

A FAST, NEAR-OPTIMAL HEURISTIC FOR THE
WORK STATION ASSIGNMENT PROBLEM

Candace Arai Yano

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117, USA

Technical Report 84-26

Revised August 1985

A FAST, NEAR-OPTIMAL HEURISTIC FOR THE
WORK STATION ASSIGNMENT PROBLEM

Candace Arai Yano
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, Michigan 48109-2117

August 1985

A FAST, NEAR-OPTIMAL HEURISTIC FOR THE
WORK STATION ASSIGNMENT PROBLEM

ABSTRACT

The problem of assigning clerks to work stations arises in the context of mail order sales operations, airline reservation facilities, utility service offices and other similar operations. The use of large numbers of part-time employees and long hours of operation can make the assignment problem difficult to solve optimally using manual procedures. Minimization of work stations also provides significant savings over a "one work station per person" policy under these conditions.

This paper describes both a fast, near-optimal heuristic and an optimal procedure to solve the assignment problem, and presents computational results which indicate that practical problems can be solved optimally, even on a personal computer. We also discuss how these techniques can be used to study the effect of work group size on facilities requirements.

1. Introduction

Many service and direct marketing operations require staff answering calls to be seated at work stations equipped with a telephone and possibly a video display terminal for reference and/or for input. If the staff work full-time shifts and the number of hours of operation is less than two full-time shifts, each person must be assigned his or her own work station in order to avoid conflicting assignments and "musical chairs" reassignments in the middle of a shift. However, if a large percentage of the work force is employed part-time, or if the hours of operation are long, several people may be assigned to a given seating location without a schedule conflict. Economies may be obtained by minimizing the number of positions required.

In situations where a large workforce can be divided into smaller work groups, it is also useful to study the effects of work group size on the ratio of positions to employees so as to optimize the trade-off between group size and number of positions. In general, it would be expected that as work group size increases, the ratio of positions to employees decreases, since the problem becomes less constrained. However, there may be some practical limitations on managerial span of control or on the number of people in the work area. Thus, the critical trade-off here involves facilities cost versus human considerations.

This work was motivated by a facility of a major communications firm where both the facility planning and work station assignment problems needed to be addressed.

The solution to the assignment problem can be used in conjunction with any standard workforce scheduling system. It is assumed here that each person has been assigned work tours for a given day or week (i.e., the workforce scheduling problem has been solved). The problem now is to assign these people

to positions. In Section 2 we formulate the problem. Section 3 describes an optimal solution procedure and a fast heuristic. An example problem is solved in Section 4, computational results are presented in Section 5, and we conclude with a summary in Section 6.

2. Problem Formulation

We make the following assumptions:

- (1) The work week can be divided into distinct days in such a way that there is no feasible work tour which spans parts of two consecutive days, and such that clerks can work only one tour a day.
- (2) The work day can be divided into an integral number of time increments of equal length in such a way that all work tours begin and end at a multiple of this time increment.

The first condition assures separability of the problem by work day. In so doing, it limits the size of the problem to the number of individuals working on a given day. This condition is usually satisfied in existing operations. The second condition is not necessary for the solution of the seating assignment problem but is assumed in most workforce scheduling algorithms and is almost always satisfied in manual scheduling systems.

The objective of this problem is to assign people to positions in such a way that the minimum number of positions is used.

$$\text{Let } x_{ipt} = \begin{cases} 1 & \text{if person } i \text{ occupies position } p \text{ and works} \\ & \text{during time period } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ipt} = \begin{cases} 1 & \text{if person } i \text{ is assigned to position } p \\ 0 & \text{otherwise} \end{cases}$$

The objective is to minimize the number of positions required or

$$\min_t \max_p \sum_i \sum_t x_{ipt}$$

Obtaining the value of the objective function is trivial: it is equal to the maximum number of people working at any point in time. The important decisions involve assignment of people to these positions.

There is one basic requirement: each position can be occupied by only one person at a given time. This can be expressed as the constraint

$$\sum_i x_{ipt} \leq 1 \quad \forall p, t$$

One additional constraint is desirable. Each person must be able to remain at the same position for his or her entire work tour on a given day. Changing positions in the middle of a work tour is undesirable both administratively and from the perspective of maintaining productivity. The constraint is expressed as:

$$(y_{ip} - 1) \sum_t x_{ipt} = 0, \quad \forall i, p$$

and $\sum_p \sum_t x_{ipt} = \text{tour length for person } i, \quad \forall i$

Some values may be prespecified as follows:

If person i does not work in time period t , set

$$x_{ipt} = 0, \quad \forall p.$$

If person i must be assigned to position p_i , set

$$x_{ipt} = 0, \quad \forall t, p \neq p_i$$

and $x_{ipt} = 1, t \in \{\text{work tour for person } i\}, p = p_i,$

This formulation yields a large nonlinear integer programming problem. In this form the problem is computationally intractable for most real problems.

The nonlinear constraints and the binary nature of the decision variables make this problem extremely difficult to solve using what would appear to be

applicable techniques. For instance, the problem is a constrained two-dimensional bin-packing problem. The height of the bin is the number of periods in a day, and the width corresponds to the number of positions. The "cartons" to be packed are one unit wide, and for each person, the height is the number of periods in the work tour. One special constraint is that each carton must be placed in the correct horizontal (time) location in the bin.

Unfortunately, the two-dimensional bin-packing problem is np-complete (Fowler, et al., 1981). The problem, however, can be formulated as a graph coloring problem and can be solved in this manner.

Let each person working on the day in question be represented as a node. Connect nodes i and j if person i 's work tour coincides at some point in time with that of person j (i.e., persons i and j cannot be assigned to the same position). The problem then is to color the nodes with the smallest number of colors in such a way that adjacent (connected) nodes do not have the same color. This is known as the minimum coloring or stable set coloring problem. All persons associated with a node of a particular color can be assigned to the same work station.

In its most general form, the minimum coloring problem is np-complete. However, the graphical representation of this problem is an interval graph.

DEFINITION: An undirected graph is called an interval graph if its vertices can be put in one-to-one correspondence with a set of intervals \mathcal{I} of a linearly ordered set (e.g., the real line) such that two vertices are connected if and only if their corresponding intervals have a non-empty intersection. (Columbic, 1980).

In the graphical representation of the problem, each node corresponds to a continuous interval of time on a real time axis. The nodes are connected if the work tours overlap, i.e., if there is a non-empty intersection. Thus, this

problem can be represented using an interval graph.

DEFINITION: Triangulated (also called chordal) graphs have the property that every circuit of length $k > 3$ has a chord.

We next establish that interval graphs are triangulated, that triangulated graphs are perfect graphs, and that perfect graphs have a perfect vertex elimination scheme. The perfect elimination scheme is needed to an efficient graph coloring algorithm which is described later.

THEOREM 1: An interval graph satisfies the triangulated graph property.

PROOF: See Hajos (1958).

Let G denote a graph. Then we have the following definition:

DEFINITION: A graph is called perfect if it has the property

$$(P_1) \quad \omega(G_A) = \omega(G) \text{ for all } A$$

where $\omega(G)$ = clique number of G : the size of the largest complete subgraph of G

$\chi(G)$ = chromatic number of G : the fewest number of colors needed to properly color the vertices of G

and G_A = subgraph induced by nodes in A

This definition is not comprehensive, but is adequate for the purposes of this paper.

THEOREM 2: Triangulated graphs satisfy the perfect property.

PROOF: See Berge (1960).

This implies that it is necessary only to find $\omega(G)$ to insure that the (minimum) chromatic number is identified.

Denote the adjacency set of a vertex v as $\text{Adj}(v)$.

DEFINITION: An ordering $\sigma = [v_1, v_2, \dots, v_n]$ of the vertices is a perfect vertex elimination scheme if each set

$$X_i = \{v_j \in \text{Adj}(v_i), j > i\}$$

is complete (i.e., there exists an edge between every pair of nodes).

DEFINITION: A vertex x of G is called simplicial if $\text{Adj}(x)$ induces a complete subgraph of G .

THEOREM 3: Every triangulated graph has a perfect elimination scheme, and any simplicial vertex can start a perfect scheme.

PROOF: See Golumbic (1980).

We have now established that the graphical representation of this problem has a perfect vertex elimination scheme.

THEOREM 4: Let d be the maximum degree and n the number of vertices in G . Then an upper bound on the computational complexity of a perfect elimination scheme for a chordal graph G is

$$nd(n+1)(d+1)/4.$$

PROOF: See Gavril (1972).

Clearly $d \leq n-1$ so $nd(n+1)(d+1)/4 < n^4/4$, so the computational complexity in the worst case is $O(n^4)$. The construction of a perfect elimination scheme is the most computationally burdensome portion of the procedure. However, the work required at each step involves checking for adjacency of two nodes, which is a very simple procedure given a node incidence matrix. We would expect that for $n < 100$, microcomputer implementation in a few minutes is feasible. Most problems fall into the range of $n < 100$ in one work group. However, telephone operator facilities in major cities may have $n > 200$, so a fast heuristic may be

necessary for such problems.

3. Optimal Procedure and a Fast Heuristic

The first step in solving this problem involves transforming output from a workforce scheduling algorithm into a graphical representation of the seating assignment problem. This involves fairly complex data management but is straightforward conceptually. We will not dwell on this process here.

Once a graphical representation has been constructed, the problem can be solved by first constructing a perfect vertex elimination scheme. Select a node v such that $\text{Adj}(v)$ is completely connected and eliminate it from the graph. Continue this process until only one node remains. The nodes are colored in the reverse order of the perfect elimination scheme.

A minimum coloring can now be obtained by coloring the relabeled nodes sequentially (node with smallest label first), using the first feasible color. Colors are sequenced by the order in which each is first used. A new color is used only when colors already utilized are infeasible. This minimum coloring algorithm is due to Gavril (1972).

Each set of nodes with the same color represents the set of people assigned to one specific position. The relabeled nodes must be translated into their original designations in order to make appropriate assignments.

The fast heuristic is quite simple. Sequence the workers in order of work tour start time, breaking ties arbitrarily. Starting with the workers with earliest start times, assign each to the lowest indexed position available. This heuristic will be compared with the optimal procedure in Section 5. First, we illustrate the optimal procedure and the effect of work group size on facilities requirements by way of an example.

4. Example

Assume that on given work day, the following work tours have been assigned:

<u>Person</u>	<u>Work tour</u> <u>(including lunch</u> <u>and breaks)</u>
A	0800-1200
B	0800-1700
C	0900-1200
D	0900-1500
E	1000-1800
F	1200-1600
G	1300-1700
H	1500-1800

The graphical representation appears in Figure 1. Note that there is one node representing each person. Nodes are connected whenever the corresponding people cannot be assigned to the same work station because of overlapping work tour assignments. Therefore, node A is connected to nodes B, C, D, and E, but not to F, G, and H.

One possible perfect elimination scheme is {H,G,F,A,B,C,D,E}. Returning nodes in reverse order, the nodes are relabeled, as illustrated in Figure 2.

The nodes can be colored as follows:

- Step 1. Node 1 colored with color #1
- Step 2. Node 2 colored with color #2
- Step 3. Node 3 colored with color #3
- Step 4. Node 4 colored with color #4
- Step 5. Node 5 colored with color #5

- Step 6. Node 6 can be colored with any color except #1, #2, and #4. The first feasible color is #3; hence node 6 is colored with color #3.
- Step 7. Node 7 can be colored with any color except #1, #2, #3 and #4. Therefore, node 7 is colored with color #5.
- Step 8. Node 8 can be colored with any color except #1, #3, #4 and #5. Therefore, node 8 is colored with color #2.

Using this information, the following positions are assigned:

<u>Position</u>	<u>Node numbers</u>	<u>People</u>
1	1	E
2	2,8	D,H
3	3,6	C,F
4	4	B
5	5,7	A,G

Now suppose that we were to split the original 8 person group into two groups of 4 - A, C, E and G, and B, D, F and H. The first group would require 3 positions (with G and A or G and C sharing one position), and the second group would require 3 positions (with H and D sharing a position). Thus, the total facilities required have increased, as we would expect. The individuals can be assigned to groups in such a way that a maximum of 5 positions is required - but generally only if the original seating assignment is solved first. Furthermore, it is probably not desirable to change group assignments for administrative reasons.

5. Computational Results

There are two objectives of our computational study. First, we wanted to determine whether problems of reasonable size can be solved optimally on a microcomputer even though the complexity of the perfect elimination scheme is $O(n^4)$. Second, we wanted to compare the heuristic and the optimal procedure both in terms of solution quality and computation times.

We randomly generated 50 different work schedules each for 20, 50, and 80 person situations. We assumed that the facility operates 16 hours per day (which is typical in real applications). For 25 of the work schedules, work tour lengths were randomly selected from 4 to 9 hours (integer values only), and for the others, the work tour lengths could vary from 3 to 6 hours (again, integers only). The start of each work tour was randomly generated from among feasible start times for that tour length. For instance, a work tour six hours in length could start from opening of the facility until ten hours hence. The programs were coded in FORTRAN77 and run on an IBM PC/XT.

Means and standard deviations of the solution times for each set of 25 problems are reported in Table 1. It is evident that solution times increase rapidly with the number of persons, but it is still feasible to determine optimal solutions for problems with 80 persons.

TABLE 1

The heuristic generated colorings which were alternate optima in 149 of the 150 cases. In the one non-optimal solution, only one additional position was required for $n=80$. The mean and standard deviation of computation times are reported in Table 2 for each set of 25 problems. The heuristic required only a fraction of the time required for the optimal solution, and the times did not increase rapidly as n increased. Thus, it appears that this simple procedure,

which can be implemented manually, can be expected to provide good (and frequently optimal) results.

TABLE 2

6. Summary

This paper details the graph theoretic solution of, and a fast heuristic for a typical seating assignment problem with many potential applications. Computational experience indicates that problems with up to 80 people can be solved optimally on a personal computer. The heuristic found alternate optima in over 99% of the problems in considerably less computation time. We have discussed briefly how the techniques may be used to aid in facilities design and capacity planning. As more business is transacted by telephone, efficient utilization of facilities to house and support these service operations will become increasingly important.

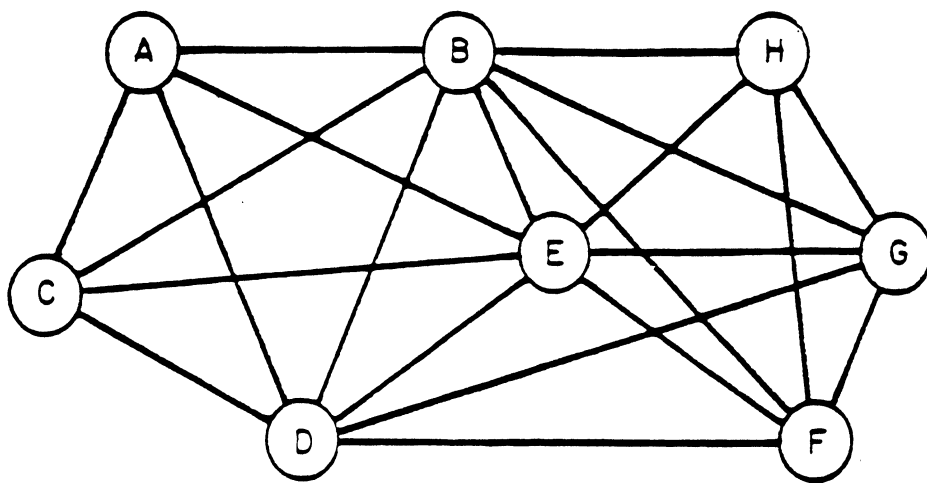


FIGURE 1

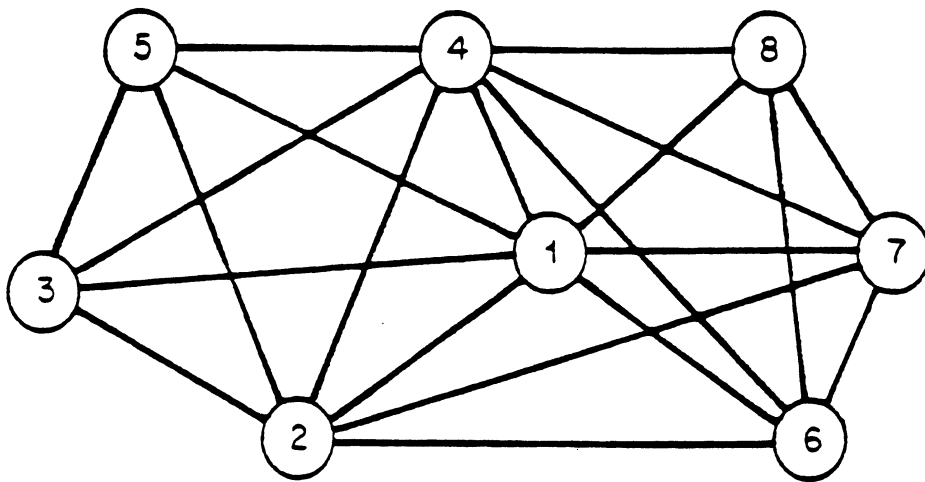


FIGURE 2

TABLE 1

Computation Time Statistics for the Optimal Procedure
(in seconds)

<u>n</u>	Work Tour Length Range (hours)			
	<u>3 - 6</u>		<u>4 - 9</u>	
	<u>x</u>	<u>s</u>	<u>x</u>	<u>s</u>
20	4.62	1.29	1.40	0.37
50	178.52	31.51	194.50	34.87
80	1184.59	228.75	352.14	54.72

TABLE 2

Computation Time Statistics for the Heuristic Procedure
(in seconds)

<u>n</u>	Work Tour Length Range (hours)			
	<u>3 - 6</u>		<u>4 - 9</u>	
	<u>x</u>	<u>s</u>	<u>x</u>	<u>s</u>
20	0.19	0.029	0.20	0.028
50	1.08	0.033	1.17	0.045
80	2.70	0.038	2.93	0.051

REFERENCES

- Berge, Claude (1960). "Les problemes de colorations en theories des graphs." Publ. Inst. Statist. Univ. Paris 9, 123-160.
- Fowler, R.J., M.S., Paterson, and S.L. Tanimoto (1981). "Optimal Packing and Covering in the Plane are NP-Complete", Info. Process. Letters 12, 133-137.
- Gavril, Fanica (1972). "Algorithms for minimum coloring, maximum clique, minimum coloring by cliques and maximum independent set of a chordal graph." SIAM J. Comput. 1, 180-187.
- Columbic, Martin C. Algorithmic Graph Theory and Perfect Graphs, New York: Academic Press, 1980.
- Hajos, G. (1957). "Uber eine Art von Graphen," Internat. Math. Nachr., p.65.

Captions for Figures:

Figure 1: Graphical Representation of Workers and Seating Conflicts

Figure 2: Perfect Elimination Scheme