COMMUN. STATIST.-THEORY METH., 19(11), 4315-4329 (1990)

PROBABILITY AND STATISITICS IN THE SERVICE OF COMPUTER SCIENCE: ILLUSTRATIONS USING THE ASSIGNMENT PROBLEM

J. Michael Steele
Wharton School, Department of Statistics
University of Pennsylvania
Philadelphia PA 19104

Key Words and Phrases: assignment problem; simulated annealing; stochastic optimization; linear programming with random costs; statistical mechanics.

ABSTRACT

Recent work on the assignment problem is surveyed with the aim of illustrating the contribution that stochastic thinking can make to problems of interest to computer scientists. The assignment problem is thus examined in connection with the analysis of greedy algorithms, marriage lemmas, linear programming with random costs, randomization based matching, stochastic programming, and statistical mechanics. (The survey is based on the invited presentation given during the "Statistics Days at FSU" in March 1990.)

1. Introduction.

The role of probability and statistics in computer science has been expanding vigorously for almost thirty years. It is no longer possible to survey that role honestly in a single article, so the path taken here is rather to look hard at one concrete problem that illustrates many different parts that probability and statistics can play.

The problem chosen for the task is the *Assignment Problem*. Before formally introducing this problem, it seems appropriate to review its qualifications as an illustrative vehicle. It is simple to state and has been extensively studied, but its most compelling charm for the present pupose is the breadth of topics to which it is naturally connected. The assignment problem allows one to discuss a number of beautiful ideas that tie together topics as diverse as linear programming and statistical mechanics.

A final benefit of studying the assignment problem is that it provides natural illustrations of a basic distinction that is sometimes eclipsed. The probabilistic analysis of algorithms is quite distinct from the analysis of probabilistic algorithms. Despite the polysyllabic overlap, these activities have radically different tools, aims, and assumptions.

2. The Assignment Problem.

Consider the task of assigning n jobs to n machines while minimizing the total processing time required to do the n jobs. We assume the required processing times (or more generally the processing *costs*) are specified by n^2 real numbers c_{ij} where c_{ij} is viewed as the cost of performing job i on machine j. Formally, the *Assignment Problem* is the task of determining a permutation π that solves

$$\min_{\pi} \sum_{i=1}^{n} c_{i\pi(i)}.$$

In the sections that follow we will consider two basic stochastic models for this problem; one is given by choosing the c_{ij} as independent identically distributed random variables, the second is given by choosing the off-diagonal values of c_{ij} as the interpoint distances of a set of n points in $\Re^d$. Each of these models is examined in the contexts of heuristic and optimal algorithms.

3. Simplest Stochastic Model.

No doubt the simplest stochastic model for the assignment problem is given by considering the c_{ij} to be independent random variables with the uniform distribution on $[0,1]$. Walkup (1979) investigated this problem and discovered the striking fact that the expected cost of the minimal assignment is bounded independently of n. More precisely, Walkup showed the cost A_n of minimal assignment satisfies $EA_n \leq 3$.

Shortly we will take up the issue of bounding (or even determining) EA_n, but it is first worth examining some naive approaches to choosing π and estimating EA_n.

3.1 Greedy Assignments.

Greedy algorithms are among the most studied methods in combinatorial optimization, and in many instances the natural greedy algorithms are indeed optimal. The theory of matroids has developed in good measure because it provides a systematic view of situations for which greedy algorithms yield genuine optima.

In the case of the assignment problem, the greedy approach is not particularly successful, but it still merits a quick look since it provides one of the few instances where a complete analysis is easy. Moreover, examining the failures of the greedy algorithms helps to reveal the subtlety behind the the analysis of the optimal solution.

Local Greedy Assignment.

Consider the assignment process that successively examines each job i, $1 \leq i \leq n$, and makes an assignment to the free machine for which c_{ij} is minimal, i.e. $\pi(i)$ is chosen so that $c_{i\pi(i)} = \min\{c_{ij} : j \neq \sigma(k), \text{ for all } k < i\}$. Under the hypothesis that the $\{c_{ij}\}$ are independent and uniformly distributed on $[0,1]$, the i'th assignment equals the minimum

of $n - i + 1$ independent uniformly distributed random variables, so the expected cost of the i'th assignment is exactly $1/(n - i + 2)$. The total expected cost of the assignment is therefore asymptotic to $\log n$.

Global Greedy Assignment.

A less naive heuristic for the assignment problem chooses matches successively out of the set of all possible matches by choosing the pair i and j for which c_{ij} is the least cost among the unmatched pairs. This method looks promising since the expected cost of the first match is just $1/(n^2+1)$, rather than the $1/(n+1)$ of the local greedy assignment. Still, this early success soon falters, and the global greedy heuristic again leads to an expected total cost of order $\log n$.

To see why this is so requires a little work. To develop a recursion via a first-step analysis, we let $a_n(t)$ denote the expected cost of the assignment produced by the global greedy heuristic under the assumption that the cost c_{ij} of the n^2 job-processor pairs are i.i.d. and uniformly distributed on the restricted interval $[t, 1]$. To solve the original problem, we therefore need to determine $a_n(0)$.

If $m_k(t, u)$ denotes the density of the minimum of k^2 random variables with the uniform distribution in $[t, 1]$, it is easy to see that

$$a_n(t) = \int_t^1 u m_n(t, u)du + \int_t^1 a_{n-1}(u)m_n(t, u)du.$$

Next, by scaling we have $a_n(t) = nt + (1 - t)a_n(0)$, so writing a_n for $a_n(0)$ and substituting into the integral recursion, we find

$$\begin{aligned} a_n &= \int_0^1 u m_n(0, u)du + \int_0^1 \{(n-1)u + (1-u)a_{n-1}\}m_n(0, u)du \\ &= \frac{n}{(1+n^2)} + a_{n-1}\frac{n^2}{(1+n^2)}. \end{aligned}$$

This recursion easily shows that a_n is exactly of order $\log n$, a result that goes back to Kurtzburg(1962).

3.2 Methods Based on Marriages.

Hall's matching theorem was christened the "Marriage Lemma" by H. Weyl(1949), and his colorful phrasing of Hall's theorem has become standard: "If a set of n men and n women has the property that any subset of k women knows at least k men, then it is possible to marry the n women to the n men so that each women marries a man she knows." We thus have at hand a sufficient condition for the existence of a perfect matching in a bipartite graph. The sufficient condition of Hall's theorem is also obviously necessary.

There are several ways the Marriage Lemma can be turned toward the assignment problem. Perhaps the simplest idea is to consider all of the pairs (i, j) such that $c_{ij} \leq t$ and then ask when it is possible to extract a perfect matching from the resulting graph. If the graph satisfies the condition of the Marriage Lemma, we obtain an assignment that has cost bounded by nt; thus if we let B_t denote the event that

$$\sum_{1\le j\le n, i\in S} 1(c_{ij} \le t) \ge |S| \text{ for all } S \subset \{1,2,\ldots,n\},$$

then we have the bound

$$EA_n \le nt + nP(B_t^c).$$

This bound can be used to improve on the $O(\log n)$ bound that was obtained by the greedy algorithm, but seems to lack the force to show EA_n is bounded independently of n. The difficulty comes from the requirement that we consider *all* S in our construction of B_t. Fortunately, there is a mild variant of the Marriage Lemma that lets us restrict attention to a smaller collection of subsets. With this new condition in hand one can indeed show the uniform boundedness of EA_n.

Symmetric Marriage Lemma and Checking Boundedness.

It is possible to give a symmetric version of the marriage lemma that has the benefit of avoiding the largest values of $|S|$. The symmetrized version says: "If for each $0 \le k \le \lceil n/2 \rceil$ it happens that each set of k women knows at least k men *and* each set of k men knows at least k women, then there is a matching between the men and the women."

It is not hard to prove the Symmetric Marriage Lemma using Hall's theorem; in fact, it can be obtained with help from almost any of the matching theory tools. For example, one can prove it using an induction after the style of the well-known proof of the marriage lemma due to Halmos and Vaughn(1950), or one can extract it from general results like Menger's theorem or the MaxFlow-MinCut theorem.

To prove EA_n is uniformly bounded we modify our earlier construction to take advantage of the new lemma. Thus, we let C_t denote the event that for all $S \subset \{1,2,\ldots,n\}$ satisfying $|S| \le \lceil n/2 \rceil$ we have the inequalities

$$\sum_{1\le j\le n, i\in S} 1(c_{ij} \le t) \ge |S|$$

and

$$\sum_{1\le i\le n, j\in S} 1(c_{ij} \le t) \ge |S|.$$

If $Z(t,j)$ denotes the number of distinct elements in a sample of size N drawn with replacement from $\{1,2,\ldots,n\}$ where N has the binomial distribution with parameters t and j, then

$$P(C_t^c) \le 2 \sum_{j=1}^{\lceil n/2 \rceil} \binom{n}{j} P(Z(t,j) < j).$$

By choosing $t = 9/n$, it is routine (but tedious) to show that EA_n is bounded by 10. Although one can try to squeeze more out of this argument, say by noting

$$EA_n \le \min_t n\{t + P(C_t^c)\},$$

but the underlying construction has built in constraints that seem likely to keep it from

equalling the best known bounds for EA_n. Consequently, attention will be directed toward other methods.

Walkup's Method.

The preceeding calculation gives us the same qualitative understanding of EA_n that Walkup obtained, but Walkup's method gives a quantitatively superior bound. Partly for this reason, we briefly sketch Walkup's original approach; but, since there are methods that are even more quantitatively precise, the main attraction in Walkup's method now comes from the further light it throws on on the use of marriage lemmas.

The last variant of the Marriage Lemma that we recall says: "If there is a $k \geq 1$ such that each man knows exactly k women and each woman knows exactly k men, then it is possible to marry the men and the women." In other words, any regular bipartite graph contains a perfect matching. To make use of this variant we would like to find a way to construct a regular bipartite graph that carries with it the information in our weighted graph. Such a construction is not presently known, but the idea of examining regular graphs is definitely fruitful. It turns out that we need to look toward regular *directed* graphs.

We first show there is a pleasant way to associate a random variable with each vertex so that these vertex variables can be related usefully to our uniform edge variables. We first note that if independent random variables Y and Z are distributed so that $P(Y \leq \lambda) = P(Z \leq \lambda) = 1 - (1 - \lambda)^{1/2} = F(\lambda)$, $0 \leq \lambda \leq 1$, then $X = \min(Y, Z)$ is uniformly distributed on $[0, 1]$. Now, if we associate with each vertex an independent random variable with distribution F and associate to each edge the minimum of the two associated vertex variables, we obtain a model of a random graph with edge weights that are uniformly distributed. Alas, the new edge variables are no longer independent; nevertheless, the process shows promise.

To get a rigorous approach to A_n under the original independent uniform edge weight model we apparently need to introduce a slightly more complex structure. Thus, for each $1 \leq i, j \leq n$ we consider independent variables $\{Y_{ij}\}$ and $\{Z_{ij}\}$ with the distribution F and use them to build a random *regular directed bipartite digraph* G. We write $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_n\}$ for the vertex set bipartition. We then take (a_i, b_j) to be a directed edge of G if Y_{ij} is among the d smallest elements of $\{Y_{ik} : 1 \leq k \leq n\}$, and finally we take a directed edge (b_i, a_j) if Z_{ij} is among the d smallest of $\{Z_{kj} : 1 \leq k \leq n\}$.

This construction is now relativly easy to analyze. If $Y_{i(k)}$ is the k'th smallest element of $\{Y_{ij} : 1 \leq j \leq n\}$, and $U_{(k)}$ is the k'th smallest of n independent uniformly distributed variables, then $EY_{i(k)} \leq 2EU_{(k)}$ and hence $EY_{i(k)} \leq 2k/(n+1)$.

With these observations in place, we just need a good bound on the probability $p(n, d)$ that a random regular bipartite digraph contains a perfect matching. Thus we have the amusing situation that the last marriage lemma we use is a *random marriage lemma*. By a counting argument and traditional estimations, Walkup(1978) showed:

$$1 - p(n, 2) \leq (5n)^{-1}$$

and

$$1 - p(n,d) \leq (122)^{-1}(d/n)^{(d+1)(d-2)} \quad \text{for } d \geq 3.$$

Given these relations, it becomes an easy calculation to show that expected cost of the optimal assignment is bounded by 3.

Although the analysis given here has always focused on the uniform distribution, this has been for sake of convenience. In Frenk, van Houweninge, and Rinnooy Kan (1982,1986), one finds the Walkup approach to the assignment problem can be developed as well for a broad class of distributions. In fact, even the early papers of Kutzrburg(1962) and Borovkov(1962) considered non-uniformly distributed variables.

So far we have seen how the two greedy algorithms can be bested by estimates based on marriage lemmas of several flavors (basic, symmetric, regular, and di-regular stochastic). This should create some appreciation of the remarkable bound that one obtains by using the tools of linear programming.

3.3 Linear Programming Bounds.

The provence of linear programming is the set of problems that can be formulated in terms of the minimization of a linear function subject to constraints:

$$\begin{aligned} &\text{minimize} \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} \sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1,2,\ldots,m \\ &\qquad\qquad x_j \geq 0, \qquad j = 1,2,\ldots,n. \end{aligned} \tag{LP}$$

The assignment problem is easily cast in this framework. We first consider the problem:

$$\begin{aligned} &\text{minimize} \sum_{i,j} c_{ij} x_{ij} \\ &\text{subject to} \sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1,2,\ldots,n \\ &\qquad\qquad \sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1,2,\ldots,n, \\ &\text{and} \quad x_{ij} \geq 0. \end{aligned} \tag{AP}$$

If we now further require the x_{ij} to be integers, it is clear that the linear programming problem AP is an appropriate reformulation of the assignment problem. Without that extra integrality constraint it is a *priori* conceivable that the optimal solution to AP is not a zero-one incidence vector. It turns out that we are actually in luck and the vertices of the convex polytope determined by the constraints of AP are *always* integers. We will not digress on the details here, but this fact follows from a beautiful (but elementary) result from the theory of polyhedra: If A is the vertex-edge incidence matrix of a bipartite graph

G, then every square submatrix of A has determinant 0, 1, or -1.

Dyer, Frieze, McDiarmid Inequality

What we need is a way to bound the expectation of the value of the solution $z^* = z^*(c_1, c_2, ..., c_n)$ of the linear programming problem LP where the cost coefficients $c_1, c_2, ..., c_n$ are non-negative random variables, but where constraint coefficients a_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$, and constraint levels b_i, $1 \leq i \leq m$ are fixed constants.

The bound on Ez^* is computed in terms of a *fixed* feasible solution $\hat{x}_j$, $1 \leq j \leq n$, where by "fixed" we mean that the $\hat{x}_j$ are determined only by consideration of (a_{ij}) and (b_j). Thus, the vector $(\hat{x}_j)$ is not permitted to depend on the specific realization of the random cost coefficients (c_j).

Theorem (Dyer, Frieze, McDiarmid.) If c_j, $1 \leq j \leq n$, are independent and uniformly distributed on $[0,1]$, and $\hat{x}_j$, $1 \leq j \leq n$ is a fixed feasible solution of LP, then the value z^* of the optimal solution satisfies

$$Ez^* \leq m \max\{\hat{x}_j : 1 \leq j \leq n\}. \qquad DFM$$

Application to the Assignment Problem

Nothing could be simpler than to apply the Dyer, Frieze, McDiarmid inequlaity to the assignment problem AP. We first, note that we have $2n$ nontrivial constraints so $m = 2n$. Further, if we let $x_{ij} = 1/n$ for all $1 \leq i, j \leq n$ then x_{ij} is clearly a feasible solution. Hence, the right hand side of DFM works out to be just 2, and the proof of Karp's bound is complete. The method that underlies the proof of the DFM inequality is extremely general and far from being exhausted. The key idea —"conditioning on a basis" — was evident already in Karp(1987), but the understanding of the method was greatly advanced through the generalizations given in Dyer, Frieze, and McDiarmid(1986) and the related paper McDiarmid(1986).

In order to review the DFM technique it will be useful to have at hand the basics of the simplex algorithm of linear programming. Probabilists and statisicians who are familiar with traditional undergraduate expositions of the simplex algorithm may be relieved to see how succinct a reformulation is possible.

Blitzkrieg Simplex Theory

The simplex method for solving linear programming problems (like LP) can be boiled down to the fact that one can rewrite LP in a form that either suggests a further reformulation — or else renders the solution obvious. It takes only a few lines to give a useful and rigorous development of this claim.

First, we rephrase LP in tidier matrix form, so it now reads

$$\text{minimize } z = \mathbf{cx}$$

$$\text{subject to } \mathbf{Ax} = \mathbf{b}$$

$$\mathbf{x} \geq 0.$$

Here, of course A is an $m \times n$ matrix, x is an n-dimensional column vector, b is an m-

dimensional column vector, and c is an n-dimensional row vector. We will sometimes need to write $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n]$ where $\mathbf{a}_j$ denotes the j'th column of A.

A fundamental construct underlying reformulation is the notion of a *basis*. This is just an index subset $B \subset \{1, 2, ..., n\}$ of cardinality m with the property that the corresponding columns of A are linearly independent. By custom, the variables x_i such that $i \in B$ are called the *basic variables*. We will denote the complement of B by N and call the x_i such that $i \in N$ the *non-basic variables*.

The first step of our reformulation takes us from the equation $\mathbf{Ax} = \mathbf{b}$ to one that expresses the basic variables as a linear combination of the non-basic variables and the constraint variable b. Thus, separating x and A into components consisting of the basic and non-basic variables and collecting these on different sides of the equation, we write

$$\mathbf{A}_B\mathbf{x}_B = \mathbf{b} - \mathbf{A}_N\mathbf{x}_N$$

wher $\mathbf{A}_B$ denotes the matix consisting of the columns $\{\mathbf{a}_i : i \in B\}$. Other subscripted variables are defined analogously.

It is useful to have an expression for $\mathbf{x}_B$; and, since our definition of the basis requires the linear independence of the columns of the square matrix $\mathbf{A}_B$, we can write

$$\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N.$$

Next we express the objective function $z = \mathbf{cx}$ in a form that reflects the division into basic and non-basic variables. Certainly, we can write $z = \mathbf{c}_B\mathbf{x}_B + \mathbf{c}_N\mathbf{x}_N$; but, by using our formula for $\mathbf{x}_B$, we can clear away the basic variables entirely and write the original problem LP in the very useful *dictionary* form:

$$\begin{aligned} \mathbf{x}_B &= \mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N \\ z &= \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{A}_N)\mathbf{x}_N. \end{aligned} \qquad DF$$

We are now in position to see the point of this maneuvering. Suppose we are lucky and all the coefficients of the non-basic variables in the formula for z in DF are non-negative, i.e. suppose that

$$c_j \geq \mathbf{c}_B\mathbf{A}_B^{-1}\mathbf{a}_j \quad \text{for all } j \in N. \qquad OC$$

In that case, the formula for z given by the second row of the dictionary makes it clear that one can minimize z over all choices of $\mathbf{x}_N \geq 0$ by letting $\mathbf{x}_N = 0$. Finally, for $\mathbf{x} = (\mathbf{x}_N, \mathbf{x}_B)$ given by $\mathbf{x}_N = 0$ and $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b}$ to be a genuine optimal solution to the problem LP we only need to check the second part of feasibility $\mathbf{x}_B \geq 0$.

We will call OC the *optimality criterion* for the linear program LP. For our theoretical needs, the key insight behind the simplex method is that it gives an explicit condition for a basis to correspond to an optimal solution.

Naturally, for this criterion to be useful algorithmically, more work is needed. For example, if one of the non-basic variables x_i has a negative coefficient in the second equation

of DF, we need to re-write DF with x_i as a basic variable. Since the basis B has exactly m elements, we therefore need to move some variable x_j from B to N to make room for x_i. The rules for such changing of the basis are part of the fundamentals of linear programming, but they need not concern us. The only additional facts we need concerning linear programming are those that support the probabilistic use of the stopping rule OC.

The first result is a classic existence theorem that tells us there is an optimal basic solution whenever the problem LP is feasible and bounded (i.e. in all the cases that matter) there is an optimal solution that satisfies OC. The second result tells us that in the cases of interest to us the optimal basic solution is unique. More precisely, we call on the following two results:

Basis Existence Theorem. *If a linear program in the form LP has an optimal solution, then it has an optimal basic solution, i.e. a solution of the form* $\mathbf{x} = (\mathbf{x}_N, \mathbf{x}_B)$ *where* B *is a basis and* $\mathbf{x}_N = 0$.

Basis Uniqueness Theorem. *If a linear program in the form LP has an optimal solution, then for any* $\delta > 0$ *there is an* ϵ *with* $|\epsilon| \leq \delta$ *such that the corresponding problem with cost vector* $\mathbf{c}' = \mathbf{c} + \epsilon$ *has a unique optimal basic solution.*

Sketch of the Conditioning Argument

We now have at hand everything needed to give a "one line" sketch of the proof of the DFM inequality. Let $\hat{\mathbf{x}}$ be a fixed feasible solution and write down the conditonal expectation of $\mathbf{c}\hat{\mathbf{x}}$ given that B is a basis. By the optimality criterion OC and linearity, this expectation ends up depending on the calculation of the conditional expectation of a uniformly distributed random variable conditional on its being larger than a given value. This is an easy calculation. Finally, one just does honest arithmetic on the resulting identity, collecting the expectations of interest on one side and then using the crudest of bounds on the rest.

Naturally, his sketch is far from complete, but study of the paper of Dyer, Frieze, and McDiarmid(1986) will show that it is nevertheless honest. The crudeness of the sketch may help to show that a great deal of unexhausted potential remains in Karp's idea of "conditioning on a basis."

4. Randomized Algorithm for Matching.

So far we have discussed how one can analyze the behavior of the objective function of an optimization problem if one imposes a probability model on the inputs. There is another way that probability comes into the theory of algorithms—sometimes it is beneficial to make randomized steps in the course of an algorithm. The best known algorithms with this feature are probably those based on "Simulated Annealing" as initiated by Kilpatrick, Gelatt, and Vecchi(1983). The simulated annealing paradigm is especially intriguing because it is rather easy to implement, and it applies to almost any problem. One drawback to simulated annealing is that it is not easy to analyze, and the analytical results that are available are of a qualitative nature. Typically, we can say that simulated annealing will converge, but we cannot say when.

Saski and Hajek(1988) studied simulated annealing in the context of Eucldean matching, and his analysis provides a rare look at how simulated annealing performs on a problem for which we have good polynomial time algorithms. The conclusion of Hajek's analysis is that simulated annealing does not even lead to a polynomial time algorithm for the matching problem, much less to an algorithm that is competitive with the earlier methods. This analysis provides perhaps the strongest negative result in the theory of simulated annealing.

An interesting contrast to the failings of simulated annealing for minimal matching is that other randomized algorithms for the problem actually provide the fastest known practical methods. One such randomized method due to Mulmulley, Vazirani, and Vazirani is particularly attractive because it is also well suited for implementation on parallel machines, (c.f. Lovász(1989)). The MVV method is a bit too involved to detail here, but the essential idea is to examine what can be learned about matchings by considering certain determinants (or more precisely, Pfaffians) in indeterminate variables. Now, determinants in indeterminate variables would seem to be computationally unattactive since they would seem a *priori* to require time and space of order n!.

There are several tricks that help one get around the apparent difficulty of computing with indeterminate variables. First, it turns out to be possible to take advantage of the fact one can evaluate determinants in "real" variables in time of order only n^3 (where n is the number of rows of the matrix). Second, if we replace the indeterminate variables with independent random uniform $[0,1]$ variables, then with probability one a polynomial in these variables evaluates to zero if and only if the polynomial is identically zero. In other words, as far as the presence or absence of polynomial identities is concerned, independent random uniforms can be used to create surrogates for indeterminate variables.

The last idea is that for N sufficiently large, a random choice from $\{1,2,...,N\}$ can be used to capture the qualitative features of a random uniform on $[0,1]$. The following techinical lemma is the tool one uses to provide a rigorous analysis of algorithms based on these ideas.

4.1 Schwartz Lemma.

The lemma in question has nothing to do with the well known lemma on conformal mappings, but rather is a tool that helps make computational sense out of arguments that might otherwise seem to call on the use of real numbers (a computational taboo). Although we could skip the proof, it is brief and contains a charming induction.

Schwartz' Lemma. Suppose $f(x_1, x_2, \ldots, x_n)$ is a non-zero polynomial of degree not greater than k. If X_i, $1 \le i \le n$ are independent random variables uniformly distributed on the finite set $\{1,2,\ldots,n\}$, then

$$P(f(X_1, X_2, \ldots, X_n) = 0) \le \frac{kn}{N}.$$

Proof. Without loss of generality, we can assume that f can be written as a polynomial in the one variable x_1 but with coefficients that are polynomials in the variables $x_2, x_3, \ldots, x_n$:

$$f(x_1, x_2, \ldots, x_n) = f_0(x_2, x_3, \ldots, x_n) + x_1 f_1(x_2, x_3, \ldots, x_n) + \ldots + x_1^k f_k(x_2, x_3, \ldots, x_n).$$

If $A = \{f(X_1, X_2, \ldots, X_n) = 0\}$ and $B = \{f_i(X_2, X_3, \ldots, X_n) = 0 \text{ for all } 0 \leq i \leq k\}$, then

$$P(A) \leq P(B) + P(A|B^c)$$

$$\leq k(n-1)/N + k/N.$$

where we have applied induction to bound $P(B)$ and the bound on $P(A)$ follows from the fact that when we hold $x_2, x_3, \ldots, x_n$ fixed and vary x_1 then f can have at most k roots. From our bound and induction on n, the lemma is thus proved.

5. Euclidean Matching and Assigning

For any set $\{x_1, x_2, \ldots, x_n\}$ of n points in the Euclidean plane, a *matching* is a collection of $\lfloor n/2 \rfloor$ disjoint pairs of points. By the *weight* of a matching we will denote the sum of the Euclidean distances between the elements of the pairs in the matching. The algorithms for finding a maximal weight matching are among the most studied in the theory of combinatorial optimization. The problem is a formally tractable one (i.e. there are polynomial time algorithms), but the deterministic algorithms for minimal matching are not particularly fast, and practical problems regularly arise that outstrip available computational resources.

The most widely implemented deterministic methods for finding a minimal weight matching have running times of order n^3. There are algorithms that are faster than this. In fact, one can show that minimal matching is of the same theoretical complexity as matrix multplication, so one can perform minimal matching at least as fast as $O(n^{\log_2 7})$ recalling Strassen's fast matrix multiplication method. So far, these newer deterministic methods have not excelled in practice, so probabilstic and heuristic algorithms remain of considerable interest.

Parallel to our analysis of the assignment problem proper, we can give an appealing heuristic for the Euclidean matching problem by proceeding myopically. In this case the resulting *globally greedy* algorithm successively matches the two closest unmatched pairs of points. Even naive implementations of this greedy heuristic are faster than traditional matching algorithms, but by making use of specialized data structures Bentley and Saxe (1980) provided an implementation of the greedy algorithm which has a worst case running time of $0(n^{3/2} \log n)$ which is better than any known exact algorithm for determining a minimal matching.

Let G_n and O_n denote the respective weights of the greedy and optimal matchings of an n-set $\{x_1, x_2, \ldots, x_n\} \subset \Re^2$. There are several known relations between G_n and O_n. In

particular there is a ratio bound due to Reingold and Tarjan (1981), the order of which cannot be improved.

In Papadimitriou (1977), the behavior of O_n was considered for points $\{X_1, X_2, ..., X_n\}$ which were chosen at random from $[0,1]^2$, and it was outlined how the techniques used in Beardwood, Halton, and Hammersley (1959) to study the traveling salesman problem, could be modified to prove

$$O_n/n^{\frac{1}{2}} \to c > 0 \quad \textit{almost surely} \quad \text{as } n \to \infty.$$

Avis, Davis, and Steele(1988) considered the corresponding issues for the greedy algorithm and proved results that imply that for G_n, the cost of the matching obtained by the greedy matching, one has:

$$G_n/n^{\frac{1}{2}} \to c > 0 \quad \textit{almost surely} \quad \text{as } n \to \infty.$$

The results of Avis, Davis, and Steele(1988) actually provide for so-called complete convergence, a mode of convergence that is stronger than convergence with probability one and that is less sensitive to modelling consideration when applied. The methods used in Avis, Davis, and Steele also apply to random variables with general distributions in $\Re^d$. In $\Re^d$ the variables G_n are asymptotic to $n^{(d-1)/d}$.

Other Euclidean Matching Problems

There are several variants of the Euclidean matching problem that have interesting connections with other problems of the theory of algorithms. One of these is the two-sample matching problem studied in Ataji, Komlós, and Tusnády(1983). A second development concerns the connection between bin packing an "up right" matching. The later problem is well discussed in Shor(1986) where one finds a discussion of several matching variants and related literature.

6. Advances via Statisitcal Mechanics.

Simulated annealing had its origin in statistical mechanics, and the design of its earliest applications rested on considerations of physical analogy. While many now regard simulated annealing as just one among many possible stochastic relaxations of the greedy algorithm, the insights that statistical mechanics can provide to the theory of optimization are far from exhausted.

Mézard and Parisi (1987,1988) have in fact made remarkable use of ideas from the theory of Spin Glasses to make important progress on the assignment problem. Although this work proceeds within a framework that is not easily linked with that used here, these authors provide interesting arguments for the limit relation

$$EA_n \to \pi^2/6.$$

They also give a derivation of a corresponding (though somewhat less explicit) limit relation for the Euclidean costs O_n. The explicitness of these results is almost shocking when one considers the energy that has been put into the upperbounds on EA_n that we have reviewed and the corresponding hard work that has gone into proving lower bounds (see Lazarus(1979) and especially Goemans and Kodialam(1989)).

These methods of Mézard and Parisi offer many directions for further study, and they surely suggest that the ideas of statistical mechanics are likely to feature prominently in subsequent work in combinatorial optimization.æ

7. Concluding Remarks.

We began with the analysis of the assignment problem under the uniform model, and we saw how several useful techniques fared in action. After the simple analysis of two greedy algorithms, we saw that a deeper understanding of the matching problem was possible using variations on the Marriage Lemma. Before leaving the uniform model we finally saw that remarkable fresh insights could be won by use of linear programming tools, and our sketch of the proof of the Dyer, Frieze, McDairmid theorem motivated blitzkrieg derivation of the simplex method.

Afterwards, we looked more broadly at matching. In particular, we reviewed a probabilstic algorithm that is solidly practical and examined a sampling of probabilistic results that are avaliable under the Euclidean model. Finally, we looked at what the ideas of statistical mechanics can contribute to matching and the assignment problem. This lead us to consider the method of simulated annealing and the remarkable developments of Mézard and Parisi.

Clearly, more was said here about the service that probability provides to computer science than about the service that statistics provides. To some extent this reflects the influence of the assignment problem as a vehicle of exposition. Nevertheless, something is also revealed about the the three disciplines. While there have been some striking applications of statistics (or more broadly *data analysis*) in computer science, the influence of statistics seems to trail that of probability.

To round out the story just a bit, we recall two illustrative applications of data analysis in computer science. The first is Bentley's use of data analytic considerations in the art of practical code speed-up's, c.f. Bentley (1982). The second is Colin Mallows' masterful use of data analysis in the solution of John Conway's challenge problem, c.f. Mallows (1989). Even briefly noted, these examples serve to recall that real statistics can also make compelling contributions to computer science.

Perhaps the most remembered suggestion made at the FSU 30'th Birthday Party ("Statistics Days at FSU") was Thomas Cover's proposal that such happy events roll across the nation as different departments celebrate historical milestones. While one might question the suitability of birthday gifts on such occasions, a natural choice might be the soon-to-appear monograph *Probability Theory of Combinatorial Optimization* , Steele(1991). This volume treats in greater depth almost all of the topics surveyed here.

ACKNOWLEDGEMENT

This research has be supported in part by the following grants: ARO-DAAL03-89-G-0092.P0001, NSF-DMS-8812868,AFOSR-89-0301.A,and NSA-MDA 904-89-H-2034. I would also like to thank Professors Anjani Jain and Abba Krieger for their comments on and earlier draft of this survey.

BIBLIOGRAPHY

Ajtai, M., Komlós, J., and Tusnády, G. (1984). "On optimal matchings," *Combinatorica*, 4, 259-264.

Avis, D., Davis, B., and Steele, J.M. (1988). "Probabilistic analysis of a greedy heuristic for Euclidean matching," *Probability in the Engineering and Information Sciences*, 2, 143-156.

Beardwood, J., Halton, J.H., and Hammersley, J.M. (1959). "The shortest path through many points," *Proc. Cambridge Philos. Soc.*, 55, 299-327.

Bentley, J.L. (1982). *Writing Efficient Programs*, Prentice-Hall, Inglewood Cliffs, NJ

Bentley, J.L. and Saxe, J.B. (1980). "Decomposable searching problems 1: static to dynamic transformations," *J. Algorithms*, 1, 301-358.

Borovkov, A.A.(1962). "A probabilistic formulation of two economic problems," *Soviet Math.*, 3 , 1403-1406.

Donath, W.E. (1969). "Algorithm and average-value bounds for assignment problems," *IBM J. Res. Develop.*, 13, 380-386.

Dyer, M.E., Frieze, A.M., and McDiarmid, C.J.H. (1986). "On linear programs with random costs," *Math. Programming*, 35.

Frenk, J.B.G., Van Houweninge, M., and Rinnooy Kan, A.H.G. (1986). "Order statistics and the linear assignment problem," Technical Report #8609/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands.

Frenk, J.B.G., Van Houweninge, M.V., and Rinnooy Kan, A.H.G. (1982). "Asymptotic properties of assignment problems," Technical Report #8233/O, Econometric Institute, Erasmus University Rotterdam, The Netherlands.

Goemans, M.X. and Kodialam, M.S. (1989). "A Lower Bound on the Expected Cost of an Optimal Assignment," Technical Report, M.I.T. Operations Research Center, Cambridge, MA.

Hall, P. (1935). "On representation of subsets," *J. London Math. Soc.*, 10, 26-30.

Halmos, P. and Vaughan, H.E. (1950). "The marriage problem," *Amer. J. Math.*, 72, 214-215.

Karp, R.M. (1987). "An upper bound on the expected cost of an optimal assignment," 1-4, in *Discrete Algorithms and Complexity: Proceedings of the Japan-US Joint Seminar*, (D. Johnson, *et al.* eds.), Academic Press, New York, NY.

Kilpatrick, S., Gelatt, C., and Vecchi, M. (1983). "Optimization by simulated annealing," *Science*, 220, 671-680.

Kurtzberg, J.M. (1962). "On approximation methods for the assignment problem," *J. ACM*, 9, 419-439.

Lazarus, A. (1979). "The Assignment Problem with Uniform (0, 1) Cost Matrix," Unpublished B.A. Thesis, Department of Mathematics, Princeton University, Princeton, NJ.

Lovász, L. and Plummer, M.D. (1986). *Matching Theory*, North-Holland Mathematical Studies, New York.

Mallows, C. (1989). " Conway's challenge problem.", Technical report, ATT Bell Laboratories, Murry Hill, NJ.

Mézard, M. and Parisi, G. (1987)."On the Solution of the Random Link Matching Problem", *J. Phyique* ,48,1451-1459.

Mézard, M. and Parisi, G. (1988)."The Euclidean Matching Problem", *J. Phys. France* ,49,2019-2025.

Mézard, M., Parisi, G., and Virasoro,M.A. (1989). *Spinn Glass Theory and Beyond*, Lecture Notes in Physics, 9, World Scientific, Singapore.

McDiarmid, C. (1986). "On the greedy algorithm with random costs," *Mathematical Programming*, 36, 245-255.

Reingold, E.M. and Tarjan R..E. (1981). "On the greedy heuristic for complete matching," *SIAM J. Comput.*, 10 , 676-681.

Papadimitriou, C.H. (1978). "The probabilistic analysis of matching heuristics," *Proc. 15th Annual Conference Comm. Contr. Computing*, Univ. Illinois, Champaign, IL.

Schor, P. W. (1986). " The average case analysis of some on-line algorithms for bin packing," *Combinatorica* , 6 , 2,179-200.

Saski, G.H. and Hajek, B. (1988) "The Time Complexity of Simulated Annealing," *J. Assoc. Comput. Mach.* 35 (2) 387-403. Steele, J.M. (1991). *Probability theory of Combinatorial Optimization*, Wadsworth Advanced Books, Carmel, CA.

Walkup, D.W. (1981). "Matchings in random regular bipartite digraphs," *Discrete Math.* 8, 440-442.

Walkup, D.W. (1979). "On the expected value of a random assignment problem," *SIAM J. on Computing*, 8, 440-442.

Weyl, H. (1949). "Almost periodic invariant vector sets in a metric vector space," *Amer. J. Math.*, 71, 178-205.

Received April 1990; Revised September 1990.

Recommended by P. E. Lin, The Florida State University, Tallahassee, FL.

Refereed by Robert J. Serfling, Johns Hopkins University, Baltimore, MD.