1-1-1983

The scheduling of employee vacations using general matching of quotas-to-quotas a derivative of the marriage problem.

Charlotte A. Lent

THE SCHEDULING OF EMPLOYEE VACATIONS USING GENERAL MATCHING OF QUOTAS-TO-QUOTAS: A DERIVATIVE OF THE MARRIAGE PROBLEM

by

Charlotte A. Lent

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1983

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

Dec 7, 1983
Date

Professor in Charge

Chairman of the Department
of Industrial Engineering

ACKNOWLEDGEMENTS

The author is grateful to her advisors, Dr. John W. Adams and Dr. George R. Wilson of Lehigh University, for their guidance and assistance.

Many thanks are extended to David Pepper of the Lehigh University Computing Center consulting staff, for his countless hours of debugging assistance.

Acknowledgement is also due to the author's supervisor, Mr. Thomas T. Vrba, for his support and cooperation, which made a graduate degree attainable.

The author also wishes to thank Jill Gallagher for her excellent skill, cooperation, and patience in typing this thesis.

Lastly, the author is especially grateful to Mr. Fred Campbell for his patience, understanding, and support during not only this thesis preparation, but during the entire graduate education experience.

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

A matching algorithm is developed to assign employee vacations subject to constraints associated with employee seniority and preference, as well as, minimum staffing requirements throughout the company and within each job category. The specific constraints were posed by an industrial firm. The basic marriage problem, originally credited to D. Gale and L. S. Shapley, is expanded to match many "proposals" to many "acceptances". The algorithm was shown to be employee optimal, with a stable matching.

Computational aspects of the problem are discussed, and sensitivity to parameters is investigated. The matching algorithm requires a moderate amount of computer time. Large employee populations drive up the requirement for an almost prohibitive amount of computer memory. The computational aspects of the matching algorithm are relatively insensitive to changes in parameters other than population size.

I. INTRODUCTION

The process of assigning individual vacations within allocation constraints becomes tedious and complicated as the size of the workforce grows. Manual efforts often lead to errors. Disgruntled employees may file grievances, resulting in large costs to the company. The obvious solution is to automate the company's assignment procedure. Previously, the problem encountered when trying to automate was the consideration of the specific assignment conditions. The constraints to be considered are:

(1) The number of weeks of vacation allotted to each employee, depends on the employee's length of service to the company.

(2) Preferences for given weeks of vacations, vary with the individual.

(3) Each job must be covered by a minimum number of workers during any given week.

(4) Priority is given to the employee with the most seniority.

(5) A minimum number of employees throughout the company must be present during each workweek.

The problem which is the subject of this paper is to determine an algorithmic approach to scheduling individual employee vacations while considering the employees' preferences, priorities associated with seniority, and the company's and work-group's minimum work requirements. The approach begins with ranking employees in seniority order. Each employee is requested to submit, in preference order, a list of desired weeks. The number of weeks on the list should exceed the employee's allotment. The employees agree to accept as their assigned vacation, any week on their submitted preference list. From this point a matching algorithm can be used to match employee ranked preferences with company ranked preferences (minimum numbers present within seniority ranking). The solution would consist of an individual schedule for each employee in which he/she was granted those weeks of preference which did not conflict with company/job minimum requirements.

Chapter II presents an overview of approaches to this constrained vacation assignment problem. Chapter III contains the algorithm for optimally solving this problem. Chapter IV discusses the results of this algorithm and Chapter V presents the conclusions. Areas for further study are shown in Chapter VI.

II. SURVEY OF THE LITERATURE

Initial investigations into the development of a vacation assignment algorithm indicate that linear programming can be used in conjunction with the transportation algorithm [1]. A weighting scheme is required in order to consider seniority and job category. The assignment of weights is a delicate procedure, and is open to subjective interpretation. Additionally, the transshipment problem [2], a special form of the transportation problem, is costly to execute.

When the vacation assignment problem is studied, often the words "match" and "rank" are used. The company is obligated to "match" the employees' wishes with the available weeks. The lists of vacations are made according to seniority "ranking." These key words prompt the analyst to investigate matching algorithms. D. Gale and L.S. Shapley [3] proposed a matching which does not include any weighting schemes. This once novel approach is now referred to as the marriage problem. Broadly stated, equal numbers of men and women are members of a social grouping. Each member independently ranks his/her preferences for a marriage partner, from the members of the opposite sex. Once all lists are complete, each man "proposes" to the first woman on his list. After all initial proposals are made, each woman orders her suitors according to her previously ranked preference list. She rejects all but the highest ranking suitor. The prospective bridegroom, however, understands that he may be rejected in later rounds.

Those men rejected in the first round, now propose to their second preferences. Each woman again ranks the suitors according to her preference list. The suitor from the first round is ranked along with second round hopefuls. All but the top suitor on this new list, are rejected. These newly rejected men propose to their next choices. The women then rank and reject the suitors. Eventually, all women will have received a proposal. As soon as all women have only one man waiting on her list, the "courting" is over and proposals are accepted as final.

The question of stability is easily answered. Suppose Bob and Sue are not engaged to be married. If Bob prefers Sue to his fiancée, then Sue must have rejected Bob in favor of a more preferred man. It is obvious that Sue prefers her intended husband to Bob. Therefore, the marriage is stable.

The Gale-Shapley method for matching is defined for equal length ranked lists. The vacation problem violates this definition when employment exceeds fifty-two (52) persons. Additional deviations occur when employees are allowed more than one week of vacation. The Gale-Shapely method has been extended to match "one" to "many," e.g., students with educational institutions, football players with professional teams. Roth [4] discusses this method and concludes:

(1) the difference between the one-to-one matching problem and the one-to-many problem is of no consequence.

(2) there will always be at least one stable outcome.

The algorithm developed in Chapter III of this paper extends the general "one-to-many" matching to a "many-to-many" matching.

III. THE ALGORITHM

This chapter will develop an algorithm to schedule employee requested vacations subject to the constraints of seniority preference, minimum attendance requirements imposed by the company, and the employee's work-group. The word "he" represents both male and female employees.

The algorithm essentially is the Gale-Shapely method for matching. The differences basically are that employees (boys) "propose" to more than one week (girl) at a time. Each employee initially bids for the number of weeks he is allotted. These bids come from a ranked list submitted by the employees. The weeks (girls) make a list of employees bidding for that week. The bid lists are then ranked by seniority. If the minimum constraint is violated, the employees ranked below the cut-off are rejected. For the specific case studied for this problem, minimum constraints existed for the work-group (job category) and also for the company as a whole. This additional constraint merely imposed the additional creation of ranking within, and rejecting from a bid list. If an employee has been accepted onto the bid list for a given week for his work-group, he may be rejected due to minimum constraints at the company level. Seniority is the ranking criteria. This additional constraint does not affect the overall matching procedure. It simply requires more time and variables.

Once the employees have all bid for their vacation allowances, the second stage begins. Those employees which were rejected from one or more weeks, say r, now "propose" to the next r weeks on their ranked preference list. The weeks receiving "proposals" rank the bidders within the current list, and reject any employee falling below the minimum criteria for the week/work-group.

Proceeding in the same manner, those employees rejected, rebid their next preference(s). The weeks, again, rank and reject, if necessary.

This procedure of bidding, ranking, rejecting continues until each employee has received his allowance of vacation time. In theory, this procedure works. In reality, constraints need to be imposed on the number of weeks on an employee's preference list. Ideally, the employee should rank all 52 weeks. This imposes an immense burden on computer resources and the employee.

The procedure outlined above, if programmed directly, grows rapidly beyond machine capability as the number of employees, the amount of vacation allowed, and the size of the individual preference list grow. To overcome this machine constraint, certain reconstructions of the algorithm were necessary. First and foremost, it was assumed that employees would be considered for vacation bidding, in a ranked seniority order. This merely imposes a sort routine before the algorithm actually begins. Once the employees are ranked, the algorithm will bid one employee at a time. Because the employees are bidding in seniority order, they automatically are placed in a ranked position. Subsequent rejection/rebidding compares first to the minimum criteria and then to rank within the lists. This saves numerous searching and additional ranking. The steps of the algorithm are outlined below.

STEPS OF THE ALGORITHM

1. Initial Bidding Process.

    (a) Begin with the most senior employee.

    (b) Begin with the employee's first, most preferred choice.

    (c) Initialize at zero, the total number of weeks awarded to the bidding employee.

    (d) Check the minimum work-group constraint for the week being considered.

    (e) Is the work-group constraint violated?

        Yes (f)
        No (h)

    (f) Go to employee's next preference.

    (g) Has the employee bid for his entire initial allotment of vacation?

        Yes (n)
        No (d)

(h) Check the minimum company constraint for the week being considered.

(i) Is the company constraint violated?

Yes (f)
No (j)

(j) Place the bidding employee on the list for the week being considered.

(k) Increase by 1, the bidding employee's award count, the company weekly count, and the work-group weekly count.

(l) Move the rejected employee's pointer to the next preference on his list.

(m) Has the bidding employee bid for his total allotment of vacation?

Yes (n)
No (d)

(n) Go to the next employee.

(o) Has the least senior employee completed the initial bidding process?

Yes (p)
No (b)

2. Initial Rebidding Process.

(p) Begin with the most senior employee.

(q) Has the bidding employee received his full allotment of vacation?

Yes (nn)
No (r)

(r) Check the minimum work-group constraint for the week which is the bidding employee's next preference.

(s) Is the work-group constraint violated?

Yes (t)
No (dd)

(t) Consider only the employees within the bidding employee's work-group. Is the bidding employee more senior than the last employee on the list for the week being considered?

Yes (w)
No (u)

(u) Reject the bidding employee's bid.

(v) Has the bidding employee exceeded the number of possible bids?

Yes (nn)
No (r)

(w) Increase by 1, the number of weeks granted to the bidding employee.

(x) Increase by 1, the number of accepted bids for the work-group of the bidding employee.

(y) Decrease by 1, the rejected employee's allowance.

(z) Decrease by 1, the number of accepted bids for the work-group of the rejected employee.

(aa) Move the rejected employee's pointer to the next preference on his list.

(bb) Drop the rejected employee from the company list for the week being considered. Add the bidding employee, in ranked order, to the company list.

(cc) Go to (q).

(dd) Check the company minimum constraint for the week being considered.

(ee) Is the company constraint violated?
Yes (ff)
No (jj)

(ff) Compare the seniority of the bidding employee with the seniority of the last employee on the list.

(gg) Is the bidding employee more senior than the last employee on the list for the week being considered?

Yes (w)
No (hh)

(hh) Reject the bidding employee's bid.

(ii) Has the bidding employee exceeded the number of possible bids?

Yes (nn)
No (r)

(jj) Place the bidding employee, in ranked order, on the list for the week being considered.

(kk) Increase by 1, the number of weeks granted to the bidding employee, the weekly count, and the work-group weekly count.

(ll) Has the bidding employee reached his full allotment?

Yes (nn)
No (mm)

(mm) Has the bidding employee exceeded the number of possible bids?

Yes (nn)
No (r)

(nn) Go to the next employee.

(oo) Has the least senior employee completed this bidding round?

Yes (pp)
No (q)

3. Subsequent Rebidding Rounds.

(pp) Begin with the most senior employee.

(qq) Has the bidding employee received his full allotment of vacation?

Yes (tt)
No (rr)

(rr) Has the bidding employee exceeded the number of possible bids?

Yes (tt)
No (ss)

(ss) Repeat Step 2.

(tt) Go to next employee.

(uu) Has the least senior employee completed this round?

Yes (vv)
No (qq)

(vv) Stop. Print lists of accepted bids.

The size of the preference list will be determined by the overall dimensions of the problem. If the problem constrains the preference list size to the point where an employee receives less than his allowed number of weeks, it was assumed, for this study, that the employee would be contacted, and asked for additional preferences. It is possible, however, to construct an algorithm to assign weeks from those where the minimum constraints have not been met.

The algorithm has been programmed in FORTRAN V to run on a Cyber 70 computer. Initial preference lists were created and ranked on a separate file. This file was input to the main program containing the algorithm. The algorithm constructs for each week a list of employees that have been granted that week for their vacation. Each employee, however, needs his own output list. To accomplish this, the weekly lists are output to a file, which is used as input to another program which will search the weekly lists and write the vacation schedule for each employee.

The results of a sample scheduling procedure can be found in Appendix A. This sample was constrained to 25 employees, three (3) work-groups, a company limit of two (2) and a work-group limit of one (1) for demonstration purposes.

IV. ANALYSIS OF THE ALGORITHM

The following criteria have been used to analyze the algorithm:

1. The optimality of the solution.

2. The impact of the dimension of the parameters, i.e. number of employees, number of work-groups, length of preference lists, length of weekly lists for the company and work-group.

It can be shown that the set of matchings of employees to weeks is employee optimal. If there exists a stable matching that grants an employee his preferred week of vacation, it can be termed week "possible" for an employee. Assume that at some point in the procedure, no employee has been denied a bid for vacation by a "possible" week. At the same time, suppose a week N rejects John from the current bid list upon receiving a bid from Tom, a more senior employee. It can be shown that N is "impossible" for John. We know Tom desires week N more than the other weeks remaining on his list. These remaining weeks, it is assumed, are "impossible" for Tom. If a hypothetical matching existed in which John were granted vacation for week N, and Tom is granted vacation for one of the other weeks remaining on his preference list, then Tom is granted a less desirable week of vacation. This conceived matching is unstable since week N and Tom could benefit by "unmatching." Week N utilizes the company's seniority list. The company benefits because the vacation list will not violate seniority ranking, thus avoiding a grievance situation.

In conclusion, the algorithm rejects employees only from those weeks where they cannot possibly be granted that time under a stable matching. The final matching is therefore employee optimal.

The size of the workforce is the most influential parameter required for the matching algorithm described in this study. A specific problem proposed by Ferrara and Marshall [2] involved 1,300 employees, a maximum of 14 non-consecutive individual weeks of vacation, and 35 work-groups. There are five arrays that are dimensioned for 1,300. One of these arrays is two dimensional. The second dimension being the number of weeks submitted as preferences. With a large workforce, the weekly vacation lists are large. Another two-dimensional array of size approaching 1,300 by 52 is therefore required. This specific problem requires 320K words.

The system time also varies with the size of the problem. Table IV-1 demonstrates the impact of varying the company-wide limit for each week of vacation. As the company limit becomes smaller and more restrictive, more rejections occur. Subsequent rebidding requires more system time. The example of limiting the company vacations to 200, demonstrated this increased time. The limit was set to a constant for each week for convenience. The program, however, allows the limit for each week to vary.

IMPACT OF THE WEEKLY COMPANY LIMIT SIZE

Maximum Number of Employees Granted Vacation	System Seconds Used	Maximum CM Words Used
1100	43.3	320000
1000	41.0	304000
300	28.7	200000
200	40.4	164000

Table IV-1

Although the effects of only company limits were studied, it is proposed that similar results would occur if work-group limits were varied.

By varying the number of work-groups, only system seconds changed. The change was small as shown in Table IV-2.

IMPACT OF THE NUMBER OF WORK-GROUPS

Maximum Number of Employees Granted Vacation	Number of Work-Groups	System Seconds Used	Maximum CM Words Used
300	35	28.7	200000
300	40	27.9	200000

Table IV-2

The size of the individual employee's preference list may affect the amount of computer words required. This is because each employee has a list. Each incremental change in the length of the list is multiplied by the number of employees. The system seconds used, however, will rise in a constrained environment where the employee's are bidding far down on their lists.

The search program used to print the individual lists of final vacations for each employee, used considerably more words and system time than the actual matching algorithm. Table IV-3 demonstrates the case of the weekly lists constrained to 300.

SYSTEM RESULTS OF THE PRINT PROCEDURE

Total Number of Employees	System Seconds Used	Maximum CM Words Used
1300	179.6	250000

Table IV-3

The conclusions are that the matching algorithm is employee optimal and that the size of the employee population is a limiting factor for large workforces.

V. CONCLUSIONS

The algorithm, although simple by design, requires "bookkeeping" that limits its use when large workforces are being scheduled. Because computing time and words are inter-related, the size of the problem that can be accommodated will vary with the actual preferences of the employees. Large numbers of rejections will require more time.

The "many-to-many" matching technique was shown to be an efficient method for automating the scheduling of vacations for a large workforce subject to company and seniority restrictions. The actual input technique would be determined by the company and the peripherals available. The employee optimal algorithm provides a vacation schedule for each employee in which there are no other combinations that will provide him a more preferred schedule.

IV. AREAS FOR FURTHER STUDY

The basic marriage problem was expanded from "one-to-one" matching to "many-to-many". Stability and "man" optimality were maintained. An assumption used when developing the "many-to-many" matching, was that the employees did not have reservations for blocks of consecutive weeks of vacations. It is proposed that stability and optimality would still be maintained if blocks of weeks were allowed. The affects on system time and the requirement for additional arrays would be prohibitive with today's computer equipment. Time saving programming techniques may help.

Further study should be in the area of the length of the preference lists for each employee. Since it is impractical to ask each employee to rank all 52 weeks of the year, a procedure should be developed to determine the optimal number of weeks each employee should submit as preferences. It is proposed that this optimal number is related to the employee's allowance, the company/work-group constraints and behavior patterns.

The company/work-group minimum requirements were constant in this study. As these constraints vary with time, it is proposed that stability and optimality will be maintained. The impact on system usage will probably vary widely.

Although not directly associated with the algorithm, other means to sort and print individual employee lists should be investigated.

BIBLIOGRAPHY

1. Adams, J. W., IE435 Mathematics of Operations Research class discussion, April 14, 1982, Lehigh University.

2. Ferrara, R., and Marshall, S., "Vacation Scheduling," IE154 Senior Project, Lehigh University, Fall 1981.

3. Gale, D. and Shapley, L. S., "College Admissions and the Stability of Marriage," Amer. Math. Monthly, Vol. 69 (1962), pp. 9 - 14.

4. Roth, A.E., "The Economics of Matching: Stability and Incentives," Mathematics of Operations Research, Vol. 7 (November 1982), pp. 617 - 628.

APPENDIX A

AN EXAMPLE OF THE VACATION SCHEDULING USING "MANY TO MANY" MATCHING

	EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
1.	J. Dickerson	June 20, 1938	1	6	6	2	Assigned
						13	Assigned
						3	Assigned
						26	Assigned
						30	Assigned
						36	Assigned
						22	Not required
2.	R. Johnson	Sept. 1, 1943	3	6	6	12	Assigned
						13	Assigned
						23	Assigned
						38	Assigned
						15	Assigned
						46	Assigned
						50	Not required
3.	G. Peters	Jan. 5, 1945	1	6	6	7	Assigned
						42	Assigned
						38	Assigned
						12	Assigned
						27	Assigned
						45	Assigned
						23	Not required

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
4. H. Santos	Dec. 8, 1948	2	6	6	31	Assigned
					43	Assigned
					39	Assigned
					32	Assigned
					22	Assigned
					20	Assigned
					50	Not required
5. F. Green	Aug. 7, 1950	3	6	6	40	Assigned
					32	Assigned
					34	Assigned
					30	Assigned
					2	Assigned
					48	Assigned
					18	Not required
6. P. Jackson	April 7, 1953	1	5	5	22	Assigned
					23	Assigned
					37	Assigned
					46	Assigned
					20	Assigned
					21	Not required
					49	Not required

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
7. O. Rizzo	Dec. 9, 1954	2	5	5	27	Assigned
					51	Assigned
					5	Assigned
					14	Assigned
					43	Rejected: group
					36	Assigned
					48	Not required
8. P. Hewitt	June 7, 1956	1	5	5	25	Assigned
					51	Assigned
					38	Rejected: group
					14	Assigned
					19	Assigned
					43	Assigned
					35	Not required
9. H. Whitacre	Mar. 18, 1958	1	5	5	31	Assigned
					28	Assigned
					49	Assigned
					39	Assigned
					20	Rejected: group
					46	Rejected: group
					44	Assigned

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
10. A. Segal	Oct. 25, 1960	2	5	5	3	Assigned
					27	Rejected: group
					16	Assigned
					20	Rejected: group
					33	Assigned
					34	Assigned
					6	Assigned
11. F. Fowler	Nov. 25, 1962	1	5	2	33	Assigned
					7	Rejected: group
					19	Rejected: group
					38	Rejected: group
					4	Assigned
					32	Rejected: co.
					37	Rejected: group
12. T. Ashford	Nov. 30, 1962	3	5	5	18	Assigned
					52	Assigned
					33	Rejected: co.
					23	Rejected: group
					41	Assigned
					44	Assigned
					19	Assigned

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
13. W. Carroll	April 17, 1964	2	4	3	12	Rejected: co.
					26	Assigned
					21	Assigned
					6	Rejected: group
					46	Rejected: co.
					28	Assigned
					27	Rejected: group
14. F. Harrison	July 21, 1968	1	3	3	49	Rejected: group
					41	Assigned
					18	Assigned
					31	Rejected: group
					21	Assigned
					16	Not required
					20	Not required
15. B. Guida	June 17, 1971	1	3	2	44	Rejected: group
					5	Assigned
					15	Assigned
					14	Rejected: group
					12	Rejected: group
					21	Rejected: group
					20	Rejected: group

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
16. S. Sampson	Feb. 20, 1973	1	2	2	19	Rejected: group
					26	Rejected: group
					49	Rejected: group
					50	Assigned
					40	Assigned
					31	Not required
					37	Not required
17. F. Kilpatrick	Jan. 15, 1975	3	2	2	26	Rejected: co.
					22	Rejected: co.
					47	Assigned
					6	Assigned
					2	Not required
					14	Not required
					45	Not required
18. W. Mitchell	May 18, 1977	3	2	2	50	Assigned
					24	Assigned
					43	Not required
					11	Not required
					32	Not required
					28	Not required
					40	Not required

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
19. A. Ryan	May 20, 1980	3	1	0	14	Rejected: co.
					21	Rejected: co.
					18	Rejected: group
					51	Rejected: co.
					19	Rejected: group
					48	Rejected: group
					24	Rejected: group
20. R. Lee	June 18, 1981	3	1	1	32	Rejected: group
					44	Rejected: group
					30	Rejected: group
					7	Assigned
					51	Not required
					36	Not required
					13	Not required
21. R. O'Reilly	Jan. 17, 1983	2	11	51	51	Rejected: group
					25	Assigned
					38	Not required
					15	Not required
					48	Not required
					29	Not required
					21	Not required

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
22. T. Walton	March 20, 1983	2	1	0	21	Rejected: group
					28	Rejected: group
					32	Rejected: group
					39	Rejected: group
					51	Rejected: group
					14	Rejected: group
					30	Rejected: co.
23. D. Matthews	Apr. 20, 1983	3	1	1	44	Rejected: group
					8	Assigned
					48	Not required
					39	Not required
					30	Not required
					22	Not required
					18	Not required
24. D. Rudolph	Sept. 15, 1983	3	1	1	18	Rejected: group
					12	Rejected: group
					17	Assigned
					35	Not required
					37	Not required
					42	Not required
					15	Not required

EMPLOYEE	SERVICE COMP DATE	WORK GROUP	NUMBER OF WEEKS ALLOWED	NUMBER OF WEEKS ASSIGNED	WEEKS DESIRED IN PREFERENCE ORDER	STATUS
25. W. Spence	Nov. 19, 1983	3	1	1	27	Rejected: co.
					16	Assigned
					41	Not required
					32	Not required
					45	Not required
					20	Not required
					37	Not required

co. - company limit violation
group - work-group limit violation

VITA

PERSONAL HISTORY

Name:	Charlotte A. Lent
Date of Birth:	July 26, 1952
Place of Birth:	Fredericksburg, Virginia
Parents:	Robert F. Lent and Mary R. Lent

EDUCATIONAL BACKGROUND

Haddon Heights High School Haddon Heights, New Jersey	Graduated - 1970
Western Maryland College Westminster, Maryland	Graduated - 1974

Bachelor of Science
in Mathematics

Lehigh University Bethlehem, Pennsylvania	1980 - 1984

Candidate for Master of
Science in Industrial
Engineering

PROFESSIONAL EXPERIENCE

U.S. Army Development and Readiness Command
Packaging, Storage, and Containerization Center
Tobyhanna, Pennsylvania

Operations Research Analyst
March 1978 - November 1981
October 1982 - present

U.S. Navy Aviation Supply Office
Philadelphia, Pennsylvania

Operations Research Analyst
November 1981 - September 1982