(2)

MATCHING THEORY - A SAMPLER: FROM DÉNES KŌNIG TO THE PRESENT

by

Michael D. Plummer*
Department of Mathematics
Vanderbilt University
Nashville, Tennessee 37235, USA

1991

1. Introduction and Terminology

Let G be a finite undirected graph without loops or multiple lines. A set of lines $M \subseteq E(G)$ is a **matching** if no two share a common endpoint. A matching is said to be **perfect** if it covers all points of G. One can take the point of view that a matching is a special case of a more general concept in graph theory—an ***n-factor***. An **n-factor** of graph G is a spanning subgraph which is regular of degree n; that is, each point has degree n. But we can generalize even more. Let f denote a function which assigns to each point v in $V(G)$ a non-negative integer $f(v)$. Then an **f-factor** is any subgraph F of G in which $\deg_F v = f(v)$. The existence of these increasingly more general concepts for a fixed graph G are all instances of what have come to be called ***degree-constrained subgraph problems.***

The reader of these Proceedings has already seen in Mulder's article [125] that these ideas were of great interest to Julius Petersen and in fact he enjoyed some considerable success in his studies of such problems. Probably the most widely known Petersen results in these areas are two.

(a) In [132], he proved that any connected cubic graph with no more than two cutlines has a perfect matching and hence decomposes into the union of a line disjoint perfect matching and a 2-factor.

It was in this paper that Petersen displayed a cubic graph (actually a multigraph) with three cutlines and no perfect matching, thus showing that in a sense his "2 cutline" theorem was best possible. Incidently, the cubic graph so displayed was attributed by Petersen to his mathematical colleague and friend, Sylvester, and is shown as Figure 4 of Mulder's paper [125].

(b) In [133], Petersen offered the now famous ten-point cubic graph, which has come to be known as the *Petersen graph*, as an example of a cubic graph which cannot be expressed as the union of three line-disjoint perfect matchings.

Of course, we do not want to imply that Petersen's reputation in graph theory rests upon the existence of two particular graphs! These two examples arose out of his ground breaking studies in the area of graph factorization, said studies having been well delineated in the paper of Mulder.

It was but a short time after the appearance of Petersen's work in print, that the young Hungarian mathematician Dénes Kōnig appeared on the scene. It was Kōnig who gave the next strong impetus to the study of graph factorization after Petersen's ground-breaking work, and it is Kōnig with whom we are charged to begin our brief summary of

* work supported by ONR Contract #N00014-85-K-0488

the history of matching theory.

It is a formidable task to undertake such a charge! Indeed, hundreds of papers and a book have been published on the topic of matching theory and there seem to be no signs that the study of matchings is in any immediate danger of "whithering on the vine"! Indeed, throughout this paper, we mention quite a number of problems which remain unsolved as we go to press.

Fortunately, matching theory serves well as an historical thread extending from the time of Kōnig (and before) up to the present, wending its way through graph theory and intersecting many of the most important new ideas which have sprung forth in our discipline. One sees in particular that after the close of World War II this intertwining of matching theory with the study of graphs as a whole became ever more inextricable, even as the study of graph theory as a discipline unto itself literally exploded upon the mathematical scene.

Although it is "jumping the gun" somewhat with respect to the organization of this paper, we can mention three major areas which have joined with graph theory to give rise to many new and deep results. These are: (1) linear programming and polyhedral combinatorics; (2) the linking of graphs and probability theory in the area of *random* graphs and finally (3) the theory of algorithms and computational complexity. (The three areas are far from mutually exclusive; but more about that later.)

But having tried to claim that our task is impossible, let us get to it. Please note, dear reader, that the word "survey" was intentionally avoided in our title and the word "sampler" used instead. We make no claim in this short paper to be comprehensive or complete, but instead readily plead guilty to having selected some of our own favorite branches of matching for more extensive discussion.

Our general plan, then, will be as follows. We present first a Chronology in tabular form of many of the most important events in the history of matching theory. We will then deal with much of this in narrative fashion, hopefully supplying enough references as we go to enable the interested reader to take the various forks in the road offered in order to study more deeply certain of the topics we mention only superficially. As a much more comprehensive guide to matching in general and to a large — but by no means exhaustive — list of references we refer the reader to the book [113]. A less comprehensive — but more up to date — survey of matching theory is the excellent chapter written by Pulleyblank for the forthcoming Handbook of Combinatorics [144].

A CHRONOLOGY OF EVENTS IN MATCHING THEORY

1912	Frobenius' reducible determinant theorem
1914, 1916	Kōnig proves every regular bipartite graph has a perfect matching. Kōnig's Line Coloring Theorem
1915	Kōnig's proves Frobenius' theorem in a graph theoretic setting
1927	Menger's Theorem (almost!)
1931, 1933	Kōnig's Minimax Theorem
1931	Egerváry's Weighted Minimax Theorem
1935	P. Hall's SDR Theorem
1936	Kōnig's book appears
1947	Tutte's 1-factor Theorem
1950	Gallai's new proof of Tutte's 1-factor Theorem; new results on regular factors of regular graphs; extensions to infinite graphs
1952	Tutte's f-factor Theorem
1955	Ore's minimax defect version of Hall's Theorem
1956	Kuhn formulates bipartite matching as a linear programming problem
1956	The Max-flow Min-cut Theorem (Ford and Fulkerson; Dantzig and Fulkerson; Elias, Feinstein and Shannon)
1958	Berge's minimax defect version of Tutte's Theorem
1958	Gallai uses LP duality and total unimodularity to derive the Max-flow Min-cut Theorem, Menger's Theorem, Dilworth's Theorem and Kōnig-Egerváry Theorem
1958	Berge's book appears (in French)
1960	Ore's book appears
1959-60	Kotzig's three "lost Slovak papers" on the structure of graphs with perfect matchings
1959-61	Erdős and Rényi publish first papers on random graphs
1962	Berge's Book appears in English

1961, 1963	Kasteleyn gives a polynomial algorithm for the *number* of perfect matchings in a *planar* graph
1964-65	Gallai and Edmonds obtain canonical decomposition for any graph in terms of its maximum matchings
1965	Edmonds develops first polynomial matching algorithm for *non-bipartite* graphs
1966	Erdős and Rényi discover threshold function for random graph to have a perfect matching
1965-73	Edmonds and Pulleyblank characterize the facets of the matching polytope
1971	Cook finds first "NP-complete" problem
1972	Karp publishes first list of NP-complete graph problems
1972	Lovász, building upon the ideas of Kotzig (1959-60), begins an as yet incomplete extension of the structure of graphs with a perfect matching
1979	Khachian develops first polynomial algorithm for LP – the Ellipsoid Algorithm
1979	*Pippenger introduces* the parallel complexity class "NC"
1979	Valiant proves that computing the *number of* perfect matchings is # P-complete (and also shows that this computation is thus NP-hard)
1982	Naddef and Edmonds, Lovász and Pulleyblank obtain two structural characterizations of the dimension of the perfect matching polytope
1984	Bollobás book on random graphs appears
1985	Razborov obtains his superpolynomial monotone complexity bound for the perfect matching decision problem
1986	Lovász-Plummer book on matching theory appears
1986	Karp, Upfal and Wigderson give RNC (parallel) algorithm for finding a perfect matching
1986-88	Broder, Jerrum and Sinclair relate approximate counting of perfect matchings to random generation of a perfect matching thereby introducing new probabilistic techniques

2. Our Narrative Begins

In 1912, Frobenius [54] published a paper in which he dealt with determinants of square matrices in which the non-zero entries were all distinct variables. The question was: when could such a determinant (i.e., a polynomial) be factored? Frobenius showed that this was possible if and only if one could permute the rows and columns of the matrix so as to display a rectangular block of zero entries. Some three years later, Kōnig [90] gave a somewhat shorter proof, but more importantly from our perspective, showed that this problem could be modeled in terms of the perfect matchings of a certain bipartite graph.

In the following year, Kōnig published twin papers — one in Hungarian [92], the other in German [91] — in which he proved that every regular bipartite graph has a perfect matching. (Actually, this result had been announced some two years earlier in a 1914 communication to the Congrès de Philosophie Mathématique in Paris. However, this communication was not published until some nine years later! See [89].) Others treated this theorem in differing contexts, but to go into this would certainly interrupt the flow of our narrative, so we refer the interested reader to the chapter notes of Kōnig's book in either its original German version [95] or the new English translation [96], a translation for which the graph theory community has had to wait far too long!

In 1927, Menger published the first proof of his now celebrated minimax theorem on connectivity in graphs [120]. The version published in this paper was the "undirected point version" known to any beginning student of graph theory. Informally, the result says that in any undirected graph with two distinguished nonadjacent points s and t, the maximum number of point-disjoint paths joining s and t is equal to the minimum size of any set of points in $G - s - t$ the deletion of which separates points s and t. (There are now four principal versions of Menger' Theorem which one may obtain by taking all combinations of the concepts "undirected graph", "directed graph", "point-disjoint paths" and "line-disjoint paths". All are really equivalent and derivations of each from others can be found throughout the literature. See, for example, the textbook by Bondy and Murty [19].)

Unfortunately, there was a hole in Menger's 1927 proof which was to prove of deep significance for all of subsequent graph theory! First, let us hasten to point out that Menger himself [121] repaired the gap and published a complete proof. But in the meantime, Kōnig had discovered the flaw in the 1927 proof; Menger had neglected the case when the graph involved was *bipartite*.

This realization by Kōnig led to the proof of what is surely one of the most influential theorems in all of graph theory — his *minimax theorem*. (See [93] and [94] for the Hungarian and German language treatments of this landmark result respectively.) The statement of the minimax theorem is easy to grasp. Let G be a bipartite graph. Then the minimum size of any set of points which collectively cover (i.e., touch) each line of G equals the maximum size of any matching in G. Also in 1931, Egerváry [37] published a more general version of the minimax version in which the lines were assigned non-negative weights.

Now one could rhapsodize about minimax theorems and their importance in graph theory at great length! Their importance can hardly be overestimated, especially today, since such ideas and results as the Max-flow Min-cut Theorem, the duality theorem of linear programming, so-called "good characterizations", etc. etc., have emerged.

But *why* are minimax results so important? Because they often tell us when a *candidate for a solution* we have in hand, in fact truly *is a solution*! Consider the simple case of bipartite matching and Kõnig's Minimax Theorem. Suppose we have a matching in hand and wish to know if it is indeed a *maximum* matching.

If, somehow, we can find a point cover for G the cardinality of which is also k, then by Kõnig we know that our matching is indeed maximum. We must be honest here and point out that how one *obtained* such a point cover has been ignored! Indeed, we have already met the crux of the idea of a *good characterization*, a concept generally attributed to Edmonds in the 1960's.

Let us stick to the setting of our paradigm problem — bipartite matching. We say that the matching number k of a bipartite graph is *well-characterized* for the following reason. If we want to convince someone that the matching number is $\geq k$, we need only *exhibit* a matching of cardinality $\geq k$. On the other hand, if we wish to show that the matching number is $\leq k$, we need only exhibit a point cover of size $\leq k$. In the modern terminology of complexity theory, we say that a graph property is in **NP** if, given any graph for which the property holds, there is a "short" proof — or *certificate* — that it holds. (For the definition of NP, it has been agreed upon that "short" means a number of steps polynomial in the size of the input graph.)

We have already been a bit cavalier about just how the problem of bipartite matching is to be posed. Is the problem to obtain a number or is it to be some kind of "yes or no" question? The class NP is normally defined as a class of "yes or no", i.e., *decision* problems. But it is easy to convert the problem of determining the size of a maximum matching into such a decision problem. Let k be any integer such that $1 \leq k \leq |V(G)|/2$. Then *for each such* k, *ask the question:* "is the size of a maximum matching in graph G at least k?" We then need at most k certificates to determine the size of any largest matching in G. But if the task of verifying *one* certificate can be done in time polynomial in the size of the input graph, then, trivially, so can the compound task of verifying k such certificates, for k is, by definition, polynomially bounded in the size of the input graph and the sum of a polynomial number of polynomials is a polynomial!

It is important to emphasize that for the purposes of defining NP, it is of no consequence how we "happened upon" the fact that the property holds, in other words, how we obtained our certificate.

In contrast to the class NP, however, if given any graph for which a property does not hold, there is a certificate showing that it does not hold, we say that the property is in the class **co-NP**. It is immediate by Kõnig's Minimax Theorem that bipartite matching is in NP$\cap$co-NP. (For the cognoscenti of complexity theory, of course one can say more; namely that bipartite matching is in class **P**. That is to say, we have an algorithm, which not only certifies that a given matching is maximum, but actually *obtains* the certificate — in fact, the maximum matching itself! — in polynomial time.)

Surely, class P must a proper subset of class NP! But is it? Most readers will know that this is the outstanding open question in the area of complexity theory today. More particularly, it is clear by definition that we have $\mathrm{P} \subseteq \mathrm{NP} \cap \mathrm{co-NP} \subseteq \mathrm{NP}$. It is unknown whether "$\subseteq$" can be replaced by "$=$" at either location.

We shall return to other complexity results and questions later in this paper.

For much more comprehensive treatments of minimax theorems in graph theory and combinatorics, we refer the reader to the surveys of Woodall [170], Schrijver [152] and [143]. As for us, we shall return to several matching-specific minimax theorems below.

But.....we have strayed from our timeline.

In 1935, Philip Hall [72] published his famous theorem on systems of distinct representatives. Although cast in the language of set theory, it was soon realized by Kõnig [95] that Hall's theorem could be simply stated in terms of bipartite graphs as follows. Let $G = (A, B)$ be a bipartite graph with point bipartition $A \cup B$ with $|A| = |B|$ and if $X \subseteq A$, let $\Gamma(X)$ denote the set of points in B each of which is adjacent to at least one point of X. In other words, let $\Gamma(X)$ denote the set of neighbors of set X. Hall proved that bipartite graph G has a perfect matching if and only if $|\Gamma(X)| \geq |X|$ for every $X \subseteq A$. In this form, Philip Hall's Theorem was to prove one of the most famous in all of matching theory and is perhaps to be found in more lists of references of matching theory papers than any other one paper. It is interesting to note that Hall was quite aware of the contents of Kõnig's 1916 papers and in fact in his paper, Hall's begins by referring to the German version thereof.

The following year, Kõnig published the first book on graph theory [95]. For the first time, graph theory was set down as an organized body of mathematical results derived from a set of axioms in a precise manner. The book was written in German. For all intents and purposes, up to that time the discipline of graph theory had been ignored by the English-speaking mathematical communities in Great Britain and the United States.

But that, along with most other matters in the affairs of man, was about to change swiftly and dramatically.

During the dark days of World War II, little, if any, graph theory was done, as was more or less the case with mathematics in general. However, significant exceptions to that statement are to be found in such areas — for the most part kept highly secret — as cryptography, ballistic trajectories, computer development, nuclear physics and navigation and communications. But the seminal ideas of a new and tremendously important branch of mathematics were also to arise in reaction to very "applied" wartime issues such as deployment and logistical supply. Soon to be born was the discipline of *linear programming*. (See Section 3 below.)

The first significant graph theory theorem of the post-war era was indeed to become one of the most significant in the entire history of the subject. In 1947, W.T. Tutte [158] published his celebrated 1-*factor Theorem*. (This, incidently, before he had finished his Ph.D.!) This result set forth the first characterization of general (i.e., not necessarily bipartite) graphs with perfect matchings.

Let G denote any graph, bipartite or non-bipartite, and let S be any set of points in G. Finally, let $c_o(G - S)$ denote the number of *odd* components of $G - S$. Then Tutte's result states that graph G has a perfect matching if and only if $c_o(G - S) \leq |S|$ for every set $S \subseteq V(G)$. We would point out that similarities in form are apparent between Hall's theorem for bipartite graphs and Tutte's theorem for general graphs. Another interesting parallel lies in the proofs. In Hall's formulation, let us call a set $S \subseteq A$ a *barrier* if $|S| > |\Gamma(S)|$. Similarly, in Tutte's formulation, let a *barrier* be any set S for which $|S| < c_o(G - S)$. Then in each of the two theorems, half of the proof is trivial. Namely, if

there exists a barrier, then a perfect matching cannot exist. The non-trivial part of each theorem lies in proving the converse.

One more remark is in order here before pressing on. Tutte's original proof of his 1-factor Theorem involved computations with the so-called *Pfaffian* of a matrix. It was not long before other proofs were found which involved only truly graph-theoretic techniques. However, the notion of a Pfaffian matrix associated with a graph was to reappear much later in the work of Kasteleyn and is still being investigated at the time of this writing. (See for example, Vazirani [88; 89].)

In 1950, one of the truly "unsung heros" of graph theory, T. Gallai, published a paper [60] in which, among other things, he gave a proof of Tutte's theorem using the method of *alternating paths*. The idea of using alternating paths, to be sure, did not originate with Gallai. In fact, Gallai himself in his two fascinating biographies of Kōnig [63; 96] points out that such methods were used by Kōnig as early as 1915 [90] and indeed, Mulder [125] has pointed out that the idea was even used by Petersen [132]! The concept of an alternating path is simple indeed. Suppose one has a fixed matching M in a graph G. Let us agree to call the lines of this matching *red* if they belong to the matching and *blue* otherwise. Then a path P in G is said to be **alternating with respect to** M (or simply M-**alternating**) if the lines of P alternate in color. An M-alternating path P is said to be M-**augmenting** if it is M-alternating and begins and ends with blue lines. Clearly, if one can find an M-augmenting path in the graph G which joins two lines not covered by M, then one can find a matching larger than M simply by exchanging on the lines of P.

It was proved by Berge [5; 9], but probably known to Petersen (see Mulder [125]), that a matching M is of maximum cardinality, or simply *maximum*, if and only if there exists no augmenting path with respect to M. This fundamental idea has proved to be the basis for the best known and most efficient combinatorial algorithms for finding maximum matchings known today. But more about that below.

In addition to his new proof of Tutte's Theorem, Gallai also extended the theory of regular factors of regular graphs first started by Petersen. (See Section 1 of the present paper.) Indeed, gradual improvements on the general question of the existence of regular factors in regular graphs had been made by Bäbler [3], Rado [147], Belck [4] and others. Although this is an interesting and important branch of graph factorization, we have chosen not to treat it in detail. Instead, fortunately, we can refer the interested reader to the paper of Bollobás, Saito and Wormald [17] for a concise summary of the status of the problem, leading up to their own result which is, to the best of the author's knowledge, the latest word on this subject in the following sense. Given integers $r \geq 3$, and $1 \leq \lambda \leq r$, the authors determine precisely for which values of k, every r-regular graph G with line-connectivity λ has a k-factor.

The reader, we hope, will forgive a personal remark at this point. We referred to Gallai as an "unsung hero" above. Indeed, his name is seldom found, for example, in introductory textbooks on graph theory. His publication list, although containing some graph theoretic results of the highest caliber, is not long. But to those fortunate enough to have come to know him, either as students or colleagues (or both), have the greatest respect for this modest, unassuming and selfless man. (See Lovász's remarks in his tribute to Gallai on the occasion of his seventieth birthday [110].) The present author fondly recalls some twenty

years ago or so when he wrote a letter to Gallai asking a question about independent sets. In response, we received a long and carefully written reply containing theorems, constructions, etc. etc. which to the best of my knowledge, still have not seen the light of day! Nor was this the only occasion upon which we benefited from Gallai's kindness. In conversations with other graph theorists, we have heard similar stories. One can only conjecture what graphical gems lay hidden on Gallai's desk which he has not seen fit to publish. Yet, in a life filled with more than one's fair share of personal tragedy, Gallai seems to have remained unfailingly kind, unselfish and considerate to his colleagues and students.

At this point, the reader's attention is directed back to the Introduction to this paper where the concept of an *f-factor* was defined. Of course, a perfect matching is just a special case of an f-factor when the function f has the value 1 on each point of the graph. In 1952, Tutte [159] published his *f-factor Theorem* in which he gave a characterization of those graphs which have an f-factor. Unfortunately, the characterization is somewhat complicated and not easy to apply. Tutte also formulated a beautifully symmetric version of his f-factor theorem by defining what he called an *f-barrier* and then showing that an arbitrary graph had an f-factor if and only if it had no f-barrier. This approach is much in the spirit of the classical theorems of P. Hall and Tutte's own on the existence of a 1-factor.

Soon after in 1954, Tutte showed that in fact his f-factor Theorem could be derived from his earlier 1-factor Theorem [160]. Although this served the double purpose of unifying his 1-factor Theorem and his f-factor Theorem and provided a much more accessible *proof of the latter, it still did not help one to use the* f-factor result. Indeed, Tutte himself was quite aware of this difficulty and some twenty years later published a paper in which he attempted to simplify matters by the introduction of *maximal* barriers [161]. Even so, it seems fair to say that despite these attempts, the f-factor theorem remains one of the most challenging results for graph theorists to assimilate and to use in their own work.

The interested reader will find the most comprehensive treatments of the f-factor Theorem in the three books [162], [10] and [113].

But the story about f-factors did not end here. All of the above-mentioned treatments of the problem deal with the *existence* of an f-factor. It is another matter indeed to actually *find* one! Of course this leads us immediately to the area of algorithms. However, since the f-factor Theorem will not be a central issue in this Sampler, we omit an algorithmic discussion from Section 6. Suffice it to say that there exist algorithms for finding f-factors which are polynomial only in the number of points and lines of the graph and are independent of the function f. (See Gabow [57] and Anstee [2].)

Having said only that, we revert to our Chronology.

In 1955, Ore [127] published his minimax version of Hall's Theorem and thereby focused attention on a more general problem: in a graph with no perfect matching, what is the size of a largest matching? We call such a matching a **maximum** matching.

It is a good idea at this point to clear up a point which is not perhaps so minor as one might think. All mathematicians are aware of the difference between the concepts of a *maximal* structure (i.e., inclusion-wise maximal) and a *maximum* structure (i.e., a structure having largest cardinality). A maximum matching, for example, is certainly maximal,

but the converse implication seldom holds. Indeed, in mathematics in general, the two concepts are seldom equivalent and yet the two words are carelessly interchanged again and again in the literature by mathematicians who should know better. Just how different the two concepts are vis-à-vis matchings will be made more apparent when we discuss algorithmic questions in Section 6. (Incidently, although it is not at all clear in which graphs all maxim*al* matchings are indeed maxim*um* (in other words, those graphs in which all maximal matchings have the same cardinality), such graphs have been characterized in such a way that they can be polynomially recognized. (See [105].)

But having gotten that off our chest, let us return to Ore's result. Let us henceforth denote the size of a maximum matching in graph G by $\nu(G)$. Now let $G = (A, B)$ denote an arbitrary bipartite graph. Define now the parameter $\nu'(G)$ as the minimum taken over all subsets $X \subseteq A$ of the quantity $|A| + |\Gamma(X)| - |X|$. Then Ore's minimax theorem — also called *Ore's Deficiency Theorem* — says that $\nu(G) = \nu'(G)$.

Two years later, Berge [6] generalized this result to all graphs. Let us modify Ore's parameter $\nu'(G)$ as follows. Let G be any graph; that is, no longer necessarily bipartite. As in the statement of Tutte's 1-factor Theorem above, for any subset $S \subseteq V(G)$, let $c_o(G - S)$ denote the number of odd components of $G - S$. Now let $\nu''(G)$ be one half the minimum over all subsets $S \subseteq V(G)$ of the quantity $|V(G)| - c_o(G - S) + |S|$. Then Berge's minimax theorem for matchings — also called *Berge's Deficiency Theorem* — says that $\nu(G) = \nu''(G)$.

It is of interest to note here that the Theorem of P. Hall, Tutte's 1-factor Theorem and their respective deficiency versions discussed above belong to a category of theorems called "self-refining results". The idea is this. Although the deficiency versions sound more general than the two 1-factor theorems to which they correspond (and to be sure the 1-factor versions *are indeed* immediate corollaries of the deficiency versions), it is somewhat surprising to discover that in fact the deficiency versions are really *equivalent* to the 1-factor results

This business of *equivalent* theorems which we have already mentioned in our discussion of Menger' Theorems in their various forms, and which we meet here again in these two deficiency results, extends even further in the area of our narrative. Although it is not central to our chronology, it can be shown that most of the main theorems we state are all equivalent! That includes such results as Kōnig's minimax theorem, Menger's Theorem, Hall's Theorem, Tutte's 1-factor Theorem, Tutte's f-factor theorem, the deficiency theorems of Ore and Berge, Dilworth's Theorem on partially ordered sets [31], and the Max-flow Min-cut Theorem which we are about to meet. Those we relish "circles of proofs" are referred to [113], to the thesis of Magagnosc [119], to Hoffman [73] and to a monograph devoted entirely to this subject [150].

At this point we will diverge from our heretofore "linear" treatment of the Chronology of matching theory to treat in somewhat more detail four "branches" of the subject.

3. The Max-flow Min-cut Theorem and Linear Programming

In the immediate post-World War II years, George Dantzig, who had, by his own admission, "become an expert on programming planning methods using desk calculators" [29; pg 78], was still in the employ of the U.S. Air Force. In 1947, based upon a family of logistical supply programs with which he had worked, Dantzig formulated the archetype of what we call today a *linear programming problem*. The idea was — and is — to optimize (i.e., to either maximize or minimize) a linear objective function subject to a set of of *linear constraints* (linear equations and inequalities). Moreover, Dantzig developed a method for efficiently solving such problems; a method which remains the favorite in practice today — *the Simplex Method*. Later in the same year, Dantzig met John von Neumann for the first time in order to consult with him on his new idea. It seems that, although no evidence existed in print, the seminal ideas of linear programming had also occurred to von Neumann, apparently during his work with Morgenstern on the theory of games. According to Dantzig, it was at this time that he (Dantzig) first learned about the now-fundamental concept of *duality* from von Neumann. The history seems muddy here, but Dantzig claims that as far as he is concerned, the famous *Duality Theorem* was known to von Neumann, although it was Gale, Kuhn and Tucker [58] who published the first rigorous proof.

The idea of the Duality Theorem is that for every linear program which, let us say, seeks the maximum of a certain linear objective function subject to a collection of linear constraints, there is a second linear program — the *dual* — definable in terms of the parameters of the first program — called the *primal* — such that the dual program is a minimization problem and that as long as both the primal and dual programs have solutions, they are in fact equal. Or stated more formally,

Theorem 2.1. (*The Duality Theorem of Linear Programming*) Let A be any matrix and b and c be vectors. Then

$$\max \{cx | Ax \leq b\} = \min \{yb | y \geq 0;\ yA = c\}$$

(provided at least one of these sets is non-empty.)

In this succinct representation, cx represents the *(primal) objective function* and yb the *dual objective function*. The linear constraints of the primal are represented as the matrix inequality $Ax \leq b$, while those of the dual are stored in the expressions $y \geq 0$ and $yA = c$.

So here again we have a minimax theorem.

From the barely countable number of references on linear programming (or "LP" for short) which exist in book and paper form, in addition to the historical article by Dantzig [29] already referred to, we suggest the books by Chvátal [23] and Schrijver [154]. For our part, we shall attempt to ruthlessly stick only to those aspects of LP which directly affect matching theory. (See also [113; Chapt. 7].)

But before continuing any discussion of linear programming and its applications to matching, let us introduce one more minimax theorem which will turn out to be important for our purposes. This theorem sounds strikingly like the Menger Theorem(s) discussed above and first proved in the 1927-32 era. But, similar though it is, our next theorem

remained undiscovered for another twenty years until proved first by Ford and Fulkerson in 1956 [47] not long after the birth of LP described above. It is commonly called the *Max-flow Min-cut Theorem.*

First we need a bit of terminology. Let D be a digraph with two distinguished points s and t called the *source* and *sink* respectively. As usual, we will denote the line directed from point u to point v by (u,v). In addition, let each line (u,v) be assigned a non-negative real number $c(u,v)$ called its *capacity*. The resulting line-weighted digraph is often called a *transportation network*, or simply, a *network*. Let $V(D) = S \cup \overline{S}$ denote a partition of the point set $V(D)$ such that the source s lies in S and the sink t in $\overline{S}$. Then the ordered pair $(S,\overline{S})$ (or sometimes the set of lines directed from set S to set $\overline{S}$) is called an $s-t$-**cut** in D. The sum $\sum_{(u,v),u\in S,\ v\in\overline{S}} c(u,v)$ is called the **capacity** of the cut $(S,\overline{S})$ and is denoted by $c(S,\overline{S})$.

Next, let ϕ denote another function from the lines of D to the real numbers satisfying the following two conditions:

(i) for each line (u,v), $\phi(u,v) \leq c(u,v)$, and

(ii) for each point $u \in V(D)$, $u \notin \{s,t\}$, $\sum_{(w,u)\in E(D)} \phi(w,u) = \sum_{(u,v)\in E(D)} \phi(u,v)$.

Such a function ϕ is called a **flow** in digraph D. We call the quantity $\sum_{(s,u)\in V(D)} \phi(s,u)$ the **value** of the flow ϕ and denote it by $|\phi|$.

We are now prepared to state the next result.

Theorem 2.2. (*The Max-flow Min-cut Theorem*) Let D be a network with source s and sink t. Then: $\max |\phi| = \min c(S,\overline{S})$, where the maximum is taken over all $s-t$ flows ϕ and the minimum is taken over all $s-t$ cuts $(S,\overline{S})$.

This extremely useful theorem sounds like a direct generalization of Menger's Theorem (directed line version) and in fact it is! But it is also yet another example of a self-refining result in that it can be derived from Menger's result and hence by the above discussion can be added to the circle of proofs already containing Kõnig's Theorem, Hall's Theorem, Tutte's Theorem, etc. The first proof of the Max-flow Min-cut Theorem is due to to Ford and Fulkerson [47], followed hot on the heels by independent proofs by Elias, Feinstein and Shannon [40], a second due to Ford and Fulkerson [48] (this one containing the now-familiar labeling algorithm for constructing a maximum flow) and yet another due to Dantzig and Fulkerson [30] using the duality theorem of LP. Thus flow theory and linear programming were essentially "joined from birth"! (Hoffman [74] claims that Kotzig had independently proved the theorem during World War II as well. Indeed, at least a line-version of Menger's Theorem appears in the Slovakian reference [97] which, because of its inaccessibility due to language among other things, has remained largely ignored. This unfortunate linguistic problem was to also rob Kotzig of credit well-deserved in the theory of graphs with perfect matchings, but more about that in Section 4 below.) One finds a nice historical treatment of the early days of the Max-flow Min-cut Theorem in the book by Ford and Fulkerson [49] including a derivation of the Kõnig Minimax Theorem from Max-flow Min-cut.

Kuhn [102; 103] at nearly the same time as the above, published an algorithm for the *Assignment Problem* (line-weighted bipartite matching) which makes use of the primal-dual approach of LP. Kuhn seems to have been the first to refer to this procedure as the

Hungarian Method and pays fulsome tribute to Kõnig and Egerváry for having developed the essence of the method in their classic papers of some twenty-five years before. Henceforth, matchings, flows and linear programming were to be inextricably bound together.

We now describe what might be called — for want of a better term — the *polyhedral approach* to matching.

Let us agree that our task is to find a *maximum* matching, using LP. In order to do this, consider each matching M in a given graph G as a binary vector of length $|E(G)| = m$, indexed by the lines of G, in which one finds a 1 in the ith slot if and only if the ith line is found in matching M. Such a binary vector will be called a **matching vector**. One can then define the **matching polytope** of a graph G, denoted by $M(G)$, as the convex hull of all these matching vectors.

But the usual approach to solving a linear programming problem — as we have already seen in the brief description of the duality theorem above — is to is to optimize (in this case maximize) a linear function subject to a set of linear constraints. Let us set about formulating the problem this way. We shall begin by stating an certain abstract linear program:

$$\text{maximize} \quad \mathbf{1} \cdot \mathbf{x} \tag{3.1}$$

$$\text{subject to} \quad \mathbf{x} \geq \mathbf{0} \tag{3.2}$$

$$A\mathbf{x} \leq \mathbf{1}, \tag{3.3}$$

where A is a matrix of (non-negative) real numbers.

Now let us begin to specialize. Let $A = (a_{ve})$ denote the **point-line incidence matrix** of graph G; that is,

$$a_{ve} = \begin{cases} 1, & \text{if } v \text{ is an endpoint of line } e, \\ 0, & \text{otherwise.} \end{cases}$$

So now A has become an integer — in fact, (0,1) — matrix.

The solutions of the above linear program will be called **fractional** matchings; the reason why will become clear in just a moment.

Now let us suppose that among the vectors $\mathbf{x}$ yielding an optimum solution to the above LP, one has as its components only 0's and 1's. Then the constraints (3.2) and (3.3) guarantee that vector $\mathbf{x}$ is a *matching* and hence $\mathbf{1} \cdot \mathbf{x}$ is just the number of lines in this matching. Thus our maximized objective function is just *the cardinality of a maximum matching*!

Moreover, in the special case when graph G is bipartite, it can be proved that such a $0-1$ solution vector $\mathbf{x}$ always exists.

But let us return to the inequalities (3.2) and (3.3) for a moment. It may be shown that the solutions to these inequalities form a polytope (or bounded polyhedron), called the **fractional matching polytope**. (In fact, a system of inequalities — or *half spaces* — like those above is one of *two* equivalent methods of defining a polyhedron; the other, via a classical result of Minkowski [124] and Weyl [169] is as the convex hull of a finite number

of vectors.) It is a well-known fact from LP that at least one optimum value of $\mathbf{x}$ will occur at a vertex of this polytope. But when G is bipartite, *all* such vertices are integral! This follows from the fact that when G is bipartite, every square submatrix of the constraint matrix A has determinant 0 or ± 1 [113]. A matrix possessing this property for each of its square determinants is called **totally unimodular**. The result we seek then follows from a more general result due to Hoffman and Kruskal [75] which in our setting says that if the constraint matrix is totally unimodular, then the corresponding polyhedron has all integral vertices.

But let us now drop the assumption that our graph is bipartite and return to the LP having constraints (3.2) and (3.3). By linear programming duality, we have a minimax result which in turn offers a "good chracterization", but a good characterization of what? The answer is: a good characterization of the cardinality of a maximum *fractional* matching. But, sadly, we can no longer necessarily conclude that among the vectors $\mathbf{x}$ which maximize the objective function, i.e, that correspond to maximum (fractional) matchings, there is *any at all* which is integral! So what are we to do?

Let us once again return to our paradigm LP above with constraints (3.2) and (3.3). To be sure, there are only a polynomial number of constraints given — namely $|VG| + 1$. Let us simply replace the constraint (3.2) with the stronger demand that:

$$\mathbf{x} \text{ is a } \mathbf{0-1} \text{ vector.} \qquad (3.2)'$$

The resulting new LP has exactly the same number of constraints, but duality theory no longer applies, so our minimax result is gone. Where can we turn?

Fortunately, there is an alternative approach. It can be shown that the integrality constraint is dropped, an *integral* minimax result is still obtainable, by *adding more linear constraints*!

But what kind of constraints can we add to accomplish this and how many of them will do the job? To this end, let us define the **matching polytope** of graph G, denoted $M(G)$, as the convex hull of all matchings in G. We know that such a polytope has an alternative definition in terms of a system of linear inequalities. We now seek to find such a system. Of course, it is natural to want to add as few additional constraints as possible in order to accomplish this task as well.

Such an alternative description of the matching polytope in terms of constraints (3.2) and (3.3) and a set of additional constraints has been accomplished by Edmonds [34].

From this point on, details become quite a bit more difficult to deal with and we refer the reader to [113] or [142; 143] as just three possible sources. First, let us define the **co-boundary** of a point v of graph G, denoted $\delta(v)$, as the set of lines incident with point v (or more carefully, in our LP language, the characteristic $|E(G)|$-vector of such a line set). Then we will replace each of the $|V(G|$ constraint inequalities in (3.3) with the inequality:

$$\mathbf{x} \cdot \delta(v) \leq 1, \text{ for each point } v \in V(G). \qquad (3.3)'$$

Now let us define an entirely new set of inequalities called the *blossom inequalities*:

$$\mathbf{x} \cdot E(S) \leq \frac{1}{2}(|S| - 1), \text{ for all } S \subseteq V(G), \text{ such that } |S| \geq 3 \text{ and odd.} \qquad (3.4)$$

Edmonds [34] proved that the matching polytope $M(G)$ is equal to the set of vectors $\mathbf{x}$ satisfying (3.2), $(3,3)'$ and (3.4).

In the case of the matching polytope, let us call a linear inequality **essential** if it is valid and is not a non-negative combination of other valid inequalities. The set of points which satisfy an essential inequality with equality is called a **facet**. (We may say equivalently that a valid inequality "$*$" is a facet if and only if every valid inequality which holds with equality whenever $*$ does, is a positive multiple of $*$.) Edmonds showed that the facets of the matching polytope are found among (3.2), $(3.3)'$ and (3.4). Somewhat later, Pulleyblank [141] in his Ph.D. thesis (see also Pulleyblank and Edmonds [145]) showed precisely which of these facets are necessary for a *unique minimal* linear system sufficient to define $M(G)$.

The situation is even more complex when one passes from the matching polytope $M(G)$ to the **perfect matching polytope**, $PM(G)$, defined as the convex hull of perfect matchings of graph G. Clearly this polytope lies inside the matching polytope in general. In fact, a linear description for $PM(G)$ can be obtained from that given above for $PM(G)$ by appending one more linear constraint, namely: $\mathbf{1} \cdot \mathbf{x} = |V(G)|/2$. However, one runs into trouble when seeking a *unique minimal* description of this polytope! (See [113].)

But let us stop here and summarize a bit. We find that one can obtain *integral* minimax theorems for both maximum matching and perfect matching. But the catch is that in order to find descriptions of the corresponding polytopes, one is saddled with an *exponential number of facets*! This puts a crimp in our style if we want to design a *polynomial algorithm* for the maximum matching or perfect matching problems *based on linear programming*! In fact, a polynomial LP algorithm was unknown until the development of the so-called Ellipsoid Algorithm due to Khachian [88]. Even then, one requires a polynomial time subroutine for testing certain constraints. Fortunately, such a subroutine algorithm has been developed. (See [129] and also [70]. See also Remark 1 at the end of this paper.)

But it remains an open question to decide which graphs G have the property that their perfect matching polytope $PM(G)$ possesses only a polynomially bounded number of facets. Gamble [64] has constructed a family of planar graphs G for which $PM(G)$ have exponentially many facets. On the other hand, he has shown that a rather widely studied family of planar graphs called *Halin* graphs always do have a polynomial number of facets. Hence the perfect matching problem can be solved on Halin graphs without further ado simply by applying any polynomial LP algorithm. At this time, that would be either the Ellipsoid method of Khachian or the more recent algorithm of Karmarkar [80].

In ending this section, we hasten to point out that there are in fact polynomial algorithms for maximum (and therefore also perfect) matching which are not at all LP in nature! The most famous of these is undoubtedly *Edmonds Blossom Algorithm*. (See [33] and also [113].) We shall return to algorithmic questions in the final section of this paper.

Returning briefly to our Chronology, it should be noted that 1958 saw the appearance of Berge's first book on graph theory [7]. This was the first book dealing with the discipline to have appeared since that of Kōnig some twenty-two years before. The Berge book was published first in French, but two years later, the year 1960 saw, for the first time, a graph

theory book in English — Ore's AMS Colloquium volume [128]. Then in 1962, Berge's volume appeared in English translation [8]. With the appearance of these two books, graph theory began to be much more widely studied by students outside continental Europe.

4. On the Number of Perfect Matchings

Suppose a graph G has at least one perfect matching. *Exactly how many* does it have? Is there a polynomial algorithm to count them?

Such questions have application in the real world. For example, this question arises in a problem in crystal physics — counting the number of dimers on a rectangular lattice. We shall return to this application below.

But first, some bad news. In 1979, Valiant [163] proved that counting the number of perfect matchings in a graph, even if it is bipartite, is #P-complete. He also showed that any problem in the class #P is at least as hard as any NP-hard problem. Hence it is highly unlikely that a polynomial algorithm exists for this *exact* counting problem.

So what can be done? We will explore three main avenues of research in this regard. Let us denote the number of perfect matchings in any graph G by $\Phi(G)$.

The first direction of investigation has been to determine *interesting subclasses* of graphs for which $\Phi(G)$ can be *polynomially* determined. The second area of research deals with *bounding* $\Phi(G)$. Finally, the third deals with finding *efficient approximations* for $\Phi(G)$. We will deal with the first two areas in this section and defer treatment of the third to Section 6 on algorithms.

Motivated by the dimer counting problem mentioned above, Kasteleyn [86; 87] developed an algorithm for determining $\Phi(G)$ for *planar* graphs. He discovered that one can always orient the lines of an (undirected) planar graph in a certain way, so that this orientation can then be used to polynomially determine the number of perfect matchings. For a rather complete treatment of Kasteleyn's method, we refer the reader to [113]. For both Kasteleyn's method and more recent extensions thereof, here we will follow the excellent summary of Vazirani and Yannakakis [166; 167].

Our discussion now reverts back to general (i.e., not necessarily planar) graphs. If G is any graph and C a cycle in G, call C **good** if it has even length and $G - V(C)$ has a perfect matching. Orient G arbitrarily to obtain a directed graph $\overrightarrow{G}$. An even cycle is said to be **oddly oriented** (with respect to the given orientation) if when one traverses the cycle (in either direction) an odd number of lines are traversed in the direction of the orientation. Orientation $\overrightarrow{G}$ of graph G is then called a **Pfaffian orientation** if every good cycle in G is oddly oriented.

Now how does such an orientation help one to determine $\Phi(G)$? Let $\overrightarrow{G}$ be a Pfaffian orientation of graph G and let $A(G)$ denote the symmetric $n \times n$ adjacency matrix of G. Modify $A(G)$ to obtain a second matrix $A_s(\overrightarrow{G})$, the so-called **skew adjacency matrix** of $\overrightarrow{G}$ as follows:

$$A_s(\overrightarrow{G})_{i,j} = \begin{cases} +1, & \text{if } (v_i, v_j) \in E(\overrightarrow{G}), \\ -1, & \text{if } (v_j, v_i) \in E(\overrightarrow{G}), \\ 0, & \text{otherwise.} \end{cases}$$

It can then be shown that since the orientation is Pfaffian,

$$\Phi(G) = \sqrt{\det\,(A_s(\overrightarrow{G}))}.$$

Since evaluation of a determinant is well-known to be polynomial, we have our polynomial scheme for computing $\Phi(G)$.

The problem, of course, lies in finding a Pfaffian orientation. As we mentioned above, Kasteleyn gave a polynomial procedure for finding one in any *planar* graph. Let us call a graph $K_{3,3}$-**free** if it contains no subdivision of the complete bipartite graph $K_{3,3}$. By the classical theorem of Kuratowski, we know that all planar graphs are $K_{3,3}$-free. Little [107] extended Kasteleyn's result for planar graphs by showing that every $K_{3,3}$-free graph has a Pfaffian orientation. His proof implicitly also gives a polynomial procedure for obtaining such an orientation. More recently, V.V. Vazirani has in turn extended Little's result by showning that, in fact, there is even an NC algorithm for obtaining a Pfaffian orientation in a $K_{3,3}$-free graph. (For more about NC algorithms, see Section 6 below.)

Let us return once more to Pfaffians and graphs in general. In fact, one can pose three (at least formally) different questions here:

(1) Does a given graph G have a Pfaffian orientation?

(2) Given an orientation $\overrightarrow{G}$, is it Pfaffian?

(3) Given a graph G, find a Pfaffian orientation $\overrightarrow{G}$.

Observe that the first two questions are *decision* questions, while the third is a *search* question. The complexity of all three questions is presently unknown. It has been shown [113] that problem (2) is in co-NP. (But that is certainly not obvious from the presentation given here!) Moreover, recently it has been shown [166; 167] that problems (1) and (2) are polynomial-time equivalent. Hence problem (1) is also in co-NP.

In the case when the graph G is *bipartite*, there is an interesting connection between problems (1) and (2) and a fourth problem which has been studied by Seymour and Thomassen [155; 157], among others.

(4) Given a directed graph $\overrightarrow{G}$, does it contain a directed cycle of even length?

Vazirani and Yannakakis [166; 167 have proved that problem (4) is polynomial-time equivalent to problems (1) and (2).

We will now turn our attention to the second area of investigation mentioned at the beginning of this section — *bounding* the number of perfect matchings.

In the special case of *regular bipartite* graphs, we have some interesting bounds — both upper and lower — on $\Phi(G)$. Suppose G is a k-regular bipartite graph on $2n$ points. We state the bounds in the following combined form:

$$n!(k/n)^n \leq \Phi(G) \leq (k!)^{n/k}.$$

The lower bound follows from the famous "conjecture" of van der Waerden [168] about doubly stochastic matrices which was proved nearly simultaneously, but independently, by Falikman [46] and Egoryčev [38; 39]. The upper bound follows from another more general permanental inequality conjectured by Minc [123] and proved by Brègman [21].

For a brief discussion of each of these inequalities, see [113] and for a much more thorough one, see Schrijver [153].

The balance of this section will report on results to date on lower bounds for $\Phi(G)$ for general graphs. In connection with this, we should mention that over the years, several methods for decomposing graphs in terms of their maximum, or perfect, matchings have been developed. The first of these due independently to Edmonds [33] and Gallai [61; 62] described a decomposition in terms of the maximum matchings. The procedure can be executed in polynomial time; that is a corollary of Edmonds' Blossom Algorithm to be discussed in more detail in Section 6. The problem with the Gallai-Edmonds decomposition, if one may say so, is that gives no information in the case when the graph in question has a perfect matching. But others were already at work on this case. Indeed Kotzig [98; 99; 100] had already begun work on a decomposition of graphs with perfect matchings; unfortunately his work remained largely unknown due to the fact that the results appeared in Slovak. In [108], Lovász, then working on his Ph.D., extended the work of Kotzig in this area. Subsequently, Lovász and Plummer [112] and Edmonds, Lovász and Pulleyblank [36] further developed this decomposition. It had now come to be called the ***brick decomposition theory*** for graphs with perfect matchings. A somewhat simpler approach leading to the same terminal decomposition was begun in [36] and developed more fully in [111] now bears the name ***tight set decomposition theory.*** We will now proceed to describe how to obtain lower bounds on $\Phi(G)$ via the tight set decomposition approach.

Unless otherwise specified, let us assume that the graph G in question has an even number of points. Also, since we seek a *lower* bound on $\Phi(G)$, we will lose nothing by assuming that our graph G has the property that every line lies in a perfect matching. Such graphs are called **1-extendable** or **matching covered.**

Let us begin by recalling the description of the matching polytope $M(G)$ in Section 3. In particular, we saw that one could obtain a description of the *perfect* matching polytope $PM(G)$ from constraints (3.2), (3.3)', (3.4) and one additonal linear constraint, namely, $\mathbf{1} \cdot \mathbf{x} = |V(G)|/2$. It turns out that there is alternate description of this polytope which lends transition to our discussion in the rest of this section.

A cutset of lines C in a graph G is called an **odd cut** if C is a line cutset separating two point sets of odd cardinality. The two odd sets separated by cut C are called the **shores** of the cut. An odd cut is said to be **trivial** if one of its shores is a singleton. It can be shown (see Edmonds [34]) that the perfect matching polytope can be described by the following constraints involving odd cuts:

$$\mathbf{x} \geq \mathbf{0} \tag{4.1}$$

$$\mathbf{x} \cdot \mathbf{C} = 1 \qquad \text{for } C \text{ a trivial odd cut} \tag{4.2}$$

$$\mathbf{x} \cdot \mathbf{C} \geq 1 \qquad \text{for } C \text{ a non} - \text{trivial odd cut} \tag{4.3}$$

We still need several more definitions. A cut is **tight** if every perfect matching contains exactly one line of the cut. Clearly, then, every tight cut is odd and trivial tight cuts are just the stars at each point of G.

Now let graph G be called **bicritical** if for every choice of distinct points u and v in G, graph $G - u - v$ has a perfect matching. A 3-connected bicritical graph is called a **brick**. Bricks form one of two basic classes of "building blocks" which we are about to develop for a canonical decomposition procedure for every 1-extendable graph.

Our second class of building blocks are the so-called *braces* of G. A bipartite graph $G = A \cup B$ is a **brace** if for every subset $X \subset A$ with $0 < |X| < |A| - 1$, $|\Gamma(X)| \geq |X| + 2$.

Note that by definition, no bicritical graph can be bipartite, and hence bricks and braces form disjoint classes of 1-extendable graphs. The close connection with tight cuts lies in the fact that a 1-extendable graph is a brick or a brace if and only if it has no non-trivial tight cuts. This result is far from trivial. The left-to-right proof is to be found in a paper of Edmonds, Lovász and Pulleyblank [36] and, in fact, forms one of the key results of that paper. It depends heavily on the polyhedral approach and the details will be omitted here. The converse is proved in [111].

Now suppose that C is any non-trivial tight cut in a 1-extendable graph G. If the two shores relative to cut C are S_1 and S_2, then denote by G_1 and G_2 the two (strictly smaller) graphs obtained from G by contracting S_1 and S_2 to single points respectively. It can be proved that both G_i's are 1-extendable. Hence this procedure can be repeated. We are only forced to stop when one of the contracted graphs we produce is either a brick or a brace. Keep a list of the bricks and braces so formed. Lovász calls this procedure a *tight cut decomposition procedure* and the list of bricks and braces obtained, the *result* of the procedure.

The reader will recognize immediately that in general one has quite a bit of freedom in this procedure, depending upon the sequence of tight cuts chosen. The truly amazing thing about this routine is that the final list of bricks and braces is *independent* of the sequence of tight cuts chosen! This deep result was really proven in two stages. In [36], the *number* of such terminal graphs was shown to be invariant. Then five years later, Lovász [111] proved that actually the terminal graphs themselves were invariant.

Where is all this leading, you may well ask? Let $r(G)$ denote the maximum number of perfect matchings (perfect matching vectors) in G which are linearly independent over $\Re$, the real numbers. Then it can be shown [36] that, if G is any 1-extendable graph with n points, m lines and $\beta(G)$ bricks in its final list resulting from a tight set decomposition procedure, then:

$$r(G) = m - n + 2 - \beta(G).$$

Since $\Phi(G)$ is at least as great as $r(G)$, we have the lower bound we seek. In the special case when the graph G we start with is itself a brick, we obtain the bound $\Phi(G) \geq m - n + 1$. This result can then be used to obtain another bound for an arbitrary bicritical graph. In [112] it was proved that if G is bicritical, then $\Phi(G) \geq n/4 + 2$ and conjectured that, in fact $\Phi(G) \geq n/2 + 1$ for these graphs. This conjectured bound can now be derived from the rank equation for a brick and an inductive argument.

The procedure for computing $r(G)$ can be done in polynomial time since the tight set decomposition procedure can be so carried out. (This too is far from obvious, based upon our sketchy treatment of the topic.)

Even more evidence that the study of bricks and braces may well be the secret to counting perfect matchings has been provided very recently by V.V. Vazirani and Yannakakis [166; 167] who have shown that for an arbitrary (1-extendable) graph G, G is Pfaffian if and only if all its bricks and braces are Pfaffian.

It is probably not surprising to learn that the complexity of the Pfaffian orientation problem for bricks remains open.

Another approach to learning more about bricks has been pursued by the author. Motivated by the concept of a 1-extendable graph, we make the following definition. Let k be an integer such that $0 < k < n/2$. A graph G (with a perfect matching) is **k-extendable** if every set of k independent lines extends (i.e., is a subset of) a perfect matching. In [135] it was proved that if G is 2-extendable and not bipartite, then it is bicritical. It was also shown that every k-extendable graph is also $(k-1)$-extendable as well as $(k+1)$-connected. It then follows that every 2-extendable graph is either a brick or a brace. In addition, it also follows immediately that as k increases, the k-extendable graphs form a nested sequence of graph families.

We have studied the interaction of k-extendability with a number of other graph parameters, for example, toughness [138] and genus [137; 136], to name but two. A survey of results in this area can be found in [139].

In bringing this section on bounds to a close, we would like to mention some lower bound results which can be obtained by more elementary means than those of the tight set decomposition. On the other hand, they do follow, for the most part, from the tight set approach.

Let us now agree to seek a lower bound for $\Phi(G)$ depending upon the *connectivity* of G. Again, as above, we will assume that G has a perfect matching.

One of the early results in this direction was obtained by Lovász [108]. He proved that if a graph G with a perfect matching is k-connected and not bicritical, then $\Phi(G) \geq k!$. At the time, the result piqued the fancy of several investigators, including your author, who found it rather counterintuitive that the bicritical graphs should be those with "few" perfect matchings. After all, trivially, bicritical graphs are 1-extendable. But, we digress.

Following Bollobás [10; pg 62], define $f(k)$ to be the minimum value of $\Phi(G)$ taken over all k-connected graphs G having at least one perfect matching. One can show that the Lovász result above can be used to obtain a lower bound on $f(k)$ due to Zaks [171; 172]. Zaks' result says that $f(k) \geq k!! = k(k-2)(k-4)\cdots$. If k is odd, the complete graph K_{k+1} serves to show that the above bound for $f(k)$ is sharp. (In fact, it may be shown that K_{k+1} is the *unique* extremal graph in this case.) If k is even, however, the situation is less clear, but more interesting. In this case, Mader [117] has shown that the extremal graph is again unique. These extremal graphs $S(k)$ are obtained by deleting a perfect matching from the complete graphs K_{k+2}.

Curiously, however, a closed form for $f(k)$ in this case is unknown. But one does have a recurrence relation for $f(k)$ for these extremal graphs. (See also Bollobás [10].)

Later, Mader [118] was able to strengthen the above results by dropping the as-

sumption that G be k-connected, and replacing it with the weaker assumption that G be 2-connected and have minimum degree at least k.

Finally, one can combine the tight set bound of $n/2+1$ for bicritical graphs with the Lovász "$k!$" theorem above to prove that any sufficiently large k-connected graph G having a perfect matching must have $\Phi(G) \geq k!$.

5. On Matchings in Random Graphs

Suppose among all the graphs on an even number of points, we select one "at random". How likely is it to contain a perfect matching? Failing that, what is the expected size of a maximum matching? Such statements, though decidedly imprecise at this stage, are nontheless, intuitively appealing. The idea of selecting a graph "at random", or selecting a "random" graph has its mathematical roots well back into the 1930's, according to Bollobás [14], but the concatenation of the two concepts of random graph and matching seems to have been first appeared in the pioneering series of papers by Erdős and Rényi [41; 42; 43; 44; 45].

Let n denote a fixed integer; we shall be considering graphs having n points.

There are quite a number of models of random graphs to be found in the literature. But two certainly predominate.

The first of these two we define as follows. Given a p, $0 < p < 1$, let $\mathcal{G}\{n,p\}$ denote the set of all graphs with point set $V = \{1,2,\ldots,n\}$ in which each line is present with probability p. Thus if G_0 denotes any fixed graph with point set V having m lines, then in this model, $\text{Prob}(G_0) = p^m(1-p)^{N-m}$, where $N = \binom{n}{2}$.

The second model of random graph which we now introduce will be the one most often used to state the results of this section. Let $\mathcal{G}(n,M)$ denote the set of all graphs having point set $V = \{1,2,\ldots,n\}$, having M lines in which each graph has the same probability. In other words, letting $N = \binom{n}{2}$, we have $0 \leq M \leq N$, the class $\mathcal{G}(n,M)$ contains $\binom{N}{M}$ members (labelled graphs) and every member occurs with equal probability $\binom{N}{M}^{-1}$. Normally, M is a function of n and is thus often denoted $M = M(n)$. A typical member of class $\mathcal{G}(n,M)$ will be denoted by $G_{n,M}$. For an arbitrary graph property Q we will say that a typical member of our probability space has property Q if the probability that a random graph on n points has property Q tends to 1 as $n \to \infty$. We also say, in this case, that *almost every* (a.e.) graph has property Q. A graph property Q is said to be **monotone** if whenever a graph G_1 has property Q and G_1 is a subgraph of G_2, then G_2 also has property Q. For example, "has a perfect matching" is a monotone property. (Remember: the number of points n is *fixed.*)

Next we need the concept of a *threshold function.* Let Q denote a monotone property. A function $M*(n)$ is a **threshold function** for property Q if

$$M(n)/M*(n) \to 0 \text{ implies that almost no } G_{n,M} \text{ has } Q,$$

and

$$M(n)/M*(n) \to \infty \text{ implies that almost every } G_{n,M} \text{ has } Q.$$

It is a fact, proved surprisingly recently, [18], that for *every* monotone property, such a threshold function must exist.

Erdős and Rényi were the first to suggest the following point of view toward random graphs. Start with the graph having no lines and add lines, one by one, selected at random; then try to determine how the graph "evolves" in doing so. It is one of the great accomplishments of random graph theory that so much can be said about this process.

We begin with a graph G_0 having n points and no lines. One line is added at random to obtain the graph G_1, another is added at random to obtain G_2, etc. In general, for $0 \leq t \leq N$, let G_t be the random graph obtained in this way.

When one adopts this point of view, it is useful to define another parameter closely associated with the threshold function, namely the so-called *hitting time* of monotone property Q. We define the hitting time of property Q as:

$$\tau_Q = \min \{t \geq 0 | \ G_t \text{ has } Q\}.$$

Then $M*(n)$ is the threshold function of Q if whenever $\omega(n) \to \infty$, the hitting time is almost surely between $M*(n)/\omega(n)$ and $M*(n)\omega(n)$. In other words,

$$\text{Prob}\{M*(n)/\omega(n) < \tau_Q(n) < M*(n)\omega(n)\} \to 1.$$

Although we are mainly interested in matchings, we cannot resist the temptation to devote a paragraph to "taking a stroll", so to speak, with our random graph as it evolves by this random addition of lines. (Our stroll will be a mathematically cavalier one; for more justification and rigor, see [14].)

Of course, our beginning graph contains no lines and hence has n components. As lines are randomly added, components begin to form. Although much can be said about what happens in the range $0 \leq M \leq n/2$, we will begin our observations only at this point. What now begins to happen is one of the great surprises to neophytes in random graph theory! One might think that components grow more or less at the same rate as we proceed. But quite the opposite is the case. In fact, there now begins the emergence of a "giant component" which ultimately swallows up all the remaining components.

When t reaches $\lfloor n/2 + 2(\log n)^{1/2}n^{2/3} \rfloor$, the graph G_t has a unique component of order at least $n^{2/3}$ and each of the other components has at most $n^{2/3}/2$ points each. As t increases further, i.e., as $t \geq c_0 n/2$, where $c_0 > 1$ is a constant, every remaining small component contains at most one cycle. Moreover, the order of these small components is $o(\log n)$. Should it happen that for any function $\omega(n)$ such that $\omega(n) \to \infty$, that $t \geq \omega(n)n$, then every component of G_t, other than the giant component, is a *tree*.

The next point of interest to us along this evolutionary trail occurs at $(n/2)\log n$. When t reaches this point, the giant component has succeeded in swallowing up all the other components (largest ones first!) and G_t becomes *connected*. This connectivity hitting time was first obtained by Erdős and Rényi in [41]. But in two subsequent papers ([44] and [45]), these same two authors were to prove two more very surprising results. First they proved that this same hitting time marked the point where the evolving graph G_t has minimum degree 1 and this in turn was the same point where (provided n is even), graph G_t has a perfect matching!

Since the beginning of this Chronicle on matching, we have often differentiated between the bipartite and non-bipartite cases. We might do the same here. Hence one might ask if one changed the probability space to one that was composed solely of random *bipartite* graphs, if the above triple coincidence of hitting times might change. The answer is "no"! (See [14].)

If one reflects a bit upon the fact that the hitting times for G_t to have a perfect matching (when n is even) and that for G_t to have minimum degree 1 coincide, it seems that the (probabilistic) obstruction to a random graph having a perfect matching is that it has points of degree 0. So what if one restricts the random graphs $G_{n,M}$ to those having minimum degree at least 1? Then it is not too surprising that the threshold drops. In fact, roughly speaking, it falls from $n/2 \log n$ to $(n/4)(\log n + 2n \log \log n$. (See [16].) (Of course, we still assume n even here.)

Let us pursue this line of enquiry one more step. Let us further restrict the minimum degree of our random graphs $G_{n,M}$ to be at least 2. Denote these random graphs by $G^{(2)}_{n,M}$. An open problem posed by Frieze [52] is the following:

Does $c > 1$ imply that a.e. $G^{(2)}_{n,cn}$ has a matching of size $\lfloor n/2 \rfloor$?

He has subsequently made some progress on this problem by showing that in the special case when the graphs are *bipartite* with minimum degree at least 2, have an even number of points, and constant $c \geq 2$, the random graph almost surely has a perfect matching [53]. The proof, however, of this result currently requires in excess of sixty pages!

We would also like to note that there is a random graph result akin somewhat to the results in Section 4 dealing with the number of perfect matchings in a k-connected (respectively, minimum degree $= k$) graph. Bollobás and Frieze [16] proved that as a random graph on an even number of points evolves, the hitting time for the minimum degree to be $\geq k$ is the same as the hitting time for the graph to have k line-disjoint perfect matchings.

Let us move on to yet another probability space — that of *r-regular* graphs. Suppose that r and n are positive integers such that $3 \leq r < n$ and that $rn = 2m$ is even. Let $\mathcal{G}(n, r - \text{reg})$ denote the set of all r-regular graphs on the point set $V = \{1, 2, \ldots, n\}$. (Since rn is even, we know this class is not empty.) Assign each member of this class the same probability. It then follows from a result of Bollobás [14; Theorem 32] that a.e. r-regular graph is r-connected. But it is also known [15] that every r-regular r-connected graph on n points contains $\lfloor n/2 \rfloor$ independent lines. Combining these results, we see that a.e. $G_{n,r-\text{reg}}$ with an even number of points has a perfect matching.

Our final example of a random graph probability space which has been studied with respect to perfect matchings is the so-called *k-out* model. It is yet another attempt to deal with the fact that points of degree 0 seem to be the blocking factor for the existence of a perfect matching in our first two random graph models studied in this section.

Once again let $V = \{1, 2, \ldots, n\}$ and let $\mathcal{D}(n, m)$ denote the set of *directed* graphs on point set V in which each point $v \in V$ has outdegree m. Choose $\overrightarrow{D}(n, m)$ uniformly at random from $\mathcal{D}(n, m)$ and then form the undirected graph $D(n, m)$ by ignoring the orientation of the lines of $\overrightarrow{D}(n, m)$. Frieze [51] proved that when n is even:

$$\lim_{n\to\infty} \text{Prob}(D(n,1) \text{ has a perfect matching}) = 0,$$

while

$$\lim_{n\to\infty} \text{Prob}(D(n,2) \text{ has a perfect matching}) = 1.$$

For most of the terminology and many of the theorems of this section, we have depended heavily upon the encyclopedic book of Bollobás [14]. For gentler introductions to the subject of random graphs, see [11] and [130]. For survey articles, see [12], [69] and [13].

6. Matching Algorithms

We will only be interested in *polynomial* algorithms (at least in the sequential machine setting) since the mathematical community has essentially unanimously agreed with Edmonds point of view expounded in his ground-breaking paper on non-bipartite matching [33] that polynomiality should be the crucial criterion for "goodness" of algorithms. With this polynomiality criterion in mind, however, it is difficult to trace the early history of matching *algorithms*.

By far the most fundamental idea in the area of matching algorithms — the *augmenting path* — has already been introduced in Section 2. But augmenting paths were known to Petersen, Kõnig, Egerváry and probably others long before anyone really cared about algorithms — be they "good" or "bad"!

Historically speaking, as we know, bipartite matching came first. We can say that although Kõnig's proof of his Minimax Theorem involved alternating paths, it was a proof by contradiction; it contained no algorithm as such. It seems that we must wait until the mid-50's for the true beginning of our algorithmic tale. In 1956, M. Hall [71] published an algorithm for constructing a system of distinct representatives, but so far as we know, no one bothered to analize its complexity. Shortly before that, Kuhn [103] who had studied carefully the methods of Kõnig and Egerváry, developed an algorithm for the Assignment Problem. (The Assignment Problem is a generalization of the bipartite matching problem in which weights are assigned to the lines of the graph.) His routine *was* polynomial, but neither he nor anyone else, as far as we know, made this observation at that time. It was in this paper that the term "Hungarian Method" was first used. To his credit, Kuhn attributes the algorithm to the seminal ideas of Kõnig and Egerváry set forth some twenty-five years earlier. In essence, Kuhn showed that a primal-dual algorithmic idea of Egerváry could be used and termination at optimality followed from Kõnig's Minimax Theorem.

In their book [49], published in 1962, Ford and Fulkerson showed that their labeling algorithm for Max-flow Min-cut could be adapted to solve the Assignment Problem, among others. Still no mention of polynomiality, but the authors did mention that in the case of *integral* capacities, their labeling method would terminate in a number of steps no greater than the maximum of the line capacities. They even gave an example of a network with irrational capacities such that their labeling process would not even terminate! In this example, the labeling algorithm produced an infinite number of flow-augmenting paths

which did, however, converge. On the other hand, the limit of convergence was far from the value of a maximum flow.

Not until some ten years later, did Edmonds and Karp overcome these difficulties [35]. They pointed out that in the Ford-Fulkerson labeling process there was a potentially dangerous ambiguity in the labeling and hence in the choice of a flow-augmenting path. Using the simple, yet ingenious, method of "first-labeled first-scanned" in the labeling procedure, Edmonds and Karp were able to show that not only did the algorithm converge to a maximum flow for arbitrary real capacities, but that the number of steps to termination was polynomial *only in the number of points* of the network. In other words, running time was independent of the capacities.

This, coupled with the fact that Ford and Fulkerson in their book [49] had shown that bipartite matching could be done with their labeling algorithm, showed explicitly for the first time that there was a bipartite matching algorithm which was polynomial in the number of points in the input graph.

At about the same time that Edmonds and Karp obtained their result, Soviet graph theorist, E.A. Dinic, published a similar improvement [32]. From that day to the present, there has been a stream of successive improvements in efficiency of flow algorithms. We will not pursue this further, since flows are not central to our mission. Instead, we direct the interested reader to [113; Chapter 2] and to the even more recent survey on flow algorithms [66].

The flow algorithm of Edmonds and Karp is $O(n^3)$. In the year following the publication of the Edmonds-Karp flow algorithm, Hopcroft and Karp [76] designed an $O(m\sqrt{n})$ bipartite matching algorithm. This remains the best time bound for bipartite matching today.

But what about *non*-bipartite matching? It was not until 1965 that we had any *polynomial* algorithm for general (i.e., non-bipartite) matching at all! In that year, Jack Edmonds published two papers which were to profoundly affect the study of matching theory. In the first [33], he developed the first polynomial algorithm for non-bipartite matching, the now-famous *blossom algorithm*. In the second, he developed the polyhedral approach to matching and employed it to develop a polynomial algorithm for *weighted* non-bipartite matching. With the benefit of hindsight, it is difficult to overestimate the importance of these papers. In [33], in addition to the important development of a polynomial algorithm for general matching, Edmonds expounded at some length on "efficient" algorithms and offered polynomiality as a measure for such efficiency. Some five years later, Cook [24] in an equally monumental work, laid the foundations for modern-day algorithmic complexity with the introduction of the classes P and NP and the discovery of the first NP-*complete* problem. (As the reader undoubtedly knows, "P" stands for "polynomial", or in contrast to NP, we should really say that class P stands for the class of decision problems for which there is a *deterministic* polynomial algorithm. Recall that P and NP were introduced in Section 2 of this paper.)

The reader may well have noted by this time that we have not treated weighted matching so far. In fact, we will not do so. Although it is an important branch of matching theory, due to space limitations, we have elected not to treat it in this paper. Fortunately, however, there are several sources to which we can direct the reader's attention. See [113],

[144], [143], [67] and [59]. It should be noted, however, that we have laid a fair amount of the groundwork for weighted matching in Section 3 by introducing the concepts of matching polytopes and their facets.

But let us return to Edmonds' algorithm. We will give only a superficial description of how it works.

Clearly Edmonds was motivated by the long-standing notion of augmenting paths. Suppose we begin the procedure with a given matching M. If M happens to be perfect, there is nothing to do, so we halt. So suppose that the set S of points not covered by M is not empty. We then construct a forest F such that every connected component of F contains exactly one point of S, every point of S belongs to exactly one component of F and every line of F which is at an odd distance from a point in S belongs to matching M. It follows that every point of F which is at an odd distance from S has degree 2 in F. Such points will be called **inner** points and all other points in F will be called **outer**. (Note that the points of S are outer.) Forest F is said to be **M-alternating**.

The idea is to try to enlarge F as much as possible. If we find an outer point x adjacent to a point y not in F, then line yz must be in M (or else y would have already been in the forest) and we can then enlarge forest F to a larger forest $F' = F + xy + yz$ which is clearly also M-alternating.

If F has two outer points x and y which belong to different components of F and which are adjacent in G, then the roots of these two components are connected by an M-augmenting path joining the respective roots of the two components and using the line xy. So we can augment M to a matching M' such that $|M'| = |M| + 1$ and we begin again with the new larger matching.

Suppose that every outer point has only inner points as neighbors. Then we claim that the matching M at hand is a maximum matching. For suppose that the forest F contains a inner points and b outer points. Clearly, $b - a = |S|$. Moreover, if we delete all the inner points of F from graph G, the remaining graph will contain all the outer points of F as *isolated* points. At this point we appeal to Berge's deficiency version of Tutte's 1-factor Theorem. Let the **defect** of any matching M denote the number of points of G not covered by M. Let d_1 denote the minimum taken over the defects of all matchings in G. On the other hand, let $d_2 = \max \{c_o(G - X) - |X| | X \subseteq V(G)\}$. Then Berge's result states that $d_1 = d_2$. Let X_i denote the set of all inner points relative to matching M and as above, assume that no two outer points are adjacent. Then M is a matching with defect $= S$, so $d_1 \leq |S|$. On the other hand, $|S| \leq c_o(G - X_i) - |X_i| \leq d_2$. Thus it follows from Berge's theorem that M is a maximum matching.

So far so good, but the careful reader will see that we have omitted treating precisely one case. What if forest F has two outer points x and y in the *same* connected component (tree) of the forest which are adjacent in G? Thus an odd cycle of a particular type is formed; namely, one rooted at an outer point and having the property that all the other points of the cycle are covered by the matching M. Such an odd cycle is called a **blossom**. What can we do with blossoms?

Edmonds answer: Shrink them! If blossom B is shrunk to a single point and the resulting smaller graph is denoted by G', and if $M' = M - E(B)$, then the crucial "Cycle Shrinking Lemma" says that M' is a maximum matching in G' if and only if M is a

maximum matching in G.

Thus we have an algorithm which terminates in a maximum matching. Moreover, the geometry of the steps is appealingly simple. That's the good news. The bad news is that the data structures necessary to implement Edmonds algorithm are very complicated and hence the time bound is not clear. The first careful analyses for running time seem to have been carried out by Gabow [55; 56] and Lawler [104] each of whom obtained an implementation of the algorithm which was $O(n^3)$.

Since 1976, a number of successive improvements have been made both in the simplicity of the data structures involved and the time bound. At the present time, the fastest known cardinality matching algorithm is due to Micali and V.V. Vazirani [122], and curiously enough, it has the same polynomial time bound as that of Hopcroft and Karp for bipartite graphs. Unhappily, they must still deal with blossoms. Indeed, it remains to be seen if one can design a polynomial matching algorithm which remains competitive with Micali-Vazirani, but somehow avoids blossoms and their attendant trouble.

Lovász [113] has built a blossom-free matching algorithm based upon the Gallai-Edmonds decomposition procedure discussed in Section 4. His algorithm is $O(n^4)$, but it is unknown to the author whether or not implementation of this algorithm has ever been undertaken.

So far, all our algorithmic discussions have dealt with the *sequential* approach. Although we want to avoid such technicalities as much as possible in this paper, one can safely assume that our algorithms have been carried out on a random access machine (RAM). For those who interested in such matters, see [1].

In the last twenty years or so, the world of computing has seen tremendous strides in the miniturization of chips and a steady decrease in the cost of their production. Such real world considerations have served to buttress the case for developing *parallel* computers in which large jobs can be subdivided into smaller tasks and farmed out to individual processors. The processors can then perform their assigned tasks "in parallel" and communicate their answers to a central processor for a final output.

Formidable problems arise immediately, however. What type of architecture should be adopted? In othe words, how should the individual processors be arranged and linked together in order to achieve "optimal" efficiency? How does one avoid network "gridlock"? That is, how can the entire parallel computation be sychronized? How can a given problem be efficiently subdivided into smaller parts which can then be effectively done in parallel and then be combined to produce an answer?

Awesome problems indeed! And, we hasten to add, far from solved in most senses of the word. But how does all this intersect with matching theory?

We will give a rough overview of the present state of affairs vis-à-vis parallel matching problems. We shall avoid discussion and comparison of various parallel computer models. In this matter, let it suffice to say that we will adopt as our model the standard parallel random access machine — or PRAM — and one can read about PRAMs in [50] as well as many other places. A particularly nice overview of various models of PRAMs from the point of view of the mathematician can be found in the new survey of parallel algorithms by Karp and Ramachandran [83]. In particular, another very important model of parallel computation is the *Boolean circuit model*. It is especially useful for defining certain com-

plexity classes for parallel computation. But due to space limitations, we shall not deal with it here. Again we refer the reader to [83] and, for the most complete and up-to-date treatment of circuit models for parallel computation, to [20] in the same volume.

When switching from sequential to parallel computation, some new philosophical questions immediately arise. (Well, philosophical, yes, but also very easily translatable into cold hard cash!)

Of course, any problem which can be solved sequentially can trivially be solved in parallel, but the important question is: can it be done *faster* in parallel? For that matter, what should "faster" mean here? And what about the *cost* of achieving such speed-up? How many processors will be needed to achieve a given speed-up factor?

Here is another question which we find particularly intriguing. Are there graph problems which are somehow "inherently sequential" in the sense that no parallel algorithms can be found which will offer a significant speedup in computation time? Although we are being deliberately vague here with the phrase "inherently sequential", let us illustrate with a simple problem from matching. Suppose we wish to find a maximal matching in a graph. (That's right; a maxim*al* matching, not a maxim*um* matching!) Such a task is a triviality in the sequential sense. Choose any line. Delete its endpoints. Choose any line in the remaining graph and add it to the first. Delete its endpoints and continue in this manner until there are no lines left to choose. Presto! We have a maximal matching — and in polynomial time. But how can one do this (non-trivially) in parallel?

Before we return to this and other matching questions, we must lay some groundwork. The reader will recall our earlier discussion of the *sequential* complexity classes P and NP. We now introduce what has become the most widely studied *parallel* complexity class. It is called NC, the letters corresponding to the abbreviation "Nick's Class" after Nick Pippenger who first studied it [134]. Class NC consists of those problems solvable in "polylog" time using a polynomial number of processors. (Here "polylog" refers to polylogarithmic time, that is to say, in time polynomial *in the logarithm* of the size of the input.) Clearly, NC lies in sequential class P. Could it possibly be that NC = P? This is an open question.

Actually, a finer classification than just NC has been introduced and studied as well. It focuses upon the degree of the polynomial in the polylog time part of the definition. More particularly, let us define class NC^i to be the class of problems in NC in which the polynomial in the logarithm of input size is of degree i. Then of course $NC = \cup_1^\infty NC^i$.

Now let us return to matching. The first result on maximal matching in parallel is due to Lev [106]. She showed that the problem was in NC^4 for bipartite graphs. Maximal matching in general graphs was first shown to be in NC by Karp and Wigderson [85] who actually solved the more difficult problem of showing that the *maximal independent set* problem was in NC. Since matchings in a graph G correspond to independent sets (of points) in the line graph $L(G)$, the matching result follows.

The Karp and Wigderson result actually showed that maximal matching is in NC^4. Luby [114] subsequently showed that in fact maximal matching is in NC^2. Luby's approach, like that of Karp and Wigderson, was through a speed-up in the maximal independent set problem. All of these results are truly eye-openers in the sense that these parallel algorithms for doing this sequentially trivial problem are quite complex. In particular, Karp

and Wigderson appeal to the use of block designs, while Luby first obtains a "randomized" version of the algorithm and then uses a beautiful probabilistic argument to show that the randomness can be eliminated! Luby's approach seems especially novel and one wonders if it will bear further fruit in designing graph algorithms. (See also Luby [115].)

In view of the above discussion on maximal matching, we approach our old friends — finding maximum or perfect matchings — with not a little fear and trepidation!

Let us start by informing the reader that no NC-algorithm for finding a maximum or a perfect matching is known. In fact, it is not even known if the decision problem "does graph G have a perfect matching?" lies in NC.

Some interesting special cases of these questions have been settled, however. For example, if a *bipartite* graph has a *polynomially-bounded* number of perfect matchings, then the decision and search problems have been shown to be in classes NC^2 and NC^3 respectively by Grigoriev and Karpinski [68].) In contrast, Dahlhaus and Karpinski [28] have shown that if a graph G is *dense* (and has an even number of points), then the search problem is in NC^2. (A graph G is **dense** if mindeg $(G) \geq n/2$. Note that in this class of graphs, the decision problem always has the answer "yes" since by a well-known theorem of Dirac, dense graphs all have Hamiltonian cycles.)

In another direction, although no polylog-time matching algorithms are yet known, there do exist *sub-linear* matching algorithms, at least in the bipartite case. To the author's knowledge, the fastest known maximum matching algorithm of this type is due to Goldberg, Plotkin and Vaidya [65].

In an exciting new direction, however, progress on matching (and many other) problems has been made when one allows *randomization* to creep into one's algorithms. But what do we mean by "randomization"? Randomization comes in all shapes and sizes and constitutes a subject for study unto itself. For our Sampler purposes, we will rely heavily upon the treatments of Karp [82] and Johnson [79]. Randomized algorithms receive, in addition to the input graph, some additional bits from some "random" source and then perform their computations based upon both types of input. Sometimes, randomized algorithms are described as ones which are allowed to flip coins during their execution. Let us agree that our output is to be either "yes" or "no". Then one can define various types of randomized algorithms and associated complexity classes by the manner in which they adher to the truth. For example, a *Monte Carlo* algorithm behaves as follows. If the input problem has "yes" as its answer, the algorithm will answer "yes" with some probability exceeding $1/2$. But if the correct answer is "no", the algorithm remains silent. Thus if the correct answer is "yes" and one runs the algorithm a number of times on this same input and if one gets even one "yes" output response from the algorithm, one is *certain* that the answer is truly "yes". If the correct answer is "no", the machine will never lie and tell you that the answer is "yes". On the other hand, if after a large number of trials, say k, the machine has always answered "no", the probability that the correct answer is "yes" is at most $1/2^k$. Thus it is very likely that the correct answer is "no".

The class of (sequential) decision problems for which polynomial-time Monte Carlo algorithms exist is denoted by **RP**. The parallel complexity class corresponding to NC in the same way that RP corresponds to P is denoted by **RNC**.

An RNC^2 algorithm for testing for the *existence* of a perfect matching was first found

by Lovász [109] (see also [113]) who reduced the problem to deciding whether or not a given integer matrix is non-singular. This, in turn, is known to be in RNC^2 by a result of Csánky [25]. But if we know that graph G *has* a perfect matching; how can we find one (in parallel)? Karp, Upfal and Wigderson [84] produced a parallel search algorithm for a perfect matching which is RNC^3. This was later improved by Mulmuley, Vazirani and Vazirani [126] who found an RNC^2 algorithm. The design of the algorithm is ingenious enough to deserve a comment. Remember: the Lovász-Csánky result stated above can be used to decide whether or not G has a perfect matching in RNC^2. So suppose we run this algorithm as a subroutine and find that indeed G does have a perfect matching. Now assign random integer weights to the lines of G, chosen randomly and uniformly from $\{1, 2, \ldots, m\}$, where $m = |E(G)|$. They then show that G will have a *unique* minimum weight perfect matching with probability at least $1/2$. Next they show how to find this minimum weight, and finally how to find the perfect matching itself!

They then show how to extend their algorithm to one for finding a *maximum* matching. The extended algorithm remains in RNC^2. In fact, they do even more.

The *exact matching problem* can be stated as follows and is credited to Papadimitriou and Yannakakis [131]. Given a graph with a distinguished subset of lines called "red" and an integer k. Does G have a perfect matching containing exactly k red lines? Mulmuley, Vazirani and Vazirani extend their methods to provide an RNC^2 algorithm to decide this problem. This is of special interest, as the *exact* matching problem is of unknown deterministic complexity! One result in this direction has been recently obtained by V.V. Vazirani, however [164: 165]. If graph G is $K_{3,3}$-free (See Section 4) then there is an NC algorithm to decide exact matching.

Before leaving the existence and search problems for matchings, we cannot resist stating two more results. Although it remains unknown whether or not the existence problem for perfect matchings lies in NC, if the graph in question happens to have a *unique* perfect matching, Kozen, Vazirani and Vazirani [101] have developed an NC algorithm for the existence problem. (Although the authors do not say so, it appears that their algorithm actually is NC^2.) On the other hand, if one is given a graph having a unique perfect matching, Rabin and Vazirani [146] have recently shown that the search problem for the perfect matching is in NC^2. The publication dates of these two papers belie the true sequence of events. The Rabin-Vazirani paper actually existed in preprint form for at least five years before its recent publication and its contents became rather widely known to those in the area, including Kozen, Vazirani and Vazirani. Thus chronologically speaking, the search problem for the special case of graphs with unique perfect matchings was actually shown to be in NC before the decision problem was similarly settled.

We would like to make one final remark before leaving the study of parallel matching algorithms per se. The classes NC and RNC by definition place strong emphasis on increasing the speed of parallel algorithms at the expense of almost ignoring the price to be paid in increased number of processors. Processors may be cheap in future parallel computers, but perhaps not that cheap! There are a number of papers scattered throughout the parallel algorithm literature, including that pertaining to matching, which do deal with such processor penalties. Due to our space limitations, however, we will not discuss this matter.

Now let us once again turn from existence and search problems for matchings back to the problem of *counting* them. The reader will recall from earlier in this paper that it is known that *exact* counting of perfect matchings is known to be #P-complete. From this it follows that the problem is at least as hard as any problem in NP. It is unknown if NP = #P, but most mathematicians think it unlikely. So what can we hope to do, algorithmically, toward counting perfect matchings?

We do have the lower bounds described in Section 5, but they are not sharp. We also have polynomial algorithms for $\Phi(G)$ in certain special cases (see the discussion of Pfaffians in Section 4). We will now briefly describe a third approach to the problem.

Quite recently, much excitement has been generated in the area of *approximating* the value of $\Phi(G)$. But here the approach is quite different from those treated in Section 4. It deals with approximating the value of the *permanent* of a matrix, and through that, the value of $\Phi(G)$ for *bipartite* graphs.

Let $A = (A_{i,j})$ denote any $n \times n$ matrix of positive integers. It is easy to define the **permanent** of matrix A as follows. Form all the $n!$ terms that one forms to compute the determinant of A, change all those terms having minus signs to plus, and add all $n!$ terms together. Despite the simplicity of definition, the permanent function has proved to be notoriously hard to handle! (See Minc [123], for a general reference on permanents.) Also despite its close resemblance to the determinant, whereas the determinant can be polynomially evaluated by any student in a first course in linear algebra, the best known algorithm for evaluating the permanent has a time bound of $O(n2^n)$, [151].

Why should we want to have anything to do with such a badly behaved function? Because if one confines the entries of the matrix A to either 0's or 1's and then considers the resulting binary matrix to be the bi-adjacency matrix of a bipartite graph G on $2n$ points, the permanent of the resulting matrix is exactly $\Phi(G)$!

But since Valiant proved that evaluating the permanent is #P-complete, what good does all this do us? Quite recently, there has been a flurry of activity in the area of *approximating* the permanent. Of course, questions always beget questions. What do we mean by "approximation"? And then what shall we adopt as a measure of "good" approximation? We shall give the briefest of reports on this. For more a more detailed survey of this topic the reader is directed to the recent paper of Luby [116].

Let us begin with a few definitions. An (ϵ, δ) **approximation algorithm** for per (A) is a probabilistic (Monte Carlo) algorithm which accepts as input the matrix A and the two positive real numbers, ϵ and δ. The algorithm then outputs a number Y as an estimate of per (A) which satisfies:

$$\text{Prob}[(1-\epsilon)\text{per }(A) \leq Y \leq (1+\epsilon)\text{per }(A)] \geq 1-\delta.$$

An (ϵ, δ) approximation algorithm is said to be a **fully-polynomial randomized approximation scheme (fpras)** (or simply **fully polynomial**) if its running time is polynomial in n, $1/\epsilon$ and $1/\delta$.

It is an open question as to whether or not there exists a fpras for the permanent function.

Very recently, two major lines of research on this question have begun. The first of these has resulted in an approximation algorithm which meets the accuracy demand of a

fpras, but in superpolynomial time. More specifically, Karmarkar, Karp, Lipton, Lovász and Luby [81] have designed a Monte Carlo algorithm which yields the desired output in time $2^{n/2}(1/\epsilon^2)\log(1/\delta)p(n)$, where $p(n)$ is a polynomial in n. For fixed ϵ and δ this is about the square root of the time bound for Ryser's algorithm.

The five authors of this work also pose the following open question: Is there a *deterministic* algorithm with running time $o(2^n)$ which accepts as input matrix A, positive real ϵ and outputs Y such that

$$(1-\epsilon)\text{per}\ (A) \leq Y \leq (1+\epsilon)\text{per}\ (A)\ ?$$

The second approach which seems to have originated with an idea of Broder [22] and been completed by Jerrum and Sinclair [77; 78]. The latter two authors succeeded in finding a fpras for *dense permanents*, that is, for dense bipartite graphs. The Jerrum-Sinclair paper (and several other companion papers) are not only important for this result, but perhaps even more so for introducing novel approaches using such esoteric concepts from probability theory as rapidly mixing Markov Chains and conductance. (Broder, too, deals with slightly modified versions of same.) In general, their common approach deals with reducing the problem of approximately counting perfect matchings to that of generating them randomly from an almost uniform distribution.

Using these ideas, Dagum, Luby, Mihail and U.V. Vazirani [27] (see also Dagum and Luby [26]) have achieved a polynomial speed up of the algorithm of Jerrum and Sinclair and used it to show that there is also a fpras for bipartite graphs with *large factor size*. The factor size of a bipartite graph $G = A \cup B$ (where still $|A| = |B| = n$) is the maximum number of line disjoint perfect matchings in G. (Observe that a graph with an αn-factor must have minimum degree at least αn, but not necessarily vice-versa.) In this area, Dagum and Luby have shown that there is a fpras for bipartite graphs with factor size at least αn for any constant $\alpha > 0$.

These results are even more interesting when compared to some related new completeness results. Broder [22] has shown that *exact* counting in dense graphs is as hard as exact counting in general and so is #P-complete. Dagum and Luby [26] and Dagum, Luby, Mihail and U.V. Vazirani [27] have shown that exact counting in $f(n)$-regular bipartite graphs is #P-complete for any $f(n)$ such that $3 \leq f(n) \leq n-3$. In fact they show that for any $\epsilon > 0$, for any function $f(n)$ such that $3 \leq f(n) \leq n^{1-\epsilon}$, a fpras for $f(n)$-regular bipartite graphs would imply the existence of a fpras for *all* bipartite graphs!

Before leaving the subject of algorithms and matching, we would be remiss if we did not at least mention the important problem of *lower* bounds on computational complexity. To this point, our emphasis has been on searching for *upper* bounds in series and parallel for such problems as deciding the existence of a perfect matching, searching for maximal and maximum matchings, counting the number of perfect matchings, etc. Moreover, the criterion for success normally adopted has been the discovery of a *polynomial* algorithm. Of course, we have not always succeeded! The main problem discussed for which success has eluded us (at least so far) is the exact determination of $\Phi(G)$.

On the other hand, we have the P = NP question. Should equality hold here, we will instantaneously have polynomial algorithms for literally hundreds of problems in graph theory and many other areas of computation; not just those problems with which we have

dealt in this paper involving matchings. But most people involved in computation believe that P $\neq$ NP. How can we possibly approach a proof that equality does not hold?

Perhaps someone someday will succeed in proving that certain problems in NP *cannot possibly have* a polynomial algorithm. It would seem at the moment, however, that we are very far from accomplishing this. But a first step recently taken by A.A. Razborov [148; 149] has generated a strong ripple of excitement in the world of computational complexity. In order to give a good explanation of Razborov's accomplishments, we must deal with a model for measuring complexity quite different (at least formally) from the RAM or Turing machine models we have dealt with without exception in this paper so far. This is the so-called *Boolean circuit model.* Here, once again, our overview will be very superficial, due to space constraints. We direct the reader to Boppana and Sipser [20] for an excellent overview of the subject, including the Razborov results we are about to describe.

The circuit model seems to have been first introduced as a measure of computational complexity by Shannon [156]. A *Boolean circuit* is essentially an acyclic directed graph. The points having indegree 0 are called **inputs** and those having indegree greater than 0 are called **gates**. Normally these gates act as one of the Boolean functions AND, OR or NOT. One of these is distinguished and designated as the *output* gate. The **size** of the circuit is the number of gates. The **circuit complexity** of a Boolean function f is the size of a smallest Boolean circuit which, given binary input to the input points of the circuit, will compute the value of f. In particular, a problem (function) has **polynomial circuit complexity** if its circuit complexity is a polynomial in the number of gates in some circuit representing f. It would be very nice indeed for our purposes if a problem were in P if *and only if it* had polynomial circuit complexity. This is not quite the case unless one inserts the concept of *uniformity* into the discussion of circuits. We will avoid this by simply noting that it is true in general that every problem in P has polynomial circuit complexity. In order to approach Razborov's results, it is perhaps better to think in terms of the contrapositive of the preceding fact. Namely, if one can show that a problem has superpolynomial circuit complexity, then it is not in P. More specifically for our purposes, if one could show that a problem *in NP* has superpolynomial circuit complexity, then P $\neq$ NP!

Clearly, no one has succeeded in accomplishing this to date. But an interesting approach has been taken involving the concept of a *monotone* circuit. Simply put, a Boolean circuit is **monotone** if it contains no NOT gates. Razborov's result on the matching problem can now be simply stated. *The decision problem for perfect matchings in bipartite graphs has superpolynomial monotone circuit complexity.* But this matching problem is well-known to be in P, so what is the significance of Razborov's result for us?

We should mention that Razborov has also shown that the clique problem for graphs also has superpolynomial monotone circuit complexity [148]. So the clique problem serves as an example of an NP-complete problem with a superpolynomial monotone circuit lower bound. The fact that this also can happen for the matching problem — a problem in P — may be more of an indication of just how much monotone and non-monotone circuits differ in power, rather than an indication of the fundamental differences, if any, between P and NP. Be that as it may, whether or not the monotone circuit approach will serve as a useful tool in the ultimate resolution of the P = NP conundrum only time will tell, of

course.

This brings us to the end of our Sampler and more or less, up to date in the field of matching theory. As we have said several times above, it is truly only a Sampler, for a number of important matching topics, such as weighted matching, b-matching, matching polynomials, matroid matching, etc., etc, have not been addressed. However, a number of other problems on maximal, maximum and perfect matchings and their relatives, together with analogous problems for independent sets (i,e, vertex packing) will be treated in forthcoming paper [140]. The emphasis there will be on complexity.

It is likely that by the time the present paper appears in print, new and even more interesting results will have been discovered which deserve to be included in our Chronology and Sampler. We certainly hope so.

Acknowledgements

We would like to take this opportunity to gratefully acknowledge the kind assistance of Richard Anstee, Laci Babai, David Johnson, Dick Karp, Laci Lovász, Mike Luby, Bill Pulleyblank and Carsten Thomassen. We would also be remiss did we not thank Pulleyblank for his professional assistance in the presentation and associated "media show" that accompanied the talk at Hindsgavl which formed the preliminary basis for this paper!

Postscripts

Remark 1. After, the completion of this paper, the author became aware of two preprints of F. Barahona (See [173; 174]). Here Barahona addresses the *as yet unsolved* problem of formulating the weighted matching problem as a linear program with a polynomial number of facets. In [173], he shows that this can be done for the weighted matching problem in *planar* graphs, while in [174], he obtains a partial solution to the general problem by showing that the problem of weighted matching in general can be solved by a polynomial sequence of linear programs each of which has a polynomial number of facets.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, 1974.

[2] R. Anstee, An algorithmic proof of Tutte's f-factor theorem, *J. Algorithms*, **6**, 1985, 112-131.

[3] F. Bäbler, Über die Zerlegung regulärer Streckencomplexe ungerader Ordnung I, *Comment. Math. Helv.*, **10**, 1938, 275-287.

[4] H.B. Belck, Reguläre Faktoren von Graphen, *J. Reine Angew. Math.*, **188**, 1950, 228-252.

[5] C. Berge, Two theorems in graph theory, *Proc. Nat. Acad. Sci. U.S.A.*, **43**, 1957, 842-844.

[6] _____, Sur le couplage maximum d'un graphe, *C.R. Acad. Sci. Paris Sér. I Math.*, **247**, 1958, 258-259.

[7] _____, *Théorie des Graphes et ses Applications*, Dunod, Paris, 1958.

[8] _____, *The Theory of Graphs and its Applications*, John Wiley, New York, 1962.

[9] _____, Alternating chain methods: a survey, in: *Graph Theory and Computing*, Ed.: R. Read, Academic Press, New York, 1972, 1-13.

[10] B. Bollobás, Chapter 2 in: *Extremal Graph Theory*, Academic Press, London, 1978, 50-101.

[11] _____, *Graph Theory — An Introductory Course*, Grad. Texts in Math. Vol. **63**, Springer-Verlag, New York, 1979.

[12] _____, Random graphs, in: *Combinatorics (Proc. Eighth British Combinatorial Conference, Swansea, 1981)*, Ed.: H.N.V. Temperley, London Math. Soc. Lecture Note Series **52**, Cambridge Univ. Press, Cambridge, 1981.

[13] _____, Lectures on random graphs, in: *Progress in Graph Theory*, Eds.: J.A. Bondy and U.S.R. Murty, Academic Press, Toronto, 1984, 3-42.

[14] _____, *Random Graphs*, Academic Press, London, 1985.

[15] B. Bollobás and S.E. Eldridge, Maximal matchings in graphs with given minimal and maximal degrees, *Math. Proc. Cambridge Philos. Soc.*, **79**, 1976, 221-234.

[16] B. Bollobás and A.M. Frieze, On matchings and Hamiltonian cycles in random graphs, *Random Graphs '89 (Poznan)*, Eds.: M. Karonski and A. Rucinski, Math. Study **118**, North-Holland, Amsterdam, 1985, 23-46.

[17] B. Bollobás, A. Saito and N.C. Wormald, Regular factors of regular graphs, *J. Graph Theory*, **9**, 1985, 97-103.

[18] B. Bollobás and A.G. Thomason, Threshold functions, *Combinatorica*, **7**, 1987, 35-38.

[19] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, MacMillan Press, London, 1976.

[20] R.B. Boppana and M. Sipser, The complexity of finite functions, Chapter 14 in: *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, Ed.: J. van Leeuwen, The MIT Press/Elsevier, Cambridge/Amsterdam, 1990, 757-804.

[21] L.M. Brégman, Certain properties of nonnegative matrices and their permanents, *Dokl. Akad. Nauk SSSR*, **211**, 1973, 27-30. (in Russian). English translation: *Soviet Math. Dokl.*, **14**, 1973, 945-949.

[22] A. Broder, How hard is it to marry at random (On the approximation of the permanent), *Proc. 18th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1986, 50-58.

[23] V. Chvátal, *Linear Programming*, Freeman, New York, 1983.

[24] S.A. Cook, The complexity of theorem proving procedures, *Proc. Third Annual ACM Symposium on Theory of Computing*, ACM, New York, 1971, 151-158.

[25] L. Csánky, Fast parallel matrix inversion algorithms, *SIAM J. Computing*, **5**, 1976, 618-623.

[26] P. Dagum and M. Luby, Approximating the permanent of graphs with large factors, preprint, February, 1991, to appear.

[27] P. Dagum, M. Luby, M. Mihail and U.V. Vazirani, Polytopes, permanents and graphs with large factors, *Proc. 29th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, New York, 1988, 412-421.

[28] E. Dahlhaus and M. Karpinski, Parallel construction of perfect matchings and Hamiltonian cycls on dense graphs, *Theoret. Comput. Sci.*, **61**, 1988, 121-136.

[29] G.B. Dantzig, Reminiscences about the origins of linear programming, in: *Mathematical Programming - The State of the Art*, Eds.: A. Bachem, M. Grötschel and B. Korte, Springer-Verlag, Berlin, 1983, 78-86.

[30] G.B. Dantzig and D.R. Fulkerson, On the max-flow min-cut theorem of networks, *Linear Inequalities and Related Topics*, Ann. of Math. Study 38, Princeton Univ. Press, Princeton, 1956, 215-221.

[31] R.P. Dilworth, A decomposition theorem for partially ordered sets, *Ann. of Math.*, **51**, 1950, 161-166.

[32] E.A. Dinic, Algorithm for solution of a problem of maximum flow in a network with power estimation, *Dokl. Akad. Nauk SSSR*, **194**, 1970, 754-757. (in Russian). English translation: *Soviet Math. Dokl.*, **11**, 1970, 1277-1280.

[33] J. Edmonds, Paths, trees and flowers, *Canad. J. Math.*, **17**, 1965, 449-467.

[34] _____, Maximum matching a polyhedron with (0,1)-vertices, *J. Res. Nat. Bur. Standards Sect. B*, **69b**, 1965, 125-130.

[35] J. Edmonds and R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *J. Assoc. Comput. Mach.*, **19**, 1972, 248-264.

[36] J. Edmonds, L. Lovász and W.R. Pulleyblank, Brick decompositions and the matching rank of graphs, *Combinatorica*, **2**, 1982, 247-274.

[37] E. Egerváry, On combinatorial properties of matrices, *Math. Lapok*, **38**, 1931, 16-28. (in Hungarian).

[38] G.P. Egoryčev, Solution of the van der Waerden problem for permanents, IFSO-13M, *Akad. Nauk SSSR Sibirsk. Otdel. Inst. Fiz.*, Krasnoyarsk, preprint, 1980. (in Russian.) English translation: *Soviet Math. Dokl.*, American Mathematical Society, Providence, **23**, 1982, 619-622.

[39] _____, The solution of van der Waerden's problem for permanents, *Adv. in Math.*, **42**, 1981, 299-305.

[40] P. Elias, A. Feinstein and C.E. Shannon, Note on maximum flow through a network, *I.R.E. Trans. on Inform. Theory*, IT-2, 1956, 117-119.

[41] P. Erdős and A. Rényi, On random graphs I, *Publ. Math. Debrecen*, **6**, 1959, 290-297.

[42] _____, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.*, **5**, 1960, 17-61.

[43] _____, On the evolution of random graphs, *Bull. Inst. Int. Statist. Tokyo*, **38**, 1961, 343-347.

[44] _____, On random matrices, *Publ. Math. Inst. Hungar. Acad. Sci.*, **8**, 1964, 455-461.

[45] _____, On the existence of a factor of degree one of a connected random graph, *Acta Math. Acad. Sci. Hungar.*, **17**, 1966, 359-368.

[46] D.I. Falikman, A proof of the van der Waerden conjecture on the permanent of a doubly stochastic matrix, *Mat. Zametki*, **29**, 1981, 931-938. (in Russian.) English translation: *Mathematical Notes of the Academy of Science of the USSR*, Consultants Bureau, New York, **29**, 1981, 475-479.

[47] L.R. Ford, Jr. and D.R. Fulkerson, Maximal flow through a network, *Canad. J. Math.*, **8**, 1956, 399-404.

[48] _____, A simple algorithm for finding maximal network flows and an application to the Hitchcock problem, *Canad. J. Math.*, **9**, 1957, 210-218.

[49] _____, *Flows in Networks*, A Rand Corporation Research Study, Princeton Univ. Press, Princeton, 1962.

[50] S. Fortune and J. Wyllie, Parallelism in random access machines, *Proc. 10th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1978, 114-118.

[51] A.M. Frieze, Maximum matching in a class of random graphs, *J. Combin. Theory Ser. B*, **40**, 1986, 196-212.

[52] _____, On matchings and Hamilton cycles in random graphs, Carnegie Mellon Univ. Dept. of Math. Research Report No. 88-36, October, 1988.

[53] _____, On perfect matchings in random bipartite graphs with minimum degree at least two, Carnegie Mellon Univ. Dept. of Math. Research Report No. 90-86, June, 1990.

[54] G. Frobenius, Über Matrizen aus nicht negativen Elementen, *Sitzungsber. König. Preuss. Akad. Wiss.*, **18**, 1912, 456-477.

[55] H.N. Gabow, Implementation of algorithms for maximum matching on non-bipartite graphs, Stanford Univ. Dept. Computer Sci. Ph.D. Thesis, 1973.

[56] _____, An efficient implementation of Edmonds' algorithm for maximum match-

ing on graphs, *J. Assoc. Comput. Mach.*, **23**, 1976, 221-234.

[57] _____, An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems, *Proc. 15th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1983, 448-456.

[58] D. Gale, H.W. Kuhn and A.W. Tucker, Linear programming and the theory of games, in: *Activity Analysis of Production and Allocation*, Ed.: T.C. Koopmans, Wiley, New York, 1951, 317-329.

[59] Z. Galil, Sequential and parallel algorithms for finding maximum matchings in graphs, *Ann. Rev. Comput. Sci.*, **1**, 1986, 197-224.

[60] T. Gallai, On factorisation of graphs, *Acta Math. Acad. Sci. Hungar.*, **1**, 1950, 133-153.

[61] _____, Kritische Graphen II, *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, **8**, 1963, 373-395.

[62] _____, Maximale Systeme unabhängiger Kanten, *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, **9**, 1964, 401-413.

[63] _____, The life and scientific work of Dénes Kõnig (1884-1944), *Linear Alg. Appl.*, **21**, 1978, 189-205.

[64] A.B. Gamble, Polyhedral extensions of matching theory, Univ. of Waterloo Dept. of Combinatorics and Optimization Ph.D. Thesis, 1989.

[65] A.V. Goldberg, S.A. Plotkin and P.M. Vaidya, Sublinear-time parallel algorithms for matching and related problems, *Proc. 29th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, New York, 1988, 174-185.

[66] A. Goldberg, É. Tardos and R. Tarjan, Network flow algorithms, in: *Paths, Flows and VLSI-Layout*, Eds.: B. Korte et al., Algorithms and Combinatorics Vol. **9**, Springer-Verlag, Berlin, 1990, 101-164.

[67] M. Gondran and M. Minoux, Chapter 7 in: *Graphs and Algorithms*, Wiley Interscience, Chichester, 1984.

[68] D. Yu. Grigoriev and M. Karpinski, The matching problem for bipartite graphs with polynomially bounded permanents is in NC (Extended Abstract), *Proc. 28th Annual IEEE Symposium on Foundations of Computer Science* IEEE Computer Society Press, New York, 1987, 166-172.

[69] G. Grimmett, Random Graphs, Chapter 7 in: *Selected Topics in Graph Theory 2*, Eds.: L.W. Beineke and R.J. Wilson, Academic Press, London, 1983, 201-235.

[70] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms and Combinatorics 2, Springer-Verlag, Berlin, 1988.

[71] M. Hall, An algorithm for distinct representatives, *Amer. Math. Monthly*, **63**, 1956, 716-717.

[72] P. Hall, On representatives of subsets, *J. London Math. Soc.*, **10**, 1935, 26-30.

[73] A.J. Hoffman, Some recent applications of the theory of linear inequalities to extremal combinatorial analysis, *Combinatorial Analysis*, Proc. Symposia in Applied Math., Vol. 10, Amer. Math. Soc., Providence, 1960, 113-127.

[74] _____, D.R. Fulkerson's contributions to polyhedral combinatorics, *Polyhedral Combinatorics — dedicated to the memory of D.R. Fulkerson*, Eds.: M. Balinski and A.J. Hoffman, Math. Prog. Study No. 8, North-Holland, Amsterdam, 1978, 17-23.

[75] A.J. Hoffman and J.B. Kruskal, Integral boundary points of convex polyhedra, *Linear Inequalities and Related Systems*, Princeton Univ. Press, Princeton, 1958, 223-246.

[76] J.E. Hopcroft and R.M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.*, **2**, 1973, 225-231.

[77] M. Jerrum and A. Sinclair, Conductance and the rapid mixing property for Markov Chains: the approximation of the permanent resolved, *Proc. 20th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1988, 235-244.

[78] _____, Approximating the permanent, *SIAM J. Comput.*, **18**, 1989, 1149-1178.

[79] D. Johnson, A catalog of complexity classes, Chapter 2 in: *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, Ed.: J. van Leeuwen, The MIT Press/Elsevier, Cambridge/Amsterdam, 1990, 67-162.

[80] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Proceedings Sixteenth Annual ACM Symposium on Theory of Computing*, ACM, New York, 1984, 302-311.

[81] N. Karmarkar, R. Karp, R. Lipton, L. Lovász and M. Luby, A Monte Carlo algorithm for estimating the permanent, preprint, May, 1988.

[82] R.M. Karp, An introduction to randomized algorithms, International Computer Science Institute Tech. Report TR-90-024, Berkeley, June, 1990.

[83] R.M. Karp and V. Ramachandran, Parallel algorithms for shared-memory machines, Chapter 17 in: *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, Ed.: J. van Leeuwen, The MIT Press/Elsevier, Cambridge/Amsterdam, 1990, 869-942.

[84] R.M. Karp, E. Upfal and A. Wigderson, Finding a maxiumu matching is in random NC, *Proc. 17th Annual ACM Symposium on Theory of Computing* ACM, New York, 1985, 22-32.

[85] R.M. Karp and A. Wigderson, A fast parallel algorithm for the maximal independent set problem, *J. Assoc. Comput. Mach.*, **32**, 1985, 762-773.

[86] P.W. Kasteleyn, The statistics of dimers on a lattice. I., The number of dimer arrangements on a quadratic lattice, *Physica*, **27**, 1961, 1209-1225.

[87] _____, Dimer statistics and phase transitions, *J. Math. Phys.*, **4**, 1963, 287-293.

[88] L.G. Khachian, A polynomial algorithm in linear programming, *Doklady Akad. Nauk SSSR*, **244**, 1979, 1093-1096. (Russian). English translation: *Soviet Math. Doklady*, **20**, 1979, 191-194.

[89] D. Kõnig, Sur un problème de la théorie générale des ensembles et la théorie des graphes, (communication made April 7, 1914, to the Congrès de Philosophie Mathématique), *Revue de Métaphysique et de Morale*, **30**, 1923, 443-449.

[90] _____, Line systems and determinants, *Math. Természs. Ért.*, **33**, 1915, 221-229. (Hungarian)

[91] _____, Über Graphen und ihre Andwendung auf Determinantentheorie und Mengenlehre, *Math. Ann.*, **77**, 1916, 453-465.

[92] _____, Graphok és alkalmazásuk a determinánsok és a halmazok elméletére, *Math. Természs. Ért.*, **34**, 1916, 104-119.

[93] _____, Graphs and matrices, *Mat. Fiz. Lapok*, **38**, 1931, 116-119. (Hungarian)

[94] _____, Über trennende Knotenpunkte in Graphen (nebst Anwendungen auf Determinanten und Matrizen, *Acta Sci. Math. (Szeged)*, **6**, 1933, 155-179.

[95] _____, *Theorie der enlichen und unendlichen Graphen*, Akademischen Verlags-

gesellschaft, Leipzig, 1936. (reprinted: Chelsea, New York, 1950.)

[96] _____, *Theory of finite and infinite Graphs*, Birkhauser, Boston, 1990. (English translation with commentary by W.T. Tutte and a biographical sketch of the author by T. Gallai.)

[97] A. Kotzig, Pravidelna Souvislost Grafov, Vysoka Skola Ekonimicka, Bratislava, 1956.

[98] _____, On the theory of finite graphs with a linear factor I, *Mat.-Fyz. Časopis Slovensk. Akad. Vied*, **9**, 1959, 73-91.

[99] _____, On the theory of finite graphs with a linear factor II, *Mat.-Fyz. Časopis Slovensk. Akad. Vied*, **9**, 1959, 136-159.

[100] _____, On the theory of finite graphs with a linear factor III, *Mat.-Fyz. Časopis Slovensk. Akad. Vied*, **10**, 1960, 205-215.

[101] D. Kozen, U.V. Vazirani and V.V. Vazirani, NC algorithms for comparability graphs, interval graphs, and testing for unique perfect matching, *Foundations of Software Technology and Theoretical Computer Science, (New Delhi, 1985)*, Ed.: S.N. Maheshwari, Lecture Notes in Computer Science Vol. **206**, Springer-Verlag, Berlin, 1985, 496-503.

[102] H.W. Kuhn, The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, **2**, 1955, 83-97.

[103] _____, Variants of the Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, **3**, 1956, 253-258.

[104] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[105] M. Lesk, M.D. Plummer and W.R. Pulleyblank, Equi-matchable graphs, in: *Graph Theory and Combinatorics: Proceedings of the Cambridge Combinatorial Conference in Honour of Paul Erdős*, Ed.: B. Bollobás, Academic Press, London, 1984, 239-254.

[106] G. Lev, Size bounds and parallel algorithms for networks, Univ. Edinburgh Dept. of Comput. Sci. Ph.D. Thesis, CST-8-80, 1980.

[107] C.H.C. Little, An extension of Kasteleyn's method of enumerating the 1-factors of planar graphs, *Combinatorial Mathematics, Proc. Second Australian Conference*, Ed.: D. Holton, Lecture Notes in Math. Vol. **403**, Springer-Verlag, Berlin, 1974, 63-72.

[108] L. Lovász, On the structure of factorizable graphs, *Acta Math. Acad. Sci. Hungar.*, **23**, 1972, 179-195.

[109] _____, On determinants, matchings and random algorithms, *Fundamentals of Computation Theory, FCT '79, (Proc. Conf. Algebraic, Arithmetic and Categorical Methods in Computation Theory, Berlin/Wendisch-Rietz 1979)*, Ed.: L. Budach, Akademia-Verlag, Berlin, 1979, 565-574.

[110] L. Lovász, Tibor Gallai is seventy years old, *Combinatorica*, **2**, 1982, 203-205.

[111] L. Lovász, Matching structure and the matching lattice, *J. Combin. Theory Ser. B*, **43**, 1987, 187-222.

[112] L. Lovász and M.D. Plummer, On bicritical graphs, *Infinite and Finite Sets (Colloq. Keszthely, Hungary, 1973)*, Eds.: A. Hajnal, R. Rado and V.T. Sós, Colloq. Math. Soc. János Bolyai, **10**, North-Holland, Amsterdam, 1975, 1051-1079.

[113] _____, *Matching Theory*, Ann. Discrete Math. **29**, North-Holland, Amsterdam, 1986.

[114] M. Luby, A simple parallel algorithm for the maximal independent set problem, *SIAM J. Comput.*, **15**, 1986. 1036-1053.

[115] _____, Removing randomness in parallel computation without processor penalty, Internat. Comput. Sci. Inst. Tech. Report TR-89-044, Berkeley, July, 1989.

[116] _____, A survey of approximation algorithms for the permanent, *Sequences (Naples/ Positano, 1988)*, Ed.: R.M. Capocelli, Springer-Verlag, New York, 1990, 75-91.

[117] W. Mader, 1-Faktoren von Graphen, *Math. Ann.*, **201**, 1973, 269-282.

[118] _____, Über die Anzahl der 1-Faktoren in 2-fach zusammenhängenden Graphen, *Math. Nachr.*, **74**, 1976, 217-232.

[119] D. Magagnosc, Cuts and decompositions: structure and algorithms, Dartmouth College Dept. of Math. Ph.D. Thesis, 1987.

[120] K. Menger, Zur allgemeinen Kurventheorie, *Fund. Math.*, **10**, 1927, 96-115.

[121] _____, *Kurventheorie*, (unter Mitarbeit von G. Nöbeling), Leipzig und Berlin, 1932.

[122] S. Micali and V.V. Vazirani, An $O(V^{1/2}E)$ algorithm for finding maximum matching in general graphs, *Proc. 21st Annual IEEE Symposium on Foundations of Computer*

Science, IEEE Computer Society Press, New York, 1980, 17-27.

[123] H. Minc, *Permanents*, Encyclopedia of Mathematics and its Applications, **6**, Addison-Wesley, Reading, 1978.

[124] H. Minkowski, *Geometrie der Zahlen (Erste Lieferung)*, Teubner, Leipzig, 1896.

[125] H.M. Mulder, Julius Petersen's Theory of Graphs, 1991, this volume.

[126] K. Mulmuley, U.V. Vazirani and V.V. Vazirani, Matching is as easy as matrix inversion, *Combinatorica*, **7**, 1987, 105-113.

[127] O. Ore, Graphs and matching theorems, *Duke Math. J.*, **22**, 1955, 625-639.

[128] _____, *Theory of Graphs*, American Math. Soc. Colloquium Publications, **38**, American Math. Soc., Providence, 1962.

[129] M.W. Padberg and M.R. Rao, Odd minimum cut-sets and b-matchings, *Mathematics of Operations Research*, **7**, 1982, 67-80.

[130] E.M. Palmer, *Graphical Evolution — An Introduction to the Theory of Random Graphs*, Wiley-Interscience, John Wiley & Sons, New York, 1985.

[131] C.H. Papadimitriou and M. Yannakakis, The complexity of restricted spanning tree problems, *J. Assoc. Comput. Mach.*, **29**, 1982, 285-309.

[132] J. Petersen, Die Theorie der regulären Graphen, *Acta Math.*, **15**, 1891, 193-220.

[133] _____, Sur le théoreme de Tait, *Interméd. Math.*, **5**, 1898, 225-227.

[134] N. Pippenger, On simultaneous resource bounds, *Proc. 20th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society Press, New York, 1979, 307-311.

[135] M.D. Plummer, On n-extendable graphs, *Discrete Math.* **31**, 1980, 201-210.

[136] _____, Matching extension and the genus of a graph, *J. Combin. Theory Ser. B*, **44**, *1988, 329-337.*

[137] _____, A theorem on matchings in the plane, *Graph Theory in Memory of G.A. Dirac*, Ann. Discrete Math. **41**, North-Holland, Amsterdam, 1989, 347-354.

[138] _____, Toughness and matching extension in graphs, *Discrete Math.*, **72**, 1988, 311-320. Reprinted in: *Graph Theory and Applications*, Eds.: J. Akiyama et

al., Ann. Discrete Math., **38**, North-Holland, Amsterdam, 1989, 311-320.

[139]_____, Extending matchings in graphs: a survey, preprint, 1990.

[140]_____, Matching and vertex packing: how "hard" are they?, 1991, in prep.

[141]W.R. Pulleyblank, Facets of matching polyhedra, Univ. of Waterloo Dept. of Combinatorics and Optimization Ph.D. Thesis, 1973.

[142]_____, Polyhedral combinatorics, in: *Mathematical Programming - The State of the Art*, Eds.: A. Bachem, M. Grötschel and B. Korte, Springer-Verlag, Berlin, 1983, 312-345.

[143]_____, Polyhedral Combinatorics, Chapter V in: *Handbooks in Operations Research and Management Science, Vol. 1: Optimization*, Eds.: G.L. Nemhauser, A.H.G. Rinnooy Kan, North-Holland, Amsterdam, 1989, 371-446.

[144]_____, Matchings and extensions, in: *Handbook of Combinatorics*, Eds.: R. Graham, M. Grötschel and L. Lovász, Springer-Verlag, Berlin, 1991, (to appear).

[145]W.R. Pulleyblank and J. Edmonds, Facets of 1-matching polyhedra, in: *Hypergraph Seminar*, Eds.: C. Berge and D. Ray-Chaudhuri, Lecture Notes in Math. Vol. 411, Springer-Verlag, Berlin, 1975, 214-242.

[146]M.O. Rabin and V.V. Vazirani, Maximum matchings in general graphs through randomization, *J. Algorithms*, **10**, 1989, 557-567.

[147]R. Rado, Factorization of even graphs, *Quart. J. Math. Oxford Ser.*, **20**, 1949, 94-104.

[148]A.A. Razborov, Lower bounds on the monotone complexity of some Boolean functions, *Dokl. Akad. Nauk SSSR*, **281**, 1985, 798-801. (in Russian). English translation: *Soviet Math. Dokl.*, **31**, 1985, 354-357.

[149]_____, A lower bound on the monotone network complexity of the logical permanent, *Mat. Zametki*, **37**, 1985, 887-900. (in Russian). English translation: *Math. Notes of the Acad. Sci. USSR*, **37**, 1985, 485-493.

[150]P.F. Reichmeider, *The equivalence of some combinatorial matching theorems*, Polygonal Publ. House, Washington, 1984.

[151]H.J. Ryser, *Combinatorial Mathematics*, The Carus Mathematical Monographs, No. 14, Mathematical Association of America, 1963.

[152] A. Schrijver, Min-max results in combinatorial optimization, in: *Mathematical Programming - The State of the Art*, Eds.: A. Bachem, M. Grötschel and B. Korte, Springer-Verlag, Berlin, 1983, 438-500.

[153] _____, Bounds on permanents, and the number of 1-factors and 1-factorizations of bipartite graphs, *Surveys in Combinatorics*, Ed.: E.K. Lloyd, London Math. Soc. Lecture Note Series Vol. **82**, Cambridge Univ. Press, Canbridge, 1983, 107-134.

[154] _____, *Theory of Linear and Integer Programming*, John Wiley & Sons, Chichester, 1986.

[155] P. Seymour and C. Thomassen, Characterization of even directed graphs, *J. Combin. Theory Ser. B.*, **42**,, 1987, 36-45.

[156] C.E. Shannon, The synthesis of two-terminal switching circuits, *Bell Systems Tech. J.*, **28**, 1949, 59-98.

[157] C. Thomassen, private communication, November, 1990.

[158] W.T. Tutte, The factorization of linear graphs, *J. London Math. Soc.*, **22**, 1947, 107-111.

[159] _____, The factors of graphs, *Canad. J. Math.*, **f**, 1952, 314-328.

[160] _____, A short proof of the factor theorem for finite graphs, *Canad. J. Math.*, **6**, 1954, 347-352.

[161] _____, Spanning subgraphs with specified valencies, *Discrete Math.*, **9**, 1974, 97-108.

[162] _____, Chapter VII in: *Graph Theory*, Vol. 21, The Encyclopedia of Mathematics and its Applications, Addison-Wesley, Menlo Park, 1984, 161-184.

[163] L.G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.*, **8**, 1979, 189-201.

[164] V.V. Vazirani, NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems, *SWAT 88; Proc. First Scandinavian Workshop on Algorithm Theory (Halmstad, July, 1988)*, Eds.: R. Karlson and A. Lingas, Lecture Notes in Computer Science Vol. **318**, Springer-Verlag, Berlin, 1988, 233-242.

[165] _____, NC algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems, *Inform. and Comput.*, **80**, 1989, 152-164.

[166] V.V. Vazirani and M. Yannakakis, Pfaffian orientations, 0/1 permanents and even cycles in directed graphs, *Automata, Languages and Programming, (Tampere, Finland, 1988*, Eds.: T. Lepistö and A. Salomaa, Lecture Notes in Computer Science Vol. **317**, Springer-Verlag, Berlin, 1988, 667-681.

[167] _____, Pfaffian orientations, 0–1 permanents, and even cycles in directed graphs, *Discrete Appl. Math.*, **25**, 1989, 179-190.

[168] B.L. van der Waerden, Aufgabe 45, *Jber. Deutsch. Math.-Verein.*, **35**, 1926, 117.

[169] H. Weyl, Elementãre Theorie der konvexen Polyeder, *Comment. Math. Helv.*, **7**, 1935, 290-306. English translation: The elementary theory of convex polyhedra, in: *Contributions to the Theory of Games*, Eds.: H.W. Kuhn and A.W. Tucker, Princeton University Press, Princeton, 1950, 3-18.

[170] D.R. Woodall, Minimax theorems in graph theory, in: *Selected Topics in Graph Theory*, Eds.: L.W. Beineke and R.J. Wilson, Academic Press, New York, 1978, 237-269.

[171] J. Zaks, On the 1-factors of n-connected graphs, in: *Combinatorial Structures and their Applications*, Eds.: R. Guy et al., Gordon and Breach, New York, 1970, 481-488.

[172] _____, On the 1-factors of n-connected graphs, *J. Combin. Theory Ser. B*, **11**, 1971, 169-180.

[173] F. Barahona, On cuts and matchings in planar graphs, Univ. Bonn Inst. für Ökonometrie und Operations Research Report 88503-OR, 1988. (revised: June, 1990.)

[174] _____, Reducing matching to polynomial size linear programming, Univ. of Waterloo Dept. of Combinatorics and Optimization Research Report CORR 88-51, 1988.