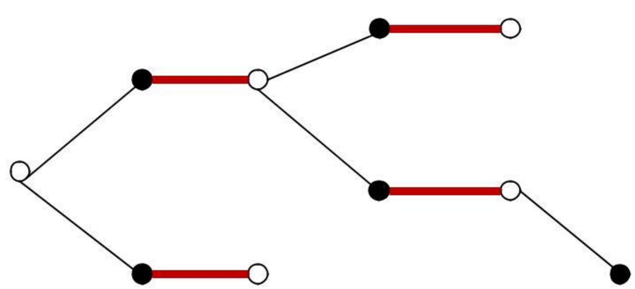
If T reaches boy node t which is not hit by matching M_1,

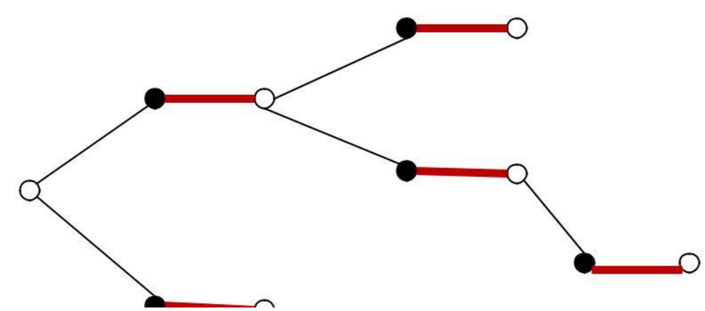
then change edges in the path in T between t and r to get a larger matching, M_2.

If T reaches a state of not being able to grow more

and not containing a node t not hit by M_1, then the boy nodes of T are the only nodes of G which
the girl nodes of T are joined to by edges of G,
and there is one more girl node in T than boy nodes in T.

And so [A_1(G) ? or A_2(G) ?] **ϵ P.**

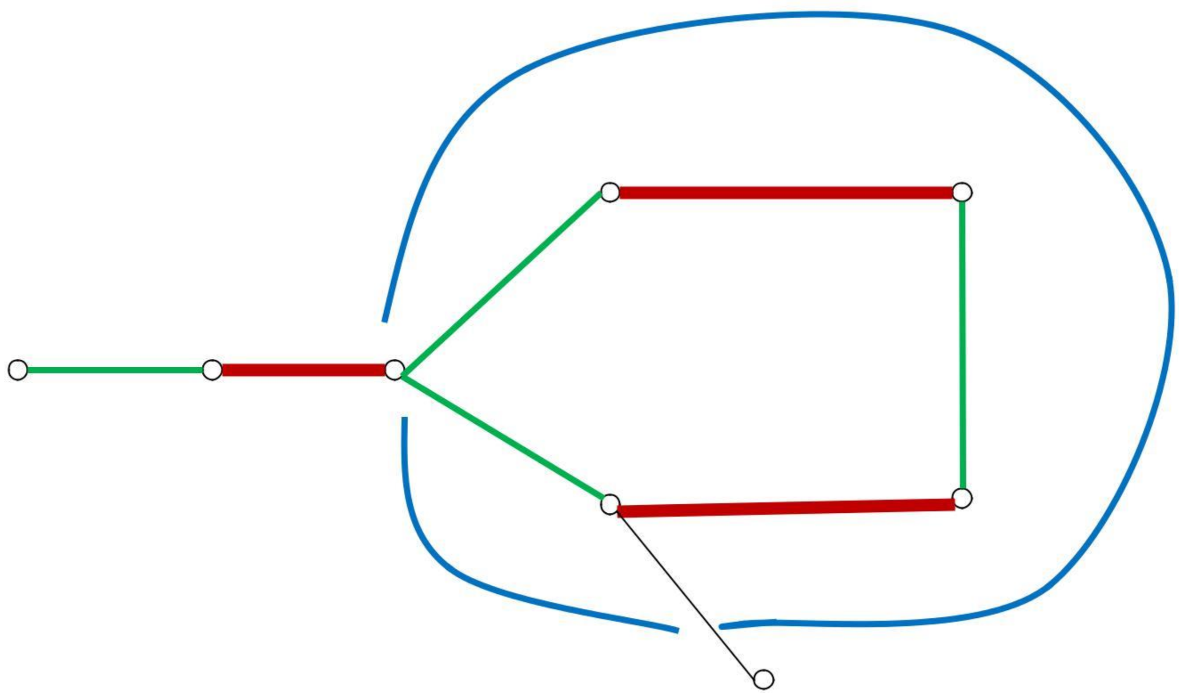




If graph G is not bipartite this algorithm is messed up
by the discovery of an edge e joining two “girl nodes” of T.
Roughly speaking, the secret to patching up the algorithm for non-bipartite use is:

T+e contains a cycle C (called a blossom) with an odd number of edges.
Shrink blossom C to become a “pseudo girl node” of T and continue growing T.

Eureka! You shrink.



A predicate p(t) is a statement with a variable input t.
For example, p(t) might be the statement:
“There is a matching in bipartite graph, t, which hits all the girl nodes.”
This particular predicate p(t) is in complexity class NP because
whenever it is true there is an easy way relative to the size of t
to show the parents that it is true.

This predicate is also in coNP because the marriage theorem provides,
whenever t is such that p(t) is not true,
an easy way to show the parents
that there is no matching in G which hits all the girl nodes.
(Of course we need a mathematical definition of “easy”.)

It turns out that this predicate p(t) is also in complexity class, P.
That is, relative to the size of t, there is an easy algorithm to decide whether or not p(t) is true.

Thinking about this in 1960, it occurred to me that maybe a general predicate p(t) is in P whenever it is in NP ∩ coNP.

It is still not known whether or not **P = NP** ∩ **coNP** in general but it has turned out to be often true. I think this is the origin of NP.

The optimum **assignment problem**, in a yes-no form, is given
t = (a bipartite G, a numerical weight cj for each edge j, a matching M in G),

is p(t) true or false, where p(t) is the statement
“M is not the largest total weight matching in G”?

Obviously this predicate p(t) is in NP because whenever p is true
we can show it easily by showing a larger weight matching.

Is this p in coNP? i.e. is “M is a maximum weight matching in G” in NP?

The 1931 Egervary Theorem says yes
because it provides an easy way to prove when an M is optimum.

Egervary's Theorem, 1931, says:
For any bipartite G where each edge e has a value c_e , the max total value of any matching in G = min sum of numbers $y_v \geq 0$ for nodes v of G such that for every edge e of G,
the sum of the y_v on the two end nodes of each edge e is $\geq c_e$.

Egervary's theorem is equivalent to: For any bipartite graph G, the convex hull of the incidence vectors of matchings in G is the solution set of the following system L of linear inequalities:
for each edge e of G, $X_e \geq 0$;
for each node v of G, sum of the x_e on all the edges e hitting node v is ≤ 1.

The matchings in a non-bipartite graph satisfy this system L
but there are extreme points of the determined polytope of this L which are fractional.

For any graph G a matching M in G is
an integer-valued vector $x \geq 0$ with a coordinate for each edge e of G such that
(1) for every node v of G, $\sum \{x_e$: edge e hits node v$\} \leq 1$.

The Matching Polytope Theorem says that the convex hull of the matchings in G
is the solution set of $x \geq 0$, inequalities (1), <u>and</u>
(2) for every subset S of nodes,
$\sum \{ x_e$: edge e has both ends in S $\} \leq$ integer part of half $|S|$.

Implicit in Egervary's proof is an easy algorithm
for deciding whether or not the matching M is optimum,
and so this predicate p(t) is in complexity class P.
In 1954 Kuhn dubbed it "the Hungarian Method".

Silvano Martello and friends have a book called
"Assignment Problems".
http://www.or.deis.unibo.it/staff_pages/martello/cvitae.html

The marriage problem is treated in books on
"Graph Theory" and most introductions to "Discrete Math".

The assignment problem and generalizations are treated
in many fine books on "Operations Research",
"Network Flows", and "Combinatorial Optimization".

Egervary's Theorem is the first known instance of
the linear programming duality theorem.
George Dantzig introduced the general field of linear programming,
and invented the simplex method for solving linear programs,
in the late 1940s.

A thrill of my life was in the mid 1960s showing to Dantzig's l.p. class
that the simplex method can grow exponentially
even for shortest path problems, and so it does not put l.p. into P.
In the 1980s Khachian rocked the world by showing
that "the ellipsoid method" <u>does put</u> l.p. into P.
(The Hungarian method is <u>not</u> the simplex method or the ellipsoid method.)

Linear programming has become since the 1950s a foundation
of departments of operations research, management science,
or industrial engineering.

The matchings M in a graph G are represented by vectors x
with a coordinate for each edge j in the edge-set E of G.

The “(Incidence) vector” x of any set M of edges in G
is $x_j = 1$ if j of E is in M, and $x_j = 0$ if j of E is not in M.
Thus an optimum matching is the vector x of a matching M
which maximizes a given linear function, cx, over matching vectors, x.

The convex hull of any finite set Q of points (vectors) in the space
of vectors indexed by the edge-set E of graph G, is a polytope,
the bounded solution-set of a finite set L of linear inequalities.
Thus optimizing cx over members of Q is the same as
the linear program of optimizing cx subject to L.

Normally for an easily described finite set Q of points in R^E,
the number of inequalities needed in L is exponentially large relative to $|E|$.
Dantzig’s simplex algorithm would then be very exponential time.

An arrangement of the girls and boys around a banquet table
so that each is, on each side, next to someone loved,
is the same as a 'Hamilton tour' in the graph G.
Suppose we want a Hamilton tour C
which optimizes the sum of love-weights on edges in C.

The Traveling Salesman Problem, in a yes-no form, is given
t = (a graph G, a love-weight c_j for each edge j, a Hamilton cycle H in G),
then where p(t) = [H is not the largest total love-weight Hamilton tour in G],
is p(t) true or false? This predicate is clearly in NP. Is it in coNP?

TSP is represented by the set Q of the 0-1 valued vectors x
of the Hamilton tours in G and a linear function cx to be optimized over Q.
Is there an **easily described** set L of linear inequalities such that
optimizing cx over solutions to L is the same as optimizing cx over Q?

In 1954, Dantzig, with Ray Fulkerson and Selmer Johnson,
solved a particular 48-city traveling salesman problem
by using linear programming.

Their linear inequality system L’,
satisfied by all the Hamilton tour vectors, is easy to describe,
though exponentially large.
Unfortunately the solution-set of L’ has some extreme points
which are fractional rather than Hamilton tour vectors,
and so optimizing subject to L’ might not be by a Hamilton tour.

Their work motivated Ralph Gomory in 1958
to introduce cutting plane methods for integer linear programming.
There was no mention of ‘easy’, meaning polynomially bounded time.

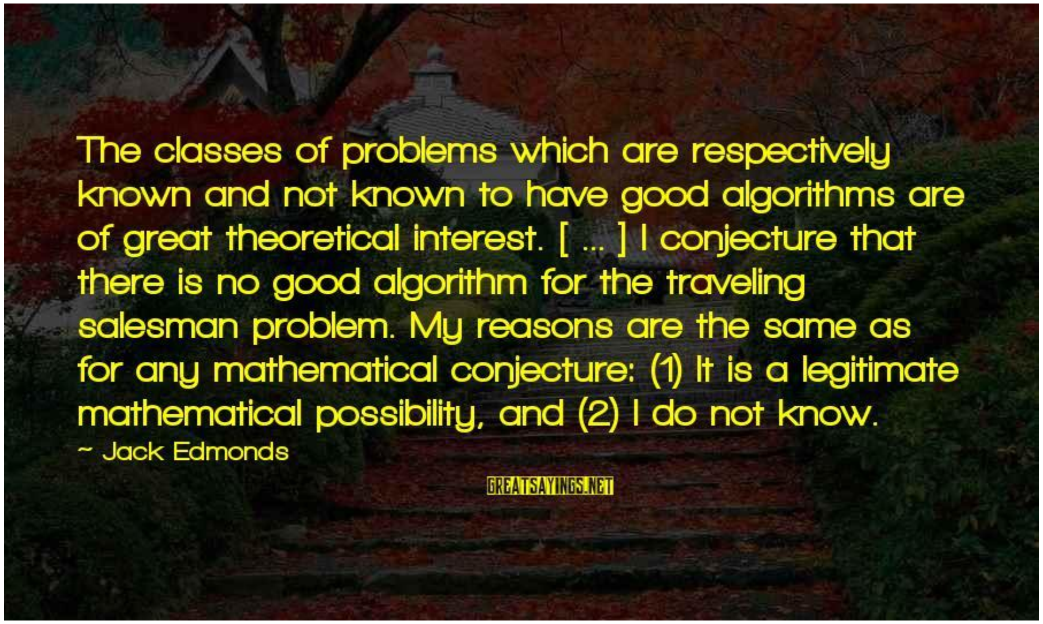
Their work prompted me in 1961 to notice the simple result that
Theorem. If there is an easy (i.e., NP) description of the set L of inequalities describing the convex hull of finite point-set Q, as well as an easy description of the members of Q, then it does not matter that L is exponentially large in order to have that the predicate p(t) = [point x of Q is optimum over Q] is in NP $\cap$ coNP.

And maybe that puts "optimize over Q" algorithmically into P.

It seemed plausible that if there is an easy description of Q then there might be an easy description of a linear system, L, describing the convex hull of Q. That is an important step toward an easy general algorithm for optimizing over Q.

I tried hard to find an NP description of a linear system L whose solution-set is the convex hull of the vectors of Hamilton tours in a complete graph, G.

I was never able to, and so I conjectured in **1966** that **NP ≠ P**, in particular that TSP is not in P.

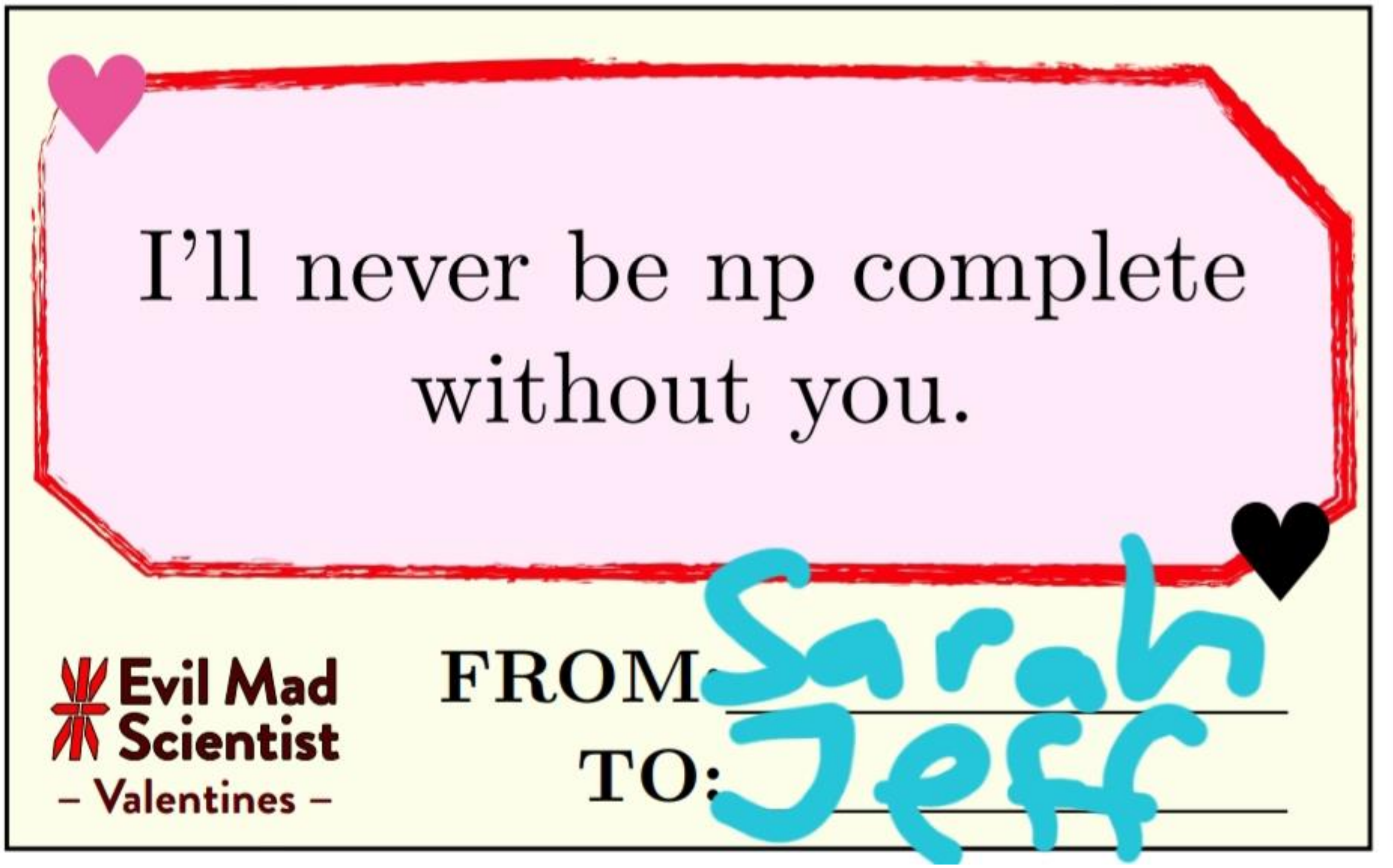


The conjecture is now usefully presumed.
It is at the top of the Clay list of unsolved math problems.
Curiously it is the only problem in the list
which is posed as a question rather than as a conjecture,
so it does not need to be called “Edmonds’ conjecture”.

In 1971 Steve Cook, and independently Leonid Levin,
showed that there exist NP complete predicates (“problems”).
Dick Karp showed that many natural combinatorial problems,
including the Traveling Salesman Problem, are NP complete.

And hence an easy algorithm for any one of them
implies an easy algorithm for any NP predicate.
Hence **NP ≠ P** implies that there is no easy algorithm for any of them.

I might as well conjecture the mathematically valid possibility
that there is no mathematical proof of **NP ≠ P**.
For many mathematically true statements,
there cannot exist any mathematical proof.
Having such a proof does not really matter.
It is as true as is generally accepted,
as long as we do not have a polytime algorithm for an NP complete problem.



Instead, how about:

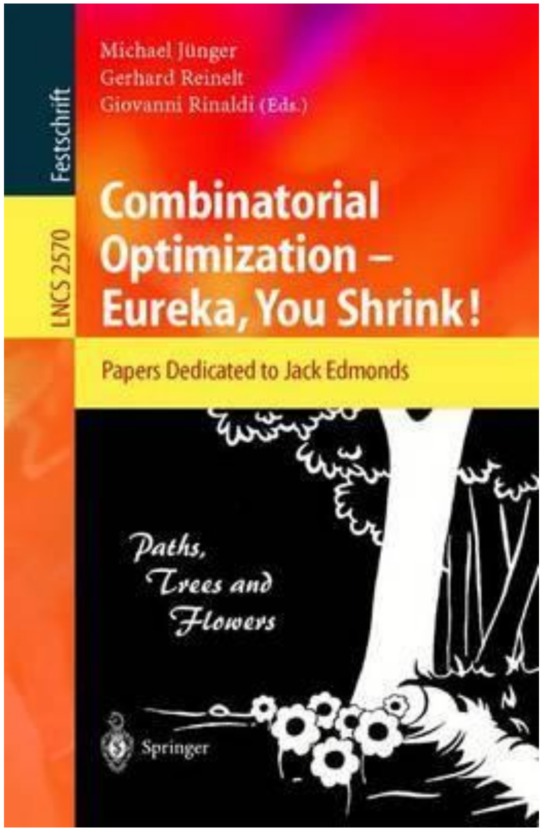
"I'll never be NP ∩ coNP without you." ?

Throughout the 1960s , though in looking for easy algorithms
of course I found some polytime reductions between problems,
I never dreamed that there exists NP <u>complete</u> problems.
It is now a cosmic tragedy that
so many NP complete problems have even given NP a bad name,
when NP itself is a wonderfully positive thing.

During the 1960s, as part of the search for TSP in P,
I did find some instances of easily described sets Q of points
to have NP descriptions of convex-hull determining linear systems, L,
and P algorithms.

One of these, which solves the useful “Chinese Postman’s Problem”, is the
“optimum matching problem” in a graph G which is not necessarily bipartite,
i.e., the optimum gay marriage problem,
by algorithmically proving an easy description of “the matching polytope”.
This was hoped to be an ingredient for solving TSP, though apparently it is not.

In 1961 while working on combinatorial optimization problems in the new Operations Research Section of the Mathematics Division of the U.S.



National Bureau of Standards, my wonderful mentor Alan Goldman arranged through his wonderful Princeton mentor, Professor Tucker, thesis supervisor of Nash's equilibria and much else, for me to be a junior participant in a Summer long workshop at the Rand Corporation, across from Muscle Beach in California. Besides the utter novice me, seemingly every known combinatorist participated.

I needed a successful example of the NP idea, such as the non-bipartite matching problem. The day before my scheduled lecture to the eminences I still did not have it. Then Eureka, I did! You shrink blossoms! So the lecture was a success, and attracted discussion. Professor Tucker offered me a job at Princeton, chairing the Combinatorics and Games Seminar.

Another P success, which led to matroid theory successes, was
a polynomial time way to solve a system of linear equations, $Ax = b$.
Well-known Gaussian elimination is not polynomial time
since the number of bits doubles as each column is treated.
The linear elimination algorithm in the very old paper,
Systems of Distinct Representatives and Linear Algebra, on the internet,
is polytime for integer entries but not with indeterminate symbol entries.
Thus there is a polynomial time **randomized algorithm**
for testing non-singularity of the matrix A by substituting integers for symbols.

A still unanswered question: Is there a polytime deterministic algorithm
for testing non-singularity of a matrix with indeterminate symbol entries?
The answer is yes for certain cases
– in particular where the entries of A are zeroes and distinct symbols.
How? By the marriage problem algorithm.
What about when the symbolic entries of A are not all distinct?

An independence system M=(E,F) : F is a family of subsets,
called independent, of the ground set E,
such that $\phi \in F$ and $(A \subseteq B \in F) \Longrightarrow (A \in F)$.

A matroid M=(E,F) is an independence system such that, for every $S \subseteq E$,
every maximal independent subset of S is the same size, called the rank $r_M(S)$ of S,

like the linearly independent subsets of the column set E of a matrix
or like the forests of the edge set E of a graph.

Matroid polytope: $P_M = \{ x \geq 0 : \text{for every } S \subseteq E , \sum \{ x_e : e \in S \} \leq r_M(S) \}$.
Theorem. The vertices (extreme points) of P_M are the vectors x^J of $J \in F$.
Proved by, and proving, the greedy algorithm.

The amazing
Matroid Polytope Intersection Theorem (1969):
For any two matroids, M1 = (E,F1) and M2 = (E,F2) ,
the vertices of $P_{M1} \cap P_{M2}$ are the vectors x^J of $J \in F1 \cap F2$.

For example, **branching systems in a network**.
Matroidal Optimization generalizes to polymatroid polytopes:

Polymatroid set function f(S): $2^E \equiv$ {subsets S of set E} --> R ≡ {real numbers},
such that for all $A \subseteq E$ and $B \subseteq E$
normalized: $r(\emptyset) = 0$;
non-decreasing: $0 \le r(A) \le r(A \cup B)$; and
submodular: $r(A \cup B) + r(A \cap B) \le r(A) + r(B)$.

The submodularity condition is equivalent to
$f(S \cup \{s,t\}) - f(S \cup \{t\}) \le f(S \cup \{s\}) - f(S)$,
or, where $S \subseteq T \subseteq E$,
$f(T \cup \{s\}) - f(T) \le f(S \cup \{s\}) - f(S)$.
That is: **any marginal increase in f(S) is non-increasing with any "increase" of S.**
This provides for 'economics applications' which we do not pursue.

Theorem: For any integer valued polymatroid function f of 2^E,
M = (E,F) is a matroid if [$J \varepsilon F \Longleftrightarrow |S| \le f(S)$ for every $S \subseteq J$] .
Linearly independent sets of columns in matrices Mi are the independent sets of matroids Mi.
$f(S) = \sum r_{Mi}(S)$ is an integer polymatroid function yielding this M as a "sum" or "union" of the Mi.

Matroid Partition Theorem (1966). For any matroids Mi = (E, Fi), and for any $J \subseteq E$,
either there is a partition of J into sets Ji ε Fi ,
or there is some $S \subseteq J$ such that $|S| > f(S) \equiv \Sigma\, ri(S)$. Not both.

Another origin of NP: In 1964, I say in the paper
"Minimum Partition of a Matroid Into Independent Subsets",

Of course, by carrying out the monotonic coloring procedure described above in all possible ways for a given matrix, one can be assured of encountering such a partition for the matrix, but this would entail a horrendous amount of work. We seek an algorithm for which the work involved increases only algebraically with the size of the matrix to which it is applied, where we regard the size of a matrix as increasing only linearly with the number of columns, the number of rows, and the characteristic of the field. As in most combinatorial problems, finding a finite algorithm is trivial but finding an algorithm which meets this condition for practical feasibility is not trivial.

We seek a good characterization of the minimum number of independent sets into which the columns of a matrix 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.
The proof of the theorem is a "good algorithm" for finding a partition into k independent sets or finding an easy proof for the supervisor that there is no such partition.

Polynomial Time algorithms for proving The Matroid Partitioning Theorem and The Matroid Polytope Intersection Theorem enable us to solve the following useful Optimum Branching Systems Problem:

For a directed graph G with a cost for each edge and with specified root nodes r_i in G, find a least cost collection of edge-disjoint spanning directed trees rooted at the r_i

Perfect Graph Polytopes.

Graph G. Independence system Q = (E, F) where E is the node-set of G and
F is the independent subsets J of nodes in G (no two nodes of J joined by an edge of G).
A clique C in G means a subset C of nodes in G
such that every pair of nodes in G is joined by an edge of G

This Q = (E, F) is rarely a matroid. However **a graph G is called perfect** if the vertices of $P_Q \equiv \{ x \geq 0 : \text{for every clique } C \text{ in } G , \sum \{ x_v : v \in C \} \leq 1 \}$ **are the vectors x^J of J ε F.**

Example: comparability graph, i.e., the undirected graph of a partial order, where edges correspond to < . (Dilworth's theorem)

Example: E is a set of subtrees of a tree.
J ε F when members of J are mutually disjoint trees ("chordal graphs").

Example: Egervary's Theorem.

Theorem: Polytope P_Q is perfect iff, for every induced subgraph G' of G,
$\max \{ |J| : J \text{ of } G', J \in F \} = \min \{ |K| : \text{cover } K \text{ of } G' \text{ by cliques of } G' \}$.

Notice that all of these broad classes of “good characterization” NP∩coNP polytopes include Egervary’s polytopes. What about “Egervary complete polytopes”?

Thank you. Please come to the course.

Jack, <jack.n2m2m6@gmail.com>. My office where you can find me is Room 803, Holiday Inn.