Two Greedy Heuristics for the Weighted Matching Problem

David Avis

School of Computer Science
McGill University

1. Introduction

The problem of finding a weighted matching in a graph has been efficiently solved by Edmonds [7] - [9] and the resulting algorithm is considered to be one of the most elegant computations in the field of combinatorial optimization. The currently best implementations of Gabow [11] and Lawler [13] require $O(n^3)$ operations. Why then would one be interested in heuristic solutions to this problem? There are two main answers to this question. Firstly, the weighted matching algorithm has been used as a subroutine in heuristics for the travelling salesman problem. See, for example [2] and [5]. These heuristics typically run in $O(n^2)$ time except for the matching subroutine which degrades the computation to $O(n^3)$ time. Since we are obtaining an approximate solution to the original problem in any event, a good heuristic solution to the matching problem obtained in $O(n^2)$ time may be preferable. This will allow many heuristic solutions to the original problem to be tried for the same cost as the one heuristic solution using the optimum matching algorithm.

A second reason for studying heuristics for this problem is motivated by the following comment of Bradley [4]. He notes that he knows of no commercial uses of the optimum matching algorithms in the solution of various routing problems, although there are many indications that heuristics are used [3], [14], [15]. In this case, the inherent programming complexity of the optimum algorithms coupled with the general inaccessibility of commercial codes is probably the cause.

In this paper we examine two greedy heuristics with running times of $O(n^2)$ and $O(n^2 \log n)$ for the solution of the weighted matching problem in complete graphs. Section 2 contains the necessary definitions, a description of the heuristics and an analysis of their running times. Section 3 contains an analysis of the average behaviour of the solutions obtained. Bounds on the weight of the expected solution are given for very general distributions for the $O(n^2)$ heuristic and exact values are

This work was supported by McGill University interim research grant 943-81-01.

PROC. 9TH S-E CONF. COMBINATORICS, GRAPH THEORY, AND COMPUTING, pp. 65-76.

obtained for the uniform and exponential distributions. The $O(n^2 \log n)$ heuristic is more complicated and bounds are presented for the expected weight of a solution under the assumption of uniform distribution of edge weights. Section 4 contains an analysis of the worst case performance of the heuristics. A modification is introduced to improve the performance of the $O(n^2)$ heuristic. Under the assumption of non-negative edge weights, it is shown that the worst case bounds on the $O(n^2 \log n)$ heuristic are far superior for the maximization problem than for the minimization problem.

2. The Heuristics

Let $m = \binom{2n}{2}$ and let K_{2n} be the complete graph on 2n vertices with non-negative weights a_{ij} assigned to each of the m edges. A set of edges is called a *perfect matching* in the graph if no two edges share a common vertex. In this paper we will often omit the adjective perfect. The problem is to find a matching of either minimum or maximum weight. Since the applications discussed in section 1 require a minimum weight matching, we will state the heuristics for this case. Obvious modifications will convert them for the maximization problem. We now state two heuristics for finding a minimum weight matching in K_{2n}.

Greedy I

Select a node i at random from the graph. Choose the edge (i,j) of minimum weight adjacent to i and add it to the matching. Delete nodes i and j and all adjacent edges. Repeat until all nodes have been matched.

Greedy II

Select the edge (i,j) of minimum weight from the graph and add it to the matching. Delete nodes i and j and all adjacent edges. Repeat until all nodes have been matched.

Greedy I can easily be implemented to run in $O(n^2)$ time. The naive implementation of Greedy II requires $O(n^3)$ time, which can be reduced to $O(n^2 \log n)$ by first sorting all the edges.

3. Analysis of the Average Performance

In this section we analyse the quality of the solutions obtained by the heuristics when applied to graphs with random edge weights. Indeed let F denote the distribution function of the edge weights and let $X_1, X_2, \ldots, X_m$ denote independent random variables each with distribution F and corresponding to a edge weight in K_{2n}. Let $X_{(1)}, \ldots, X_{(m)}$ denote the order statistics, so that

$$X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(m)} .$$

Finally let F_m be the distribution function of $X_{(1)}$ and let G_m be the distribution function of $X_{(m)}$. The reader wishing more information on this subject is referred to the excellent book by David [6].

We note the following basic relations:

$$F_m(x) = P\{\text{smallest edge weight} \leq x\} = 1 - (1-F(x))^m$$

$$G_m(x) = P\{\text{largest edge weight} \leq x\} = (F(x))^m .$$

Let A_{2n} be the random variable whose values are the weights of matchings produced by Greedy I on K_{2n}, when the edge weights are chosen independently with distribution function F. Let B_{2n} be the corresponding random variable for the maximization problem.

<u>Theorem 3.1</u> $$E(A_{2n}) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} x \, F_{2i-1}(dx) \tag{1}$$

$$E(B_{2n}) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} x \, G_{2i-1}(dx) . \tag{2}$$

<u>Proof</u>: We consider the minimization problem, the maximization problem is similar. Greedy I picks a node at random and selects the minimum weight adjacent edge. This weight has distribution F_{2n-1} at iteration 1 and $F_{2n-2i+1}$ in general at iteration i, since two edges are deleted from each unmatched node at each iteration. The formula 1 follows. □

We consider two special cases: when the edge weights are chosen according to uniform and exponential distributions. The *harmonic series* will be used in the analysis, the n^{th} term of which is given by

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}.$$

The approximate size of H_n is given by (See Knuth [12])

$$H_n = \ln n + \gamma + 0\left(\frac{1}{n}\right)$$

where $\gamma = .57721 \ldots$ is Euler's constant.

Corollary 3.1 If $F(x) = x$, for $0 \le x \le 1$, then $E(A_{2n}) = \frac{H_n}{2} = \frac{1}{2}(\ell_n\, n + \gamma + 0\left(\frac{1}{n}\right)$, and $E(B_{2n}) = n - E(A_{2n})$.

Proof: A routine calculation shows that

$$\int_0^1 x\, F_k(dx) = \frac{1}{k+1} \quad \text{and} \quad \int_o^1 x\, G_k(dx) = \frac{k}{k+1}.$$

Hence $E(A_{2n}) = \sum_{i=1}^{n} \frac{1}{2i} = \frac{H_n}{2}$.

$$E(B_{2n}) = \sum_{i=1}^{n} \frac{(2i-1)}{2i} = n - \sum_{i=1}^{n} \frac{1}{2i} = n - E(A_{2n}). \qquad \square$$

Corollary 3.2 If $F(x) = 1 - e^{-x}$, then

$$E(A_{2n}) = H_{2n-1} - \frac{1}{2} H_n = \frac{1}{2} \ln n + \frac{3\gamma}{2} + \ln 2 + 0\left(\frac{1}{n}\right).$$

$$E(B_{2n}) = \sum_{i=1}^{n} H_{2i-1} = n[\ln(2n-1) + \gamma - 1] + 0(\ln n).$$

Proof: A simple calculation shows that

$$F_k(x) = 1 - e^{-kx}, \quad G_k(x) = (1 - e^{-x})^k$$

and hence

$$\int_0^\infty x\, F_k(dx) = \frac{1}{k}, \quad \int_0^\infty x\, G_k\, dx = \sum_{i=1}^{k} \frac{1}{i} = H_k.$$

Thus, applying theorem 3.1,

$$E(A_{2n}) = \sum_{i=1}^{n} \frac{1}{2i-1} = H_{2n-1} - \frac{1}{2} H_{n-1} = \ln(2n-1) - \frac{1}{2} \ln n + 3\frac{\gamma}{2} + 0\left(\frac{1}{n}\right)$$

$$= \frac{1}{2} \ln n + 3\frac{\gamma}{2} + \ln 2 + 0\left(\frac{1}{n}\right).$$

$$E(B_{2n}) = \sum_{i=1}^{n} H_{2i-1} = \frac{1}{2}\ (2n+1)\, H_{2n-1} - 2n+1 - \frac{H_{n-1}}{2}$$

$$= n\ (\ln\ (2n-1) + \gamma - 1) + 0(\ln n). \qquad \square$$

The analysis for Greedy II is more complicated. This is due to the fact that after each edge is chosen for the matching, the distribution of the remaining edge weights changes. We will restrict ourselves to a uniform distribution of edge weights.

Let C_{2n} be the random variable whose values are the weights of matchings produced by Greedy II on K_{2n}, when the edge weights are chosen independently with distribution F.

<u>Theorem 3.2</u> If $F(x) = x$, for x in the range $0 \le x \le 1$, then

$$\frac{1}{4} \ln n - \frac{1}{4} \le E(C_{2n+2}) \le \frac{1}{2} \ln (n+1) + 1 .$$

We begin with two combinatorial lemmas. Consider the sequence defined by

$$b_{n+1} = \frac{(b_n + 1)a_n}{1 + a_n} , \qquad n = 0,1,2,\ldots$$

where $b_0 = 0$ and $a_n = \binom{2n}{2}$.

<u>Lemma 3.1</u> For every integer k, if $n \ge e^{4k+1}$ then $b_n < n-k$.

<u>Proof</u>: First observe that if for some n_0, $b_{n_0} < n_0 - k$, then $b_n < n-k$ for all $n \ge n_0$. Indeed, in this case

$$b_{n_0+1} < \frac{(n_0 - k+1)\ (1+a_n) - (n_0-k+1)}{1+a_n} < (n_0+1) - k .$$

Now suppose that $b_n \ge n-k$ for $1 \le n < e^{4k+1}$. Then for n in this range

$$b_{n+1} = b_n + 1 - \frac{1+b_n}{1+a_n} < b_n + 1 - \frac{n-k+1}{1+(n+1)(2n+1)} .$$

Iterating,

$$b_{n+1} < n+1 - \sum_{i=1}^{n} \frac{i-k+1}{(i+1)(2i+1)+1} < n+1 - \frac{1}{2} \sum_{i=1}^{n} \frac{1}{i+1} + \sum_{i=1}^{n} \frac{k}{(i+1)^2} .$$

$$\le n+1 - \frac{1}{2} (\ln n -1) + k(1 - \frac{1}{n}) < n - k$$

for $n = e^{4k+1}$. □

<u>Lemma 3.2</u> For every integer $n = 1,2,\ldots.$

$$b_{n+1} \ge n - \frac{1}{2} \ln n.$$

<u>Proof</u>: From the definition of b_n,

$$b_2 = {}^9/_7 \ge 1 - \frac{1}{2} \ln 1 .$$

Proceeding inductively,

$$b_{n+2} \ge n - \frac{1}{2} \ln n + 1 - \frac{n - {}^1/_2 \ln n + 1}{(n+2)(2n+3) + 1}$$

$$\ge (n+1) - \frac{1}{2} (\ln n + \frac{1}{n+1}) + \frac{\frac{1}{2} \ln n + 1}{(n+2)(2n+3)+1}$$

$$\geq (n+1) - \frac{1}{2} \ln (n+1) . \qquad \square$$

Proof of theorem. Let $f_{2n}(y)$ be a function defined for y in the range [0,1] whose value is the expected weight of the matching found by Greedy II on K_{2n} , when the edge weights are chosen independently and uniformly in the range [y, 1]. Thus $E(C_{2n}) = f_{2n}(0)$. A simple computation shows that the density function g for the minimum of a independent and uniformly distributed random variables on [y,1] is given by $g(x) = \frac{a(1-x)^{a-1}}{(1-y)^a}$, when $0 \leq y < 1$.

At iteration one, Greedy II picks the minimum of $\binom{2n}{2}$ random variables uniformly distributed on [0,1]. Suppose that this edge has weight x. Then the remaining edge weights are uniformly and independently distributed on [x,1]. Those considerations lead to the recursion:

$$f_{2n+2}(y) = (1-y)^{-a_n} \int_y^1 [x + f_{2n}(x)] a_n (1-x)^{a_n - 1} dx , \quad 0 \leq y < 1$$
$$n = 0,1,2,...$$

where $f_0(y) = 0$ and $a_n = \binom{2n+2}{2}$.

We are interested in determining tight bounds for $f_{2n+2}(y)$. To this end we begin by showing that, for suitable constants b_n and c_n ,

$$f_{2n}(y) = b_n y + c_n . \tag{3}$$

Indeed, assume inductively that (2) holds. We note that the mean of the distribution given by g(x) is

$$(1 - y)^{-a} \int_y^1 x \, a \, (1-x)^{a-1} dx = \frac{1 + ay}{1+a} .$$

Hence,

$$f_{2n+2}(y) = (1-y)^{-a_n} \int_y^1 (c_n+(b_n+1)x) \, a_n \, (1-x)^{a_n-1} dx$$

$$= c_n + (b_n+1) \frac{1 + a_n y}{1 + a_n} ,$$

$$b_{n+1} = \frac{(b_n+1)a_n}{1 + a_n} \quad \text{and} \quad c_{n+1} = \frac{(a_n + 1) c_n + b_n + 1}{1 + a_n} .$$

Observe that $b_{n+1} + c_{n+1} = b_n + c_n + 1 = ... = n + 1 + b_0 + c_0$

$= n + 1$.

This can also be seen by noting that $b_n + c_n = f_{2n}(1)$, which is just the expected size of a matching given all edges have weight one, which is obviously n .

Therefore we obtain the formula

$$c_{n+1} = n+1 - \frac{(b_n+1)a_n}{1 + a_n} .$$

Applying lemma 3.1, for any integer k, setting $n = e^{4k+1}$ yields:

$$f_{2n+2}(0) = c_{n+1} > n+1 - \frac{(n-k+1)a_n}{1+a_n} = k + \frac{n-k+1}{1+an} \geq \frac{\ell n 4}{4} - \frac{1}{4} .$$

For the upper bound, we employ lemma 3.2 to obtain:

$$f_{2n+2}(0) = c_{n+1} \leq (n+1) - \frac{(n - 1/2 \ \ell n(n-1)a_n}{1 + a_n}$$

$$\leq 1 + \frac{1}{2} \ell n (n-1) + \frac{n + 1/2 \ \ell n \ (n-1)}{1 + a_n} \leq 1 + \frac{1}{2} \ell n \ (n-1) + \frac{1}{2n}$$

$$\leq 1 + \frac{1}{2} \ell n \ (n+1) . \qquad \square$$

4. Worst Case Analysis

Ideally, a good heuristic will produce solutions that are guaranteed to be within a constant factor of the optimum solution. For the weighted matching problem, however, this is likely to be a difficult task. Indeed, such a heuristic would have to solve the unweighted perfect matching problem for graphs. Here one is given an arbitrary graph G on 2n nodes and is asked to find a perfect matching. We can convert this problem into a weighted matching problem by embedding G into K_{2n} by assigning weights of one to edges of G and some large positive constant M to the other edges. Now unless the heuristic finds the perfect matching, the weight of the solution found will be at least M/n times greater than the optimal solution. Actually both heuristics can find a "pessimum", or worst possible solution, as figure 4.1 shows.

We now give a modification to Greedy I to guarantee a solution that in worst case is not much more than the average weight of a matching. By the latter we mean the average weight of the set of all perfect matchings. In K_{2n}, this is easily seen to be n times the average edge weight.

Largest Node Sum Rule

1. For each node i, compute the sum w_i of the weights of all adjacent edges.
2. Scan the nodes in order of decreasing node sum.

The idea is to try to locate at an early stage, nodes that are adjacent to heavily weighted edges, to reduce the possibility of having to use these edges at later iterations. Let A^*_{2n} denote the weight of the matching found by Greedy I using the largest node sum rule, when applied to K_{2n} with edge weights a_{ij}. Let $\bar{a}$ be the average edge weight, so that $n\bar{a}$ is the average weight of a matching.

Theorem 4.1

$$A^*_{2n} \leq 2n\bar{a}\ (H_{2n-1} - \tfrac{1}{2} H_{n-1}) = n\bar{a}\ (\ln n + 3\gamma + 2\ln 2) + O(\bar{a}) .$$

Proof: Assume that the nodes have been numbered so that $w_1 \geq w_2 \geq \ldots \geq w_{2n}$. Then,

$$A^*_{2n} \leq \sum_{i=1}^{n} \frac{w_i}{2n-2i+1}, \tag{4}$$

since the i^{th} node picked can have node sum at most w_i and Greedy I picks the minimum weight of the $2n - 2i + 1$ adjacent edges. For a given average edge weight $\bar{a}$, the right hand side of (4) is maximized when $w_1 = w_2 = \ldots = w_n = (2n - 1)\ \bar{a}$ and $w_{n+1} = w_{n+2} = \ldots = w_{2n} = 0$.

Therefore we obtain

$$A^*_{2n} \leq w_1 \sum_{i=1}^{n} \frac{1}{2i-1} \leq 2n\bar{a}\ (H_{2n-1} - \tfrac{1}{2} H_{n-1}),$$

and the theorem follows. □

It is clear that the largest node sum rule can be implemented in $O(n^2)$ operations.

We conclude this section with the somewhat surprising observation that Greedy II has a reasonably good worst case bound for the problem of finding a *maximum* weight matching in K_{2n} with non-negative edge weights. Let M_{2n} be the weight of the matching found by Greedy II and let M^*_{2n} be the weight of the optimal solution.

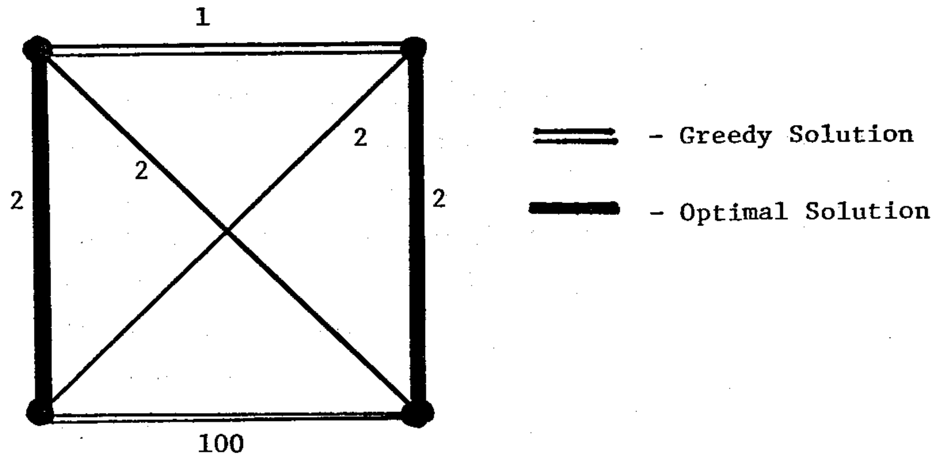

Figure 4.1

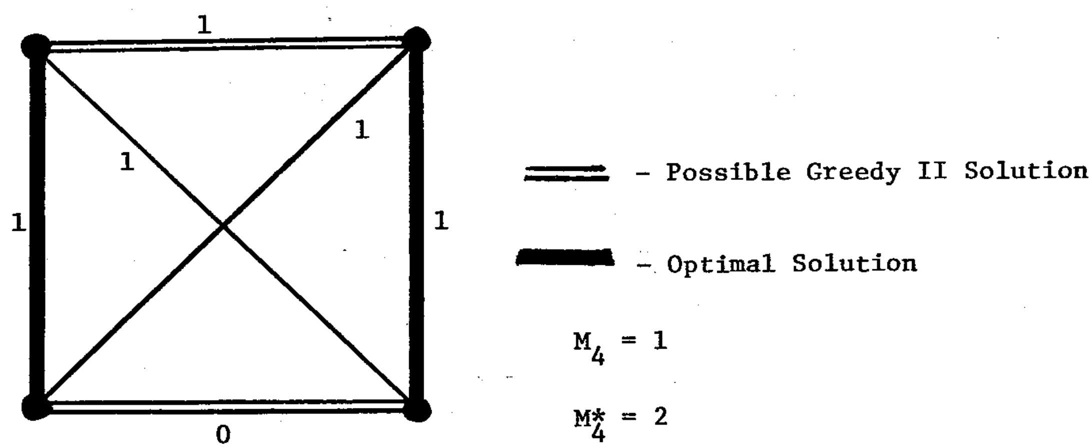

Figure 4.2

Theorem 4.2 $$M_{2n} \geq \frac{1}{2} M^*_{2n} .$$

Proof: Let x be the weight of the first edge (i,j) that is selected by Greedy II, so that x is in fact an edge of maximum weight in K_{2n} . Now when (i,j) and all incident edges are deleted, at most two edges of the optimal matching may be removed. Further, the sum of their weights cannot exceed 2x. The other n-2 or more edges of the optimal matching are candidates for selection at the next iteration of Greedy II. The argument may be repeated for each of the first $^n/2$ iterations of Greedy II. Since all edge weights are non-negative, the theorem is proved. □

The assymmetry introduced by the assumption of non-negative edge weights makes the bound possible for maximization problem, whereas we have seen that no such bound exists for the minimization problem. Figure 4.2 shows an example of when $M_4 = \frac{1}{2} M^*_4$.

5. Conclusions and Extensions

This work was motivated by the need to find a "good" matching in a weighted graph in $O(n^2)$ operations. Selim Akl [2] points out that local improvement of a matching is possible within this time bound. He calls a matching 2-optimal if for every two matching edges (i,j) and (k,ℓ) ,

$$a_{ij} + a_{k\ell} \leq \min (a_{ik} + a_{j\ell} , a_{i\ell} + a_{jk}) .$$

This is the matching equivalent of a similar notion that has been used in heuristics for the travelling salesman problem for some time. See, for example, Lin [17].

If two edges do not satisfy this condition, the appropriate interchange is made. 2-optimality may be tested in $O(n^2)$ operations and may be implemented as a second 'phase' on the matching obtained by a greedy heuristic.

Empirical studies of how these procedures work as subroutines for travelling salesman heuristics is currently being conducted. These results will be reported in [2].

In conclusion, we mention a very interesting paper of Angluin and Valiant [16]. In it they describe and analyze a heuristic for finding a perfect matching in an arbitrary unweighted graph. This heuristic runs with probability tending to one in $O(n \log n)$ time on a random graph and finds a solution with probability $1 - O(n^{-\alpha})$, where α is a positive constant. This analysis is based on the theory of matchings in random graphs developed by Erdös and Renyi [10]. The heuristic is not based on the "greedy" principle, but rather builds up a matching using a random selection procedure and a form of local search to improve on the partial matchings.

Acknowledgement

The author is grateful to Selim Akl for bringing these problems to his attention, and for many interesting discussions.

References

[1] Aho, A., Hopcroft, J. and Ullman, J., *The Design and Analysis of Computer Algorithms*, Addison Wesley, 1975.

[2] Akl, S., "A Statistical Analysis of Various Aspects of the Travelling Salesman Problem," Ph.D. Thesis, McGill University (in preparation).

[3] Beltrani, E. and Bodin, L.D., "Networks and Vehicle Routing for Municipal Waste Collection," *Networks 4*, 65-94, 1974.

[4] Bradley, G., "Survey of Deterministic Networks," *AIIE Transactions*, 222-234, September 1975.

[5] Christofides, N., "Worst Case Analysis of a New Heuristic for the Travelling Salesman Problem,"Man. Sci. Research Report No. 388, Carnegie Mellon University, February 1976.

[6] David, H., *Order Statistics*, Wiley & Sons, 1970.

[7] Edmonds, J., "Paths, Trees and Flowers," *Can. J. Math.*, *17*, 449-467, 1965.

[8] Edmonds, J., "Maximum Matching and a Polyhedron with 0,1 Vertices", *JRNBS*, 696. 35-40, 1965.

[9] Edmonds, J. and Johnson, E.L., "Matching, Euler Tours and the Chinese Postman", *JACM*, 192, 248-264, 1972.

[10] Erdös, P. and Renyi, A., "On the Existence of a Factor of Degree One of a Connected Random Graph," *Acta Math. Sc. Hung.*, *17*, 359-368, 1966.

[11] Gabow, H., "An Efficient Implementation of Edmond's Maximum Matching Algorithm," Tech. Rept. 31, Stanford University Computer Science Department, June 1972.

[12] Knuth, D., *The Art of Computer Programming*, Vol. 1, Addison Wesley, 1967.

[13] Lawler, E., *Combinatorial Optimization*, Holt, Rinehart and Winston, 1977.

[14] Marks, D. and Stricker, R., "Routing for Public Service Vehicles," *ASCE Journal of Urban Planning and Development Division*, 165-178, December 1971.

[15] Wyskida, R. and Gupta, J., "IE's Improve City's Solid Waste Collection," *Journal of Industrial Engineering*, *4*, 12-15, June 1972.

[16] Angluin, D. and Valiant, L., "Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings," *9th Annual ACM Symposium on Theory of Computing*, May 1977.

[17] Lin, S., "Computer Solutions of the Travelling Salesman Problem," *Bell Syst. Tech. J.*, *44*, 2245-2269, 1965.