

15818



This is to certify that the

dissertation entitled

A Comparative Study of Manpower Scheduling Algorithm:
A Multi-Objective Heuristic Procedure, Tabu Search,
and Simulated Annealing

presented by

Aaron Paul Blossom

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Operations Management

Ram Narasimhan
Major professor

Date 7/20/93

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
 TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
APR 28 2003		
111602		

MSU Is An Affirmative Action/Equal Opportunity Institution

c:\circ\datedue.pm3-p.1

**A COMPARATIVE STUDY OF MANPOWER SCHEDULING ALGORITHMS: A
MULTI-OBJECTIVE HEURISTIC PROCEDURE, TABU SEARCH, AND SIMULATED
ANNEALING**

By

Aaron Paul Blossom

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Management

1993

ABSTRACT

A COMPARATIVE STUDY OF MANPOWER SCHEDULING ALGORITHMS: A MULTI-OBJECTIVE HEURISTIC PROCEDURE, TABU SEARCH, AND SIMULATED ANNEALING

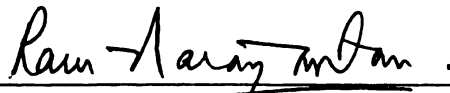
By A. P. Blossom

Achieving minimum staffing costs, meeting worker's requests for hours worked and time off while maintaining required staffing levels are all important but conflicting objectives when scheduling service employees. Current employee scheduling methods address only one or two of these objectives.

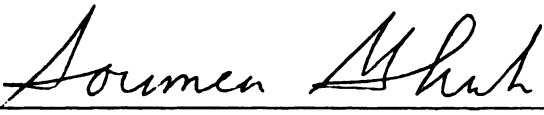
This research implements two algorithms new to the manpower scheduling field, simulated annealing and tabu search. The algorithms were then compared against each other and a multi-objective heuristic developed by Loucks on several measures of solution quality. This research was conducted in the domain of the tour scheduling problem with non-interchangeable workers and using the full set of possible shifts. All previous comparative research has used a working subset approach.

Tests were conducted with problems constructed with data from service businesses and with synthetic problems. The solution methods were tested for the effect of several independent variables measuring solution quality, cost and time-to-completion. Simulated Annealing was found to generally outperform Loucks' heuristic and tabu search on the measures of solution quality and cost, while Loucks' heuristic was clearly better in time-to-completion.

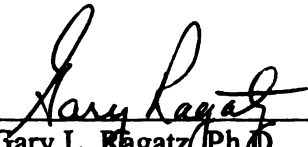
Accepted by the Graduate Faculty, Michigan State University, in partial fulfillment of the requirements of the degree of Doctor of Philosophy



Ram Narasimhan, Ph.D.



Soumen Ghosh, Ph.D.



Gary L. Kagatz, Ph.D.

July 20, 1993

**Copyright by
Aaron Paul Blossom
1993**

**Dedicated to my boys,
Aaron M. Blossom,
Andrew P. Blossom, and
Joseph C. Blossom
for their patience and help.**

ACKNOWLEDGEMENTS

I wish to thank my dissertation committee, Dr. Ram Narasimhan, Dr. Soumen Ghosh, and Dr. Gary Ragatz for their assistance, guidance and support in the preparation of this research. This dissertation could not have been completed without their contributions.

The Chairman of the Dissertation Committee, Dr. Ram Narasimhan, has my profound gratitude for the mentoring, counsel and support throughout my doctoral student career. His confidence and continuing interest in this student and the research made the dissertation possible.

Dr. Gary Ragatz and Dr. Soumen Ghosh generously took time to discuss this research with me whenever necessary. They offered important suggestions throughout the dissertation process and contributed greatly to my ability to complete the work.

I also wish to thank my boys, Aaron, Andrew and Joseph, for their sacrifices of time and attention with their father. Without their help, this dissertation would have been impossible to complete.

Table Of Contents

List of Tables	x
List of Figures	xii
Chapter 1. Introduction	1
1.1. Introduction	1
1.2. Special Characteristics Of Service Businesses	1
1.3. Objectives for Scheduling in Service Systems	3
1.4. Problem setting	3
1.5. Problem Statement	4
1.5.1. Three Step Manpower Scheduling Process	5
1.5.2. Restrictions on Work Tours For This Research Setting	7
1.6. Research Focus and Objectives	9
1.6.1. Critical aspects of research setting	10
1.6.2. New Methods Used For Solution Of This Problem	11
1.7. Algorithms Previously Applied in this Setting	11
1.8. Organization of the Dissertation	12
Chapter 2. Review of Related Literature	13
2.1. Organization of Literature Review	13
2.2. No Work-Time Scheduling Decision	16
2.2.1. No Work Time Decision - Unlimited Worker Interchangeability	16
2.2.2. No Work Time Decision - Limited Worker Interchangeability	17
2.3. Days-Off Scheduling Decision	17
2.3.1. Days-Off Scheduling Decision - Unlimited Worker Interchangeability	17
2.3.2. Days-Off Scheduling Decision - Limited Worker Interchangeability	19
2.4. Shift Scheduling Decision	19
2.4.1. Shift Scheduling Decision - Unlimited Worker Interchangeability	19
2.4.2. Shift Scheduling Decision - Limited Worker Interchangeability	21
2.5. Tour Scheduling Decision	22

2.5.1. Tour Scheduling Decision - Unlimited Worker Interchangeability	22
2.5.2. Tour Scheduling Decision - Limited Worker Interchangeability	24
2.6. Summary of the Review Of Labor Scheduling Literature.....	25
2.7. Opportunities for Research	25
2.7.1. Application of new algorithms	25
2.7.2. Comparison of Previously Applied Algorithms	26
Chapter 3, Mathematical Formulation	27
3.1. Choice of Formulation.....	27
3.2. Problem Formulation.....	28
3.3. Assumptions of Formulation.....	31
Chapter 4, Generation of Initial Feasible Solution.....	33
4.1. Introduction.....	33
4.2. Strategy of initial feasible solution generator	35
4.2.1. "Fat" solution, vs. "Lean" solution	35
4.2.2. Margin on task hours	36
4.2.3. Choice of worker for task hour	36
4.2.4. Elimination of Constraint Violations	39
4.3. Step by Step Explanation Of IFS Generator.....	43
4.3.1. Explanation of Steps in Flow Chart	45
4.3.2. Pseudo Pascal Description of IFS Generator Algorithm	49
4.3.3. Conclusion	50
Chapter 5 - Tabu Search: Theory and Implementation.....	51
5.1. Introduction.....	51
5.1.1. Overview of Tabu Search	51
5.1.2. Description of Tabu Search Algorithm In Pseudo-Pascal	56
5.1.3. Example of Tabu Search Algorithm	57
5.2. Tabu Search Implementation Issues	58
5.2.1. Tabu List Types	58
5.2.2. Aspiration Criteria	59
5.2.3. Selection of Parameter Values	60
5.2.4. Comparison of Results for Different Tabu List Lengths.	61
5.2.6. Generation of Moves or New Solutions	61
5.2.7. Move Generators	62
5.2.8. Formulation of constraints.	80
5.2.9. Objective Function Estimator	80
5.3. Conclusion.....	81
Chapter 6 - Simulated Annealing; Theory and Implementation.....	82
6.1. Introduction.....	82
6.1.1. Statistical Mechanics	82

6.1.2. The Simulation of Particles at Thermal Equilibrium	83
6.1.3. The Relationship to Combinatorial Optimization	84
6.1.4. Requirements for Simulated Annealing Optimization	85
6.1.5. Description of Simulated Annealing Algorithm in Pseudo-Pascal	86
6.1.6. Flow Chart of Simulated Annealing	86
6.1.7. Example of The Simulated Annealing Algorithm	88
6.2. Simulated Annealing Implementation issues	89
6.2.1. The Annealing Schedule	89
6.2.2. Setting the parameter values	90
6.2.3. Stopping Rules	91
6.2.4. Perturbation Using Configuration Generators	92
6.2.5. Inclusion of Both Improvement And Random Configuration Generators	116
6.2.6. Probabilities used for selection of configuration generators	119
6.2.7. Improvement Path for Simulated Annealing	120
6.3. Conclusion	120
Chapter 7 - Research Methodology	122
7.1. Introduction	123
7.2. Choice of Algorithm for Comparison	124
7.3. Hypotheses	124
7.3.1. Time - Speed of Algorithm	124
7.3.2. Cost - Total Manpower Cost of the Schedule.	125
7.3.3. Quality Of Solution	125
7.3.4. TSD - Sum of squared deviations between scheduled and targeted work hours.	126
7.3.5. OBJ - Objective Function Value	126
7.3.6. Interactions among variables	127
7.4. Selection of Experimental Design	127
7.4.1. Factor Levels	127
7.4.2. Choice of Design	128
7.4.3. Sample size of experiment	128
7.4.4. Test data	129
7.4.5. Analysis	130
7.4.6. Choice of sample size	130
7.4.7. Replications for Simulated Annealing	131
7.5. Generation of Test Problems	133
7.6. Data Gathering	135
Chapter 8 - Results of Experimentation	136
8.1. Introduction	136
8.2. Tests for Assumptions of ANOVA	136
8.2.1. Normality of error residuals	137
8.2.2. Normal Probability Plots of Residuals	138

8.2.3. Independent distribution of error residuals	140
8.2.4. Homogeneity of variance	141
8.2.5. Bartlett's Test	144
8.3.Box and Cox Analytical Method for Determining Transformation.....	144
8.3.1. Explanation of Analytical Determination of Transformation	145
8.3.2. Results of the Analytical Determination of Transformation	146
8.3.3. Normal probability plots of residuals	147
8.3.4. Independent distribution of error residuals	150
8.3.5. Homogeneity of Variance	151
8.3.6. Bartlett's test after transformations	154
8.4.Justification for use of Analysis of Variance.....	154
8.5.Results of ANOVA.....	155
8.5.1. Detailed Results	156
8.6.Differential Difficulty of Problem Types	177
Chapter 9 - Summary, Conclusions and Recommendations.....	179
9.1.Summary	179
9.2.Contributions	180
9.3.Opportunities for Further Research	182
Appendix 1.	184
Calculation of Theoretical Number of Work Tours adapted from Loucks [1987].	184
Notation	184
Assumptions.....	184
Equation.....	185
Example	185
Appendix 2	186
Transformation Of Problem Formulation Into Single Function Form.....	186
Appendix 3	188
Synthetic Problem Generator.....	188
Appendix 4	190
Raw Data.....	190
Appendix 5	217
Transformed Treatment Averages.....	217
Bibliography.....	220

LIST OF TABLES

Table 1-1, Table for Translation of Sales into Required Manpower.....	6
Table 5-1, Comparison of Tabu List Length and Objective Function Value	61
Table 6-1, ANOVA for Objective Function	117
Table 6-2, TUKEY HSD Multiple Comparison Test, Objective Function Value	118
Table 7-1, Factors and Factor Levels	128
Table 7-2 Results of the analytic procedure for determination of sample size	130
Table 8-1, Autocorrelation Values For Dependent Variables	141
Table 8-2, Bartlett's Test For Homogeneity of Variance.....	144
Table 8-3, Results of Procedure for Analytic Determination of Transformation	147
Table 8-4, Autocorrelation of transformed residuals	150
Table 8-5, Bartlett's Test For Homogeneity of Variance	154
Table 8-6 Analysis Of Variance Results	156
Table 8-7. ANOVA on TIME, Transformed	159
Table 8-8, Table of the best TIME performance on problem type by algorithm	161
Table 8-9, Table of the worst TIME performance on problem type by algorithm	161
Table 8-10. ANOVA on COST, Transformed	163
Table 8-11, Correlation coefficients on COST performance between Algorithms	164
Table 8-12, Best COST performance across problem types by algorithm	165
Table 8-13, Worst COST performance across problem type by algorithm	166

Table 8-14. ANOVA on TOS, Transformed	168
Table 8-15, Best TOS performance across problem type by algorithm	170
Table 8-16, Worst TOS performance across problem type by algorithm	170
Table 8-17. ANOVA on TSD, Transformed	172
Table 8-18 Best TSD performance across problem type by algorithm	174
Table 8-19, Worst TSD performance across problem type by algorithm	175
Table 8-20, Best OBJ performance across problem type by algorithm	177
Table 8-21, Worst OBJ performance across problem type by algorithm	177
Table A4-1. Raw Data	189
Table A5-1, Treatment Averages Of Transformed Data	215

LIST OF FIGURES

Figure 1-1, Typical Demand Pattern For a Fast Food Restaurant	4
Figure 2-1, Taxonomy of Literature Review	14
Figure 2-2, A Classified Sample of Labor Scheduling Research	15
Figure 4-1, Selecting a Worker for Assignment	37
Figure 4-2, Eliminate Constraint Violations	40
Figure 4-3, Procedure to Generate Initial Feasible Solution	44
Figure 4-4, Create or Add to Shift	48
Figure 5-1, Tabu Search Algorithm	53
Figure 5-2, Tabu Search Improvement Path for Different Tabu List Lengths	61
Figure 5-3, Improvement Path for Tabu List Length = 7	62
Figure 5-4, Reduce Over Staffing	65
Figure 5-5, Increase Over Staffing	67
Figure 5-6, Give An Hour to Another Worker.....	69
Figure 5-7, Give A Day to Another Worker	71
Figure 5-8, Trade Shifts With Another Worker	73
Figure 5-9, Cross Trade Shifts With Another Worker	75
Figure 5-10, Give Away of a Worker's Shift	77
Figure 5-11, Give Away A Short Shift, Hour By Hour	79
Figure 6-1, Simulated Annealing Flowchart	87

Figure 6-2, Reduce Over Staffing	93
Figure 6-3, Increase Over Staffing	95
Figure 6-4, Give An Hour to Another Randomly	97
Figure 6-5, Give A Day To Another Randomly	99
Figure 6-6, Trade Tours Randomly	101
Figure 6-7, Cross Trade Tours Randomly	103
Figure 6-8, Give An Hour to Another	105
Figure 6-9, Give A Day to Another	107
Figure 6-10, Trade Tours Between Workers.....	109
Figure 6-11, Cross Trade Tours Between Workers	111
Figure 6-12, Give Away Part Of Shift To Other Workers.....	113
Figure 6-13, Give Away a Short Shift	115
Figure 6-14, Simulated Annealing Improvement Path	120
Figure 8-1, Normal probability plot of TOS residuals	138
Figure 8-2, Normal probability plot of TSD residuals	138
Figure 8-3, Normal probability plot of OBJ residuals	139
Figure 8-4, Normal probability plot of COST residuals.....	139
Figure 8-5, Normal probability plot of TIME residuals	140
Figure 8-6, Plot of Residual vs..Estimate - TOS	141
Figure 8-7, Plot of Residual vs..Estimate - TSD.....	142
Figure 8-8 Plot of Residual vs..Estimate - OBJ	142
Figure 8-9, Plot of Residual vs..Estimate - COST	143
Figure 8-10, Plot of Residual vs..Estimate - TIME	143
Figure 8-11, Normal probability plot of TOS residuals After Transformation	148
Figure 8-12, Normal probability plot of TSD residuals After Transformation	148

Figure 8-13, Normal probability plot of OBJ residuals After Transformation	149
Figure 8-14, Normal probability plot of COST residuals After Transformation	149
Figure 8-15, Normal probability plot of TIME residuals After Transformation	150
Figure 8-16, Plot of Residual vs..Estimate - TOS After Transformation	151
Figure 8-17, Plot of Residual vs..Estimate - TSD After Transformation	152
Figure 8-18, Plot of Residual vs..Estimate - OBJ After Transformation	152
Figure 8-19, Plot of Residual vs..Estimate - COST After Transformation	153
Figure 8-20, Plot of Residual vs..Estimate - TIME After Transformation	153
Figure 8-21, Plot of TIME by Algorithm vs..Problem Type	160
Figure 8-22, Plot of COST by Algorithm vs..Problem Type.....	164
Figure 8-22, Plot of TOS by Algorithm vs..Problem Type	169
Figure 8-22, Plot of TSD by Algorithm vs..Problem Type	173
Figure 8-23, Plot of OBJ by Algorithm vs..Problem Type.....	176

CHAPTER 1. INTRODUCTION

1.1. INTRODUCTION

Labor scheduling is a required managerial task in any firm which employs workers in the production of goods and services. Labor is typically a limited and indispensable resource which must be managed effectively in order for the firm to achieve its objectives. The effective scheduling of manpower is a major component of the management of labor. As such, manpower scheduling has a great deal of impact on profitability, customer service and employee morale. The specific manpower scheduling decisions that a manager must consider depend upon the environment, business and labor, that the scheduling decisions are made in. This research focuses on manpower scheduling in the service sector, which has characteristics distinct from other sectors of the economy.

1.2. SPECIAL CHARACTERISTICS OF SERVICE BUSINESSES

The special characteristics of the service environment relevant to manpower scheduling are [Aggarwal, 1982]:

- The output of service systems (a service system is a "factory" that provides services), generally cannot be placed into inventory. The exceptions are products, such as hamburgers, that have a very short shelf life (10 minutes). Consequently in service systems it is not possible to level output to meet demand.
- Service systems require equipment, labor and space capacity to handle peak or near peak demand conditions. In other words, the capacity decision in service systems is to buy capacity for peak, rather than average, demand levels.
- The demand for output of service systems varies greatly from month-to-month, week-to-week and day-to-day. Extreme variations from hour-to-hour exist in some service systems. The demand may also have seasonal variations.
- The demand for output of service systems cannot be easily backlogged. Usually unsatisfied demand is lost business. In cases where the demand can be backlogged, the indirect costs may be significant.
- Since in most service systems, the customer receives the service directly from the server, they are labor intensive.

Since demand must be satisfied or lost and material costs are fixed for a particular product, the most significant opportunity for reducing costs is through better scheduling of manpower. In addition, the opportunities for improving customer service and employee morale lie primarily in improved manpower scheduling.

1.3. OBJECTIVES FOR SCHEDULING IN SERVICE SYSTEMS

Service systems also differ from manufacturing systems in that they have different objectives. Indisputably, cost minimization is an objective of service systems, but their primary objectives are often different because of the direct interaction with the customer. The objectives of a service firm might include the following:

- Minimization of average response time to start of service delivery.
- Minimization of the average customer waiting time.
- Minimization of average customer service time.
- Minimization of the average number of facilities or work crews.
- Improvement and maintenance of employee morale through scheduling employees at preferred times (of the employee).

Additionally, the manager of the service system must operate within a large number of legal, regulatory, union, policy, and budgetary constraints.

In summary, labor scheduling is a major concern in service businesses, with the conflicting objectives of maintaining desired levels of customer service and minimizing costs.

1.4. PROBLEMSETTING

The research is concerned with scheduling problems faced by four environments, food service, retail, banking and a series of synthetic problems. The research will consider scheduling of work tours with limited worker interchangeability in the four service environments mentioned above. Limited worker interchangeability refers to

restrictions placed on the assignments of workers. Not all workers can perform all tasks, and must not be assigned to tasks for which they are not qualified.

A typical demand pattern from a fast food restaurant is shown in figure 1-1.

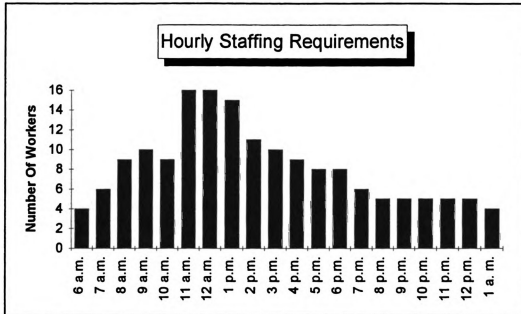


Figure 1-1, Typical Demand Pattern For A Fast Food Restaurant

Note that the demand varies significantly over short time periods. In order to minimize costs, scheduling is typically done in relatively small time increments, one hour or less, creating a large set of possible shifts. A review and taxonomy of labor scheduling problems is presented in chapter 2. The particular problem studied here of work tour scheduling with limited worker interchangeability is detailed in section 2.5.2.

1.5. PROBLEM STATEMENT

The manpower scheduling problem presented here is one of deciding who works when and at what task. Stated more precisely, the manager must decide an employees

work shifts and task assignment over the time span chosen. In this research a one week time horizon is chosen for the scheduling activity. Further, each day is broken into 1 hour increments. The manager, in order to provide adequate customer service, will desire to have on duty, during every hour of each day scheduled, enough employees to satisfy the demand for services forecasted during that hour. Because of restrictions in the problem setting, such as minimum shift length, a manager may not be able to schedule the optimum number of workers during a particular hour. The manager may then decide to over staff a particular hour.

1.5.1. Three Step Manpower Scheduling Process

The manpower scheduling process consists of three steps, Swart and Donno [1981]. First, a weekly sales/demand forecast is developed. Second, the weekly sales/demand forecast is converted into hourly staffing requirements. And third, the manpower schedule is developed, which specifies for each worker, the hours on duty, i.e. the "tour", and the task to be performed during each on-duty hour.

Sales / Volume Forecasting. Short-term, i.e. one week, sales/demand forecasting is the first step in the manpower scheduling process. This forecast is typically made using three sources of data, historical sales/demand data, current trends of sales/demand and the schedule of special events. The manager will gather past sales/demand data (for example, last week's sales/demand data, or same week last year sales/demand data). This data may be used as is or the manager may average several past periods to provide an initial sales/demand forecast.

Second, the manager will adjust the forecast using the two remaining sources of information, current sales/demand trends and the schedule of special events. Trends in

the sales/demand patterns of the firm may cause the manager to adjust the forecast of individual hours or the entire week up or down. Special events, such as a sale, athletic event, or parties may cause increased demand for the services of the organization. In this case, the manager would estimate the effect of the special event and adjust the forecast accordingly.

Determination of Staffing Requirements. Once the sales/demand forecast is complete, the forecast is converted into staffing requirements for each hour. These requirements would be the number of workers required to staff each task for each hour forecasted.

Often, a table is used to convert hourly sales forecasts into staffing requirements. Such a table is shown in table 1.1. This particular table divides the work load into eight tasks, each of which may or may not be staffed at a particular demand level. Note that this table specifies tasks and staffing levels associated with meeting customer demand and does not include activities where the timing is not critical, such as planned maintenance. Tables like these may be developed in a variety of ways, such as expert opinion and simulation, Swart and Donno [1981].

Table 1-1, Table for Translation of Sales into Required Manpower.

Sales	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Total
\$100	1		1				2
\$200	1	1	1				3
\$300	1	1	1	1			4
\$400	2	1	1	1	1		6
\$500	2	1	2	1	1		7
\$600	2	1	2	1	1	1	8
\$700	3	1	2	1	1	1	9
\$800	3	2	2	1	1	1	10
\$900	3	2	3	1	1	1	11
\$1,000	3	2	3	2	1	1	12

Two additional points about staffing requirements may be made. First, the staffing requirements shown on the table are minimum staffing requirements. And second, typically in fast food and retail establishments, some amount of staffing is required before opening and after closing. The staffing requirements for these hours cannot be determined by using a table based on hourly sales/demand volumes.

Task and Time Scheduling of Workers. The final stage of the manpower scheduling process is the development of a manpower schedule that specifies the workers on duty and the task assignments. A schedule usually consists of at least two reports, one for the manager that lists the workers for each hour and task, and the second for each worker that lists the days scheduled, the hours scheduled during those days, and the tasks assigned during those hours.

The objectives for the manpower schedule development include minimizing overstaffing of task hours, minimize under staffing of task hours, and meeting the worker's preferences for the number of hours scheduled, while not violating work rules such as minimum shift length, maximum shift length, task qualifications, etc.

It might seem that once a schedule for a week is developed, it would only need minor modification for use in subsequent weeks. There are several factors that cause that supposition to be true only occasionally. These factors include, high worker turnover (at least in some industries/firms), variations in sales/demand patterns, special events, changes in worker availability, and etc.

This final stage of the manpower scheduling process is the focus of this research.

1.5.2. Restrictions on Work Tours For This Research Setting

To facilitate discussion of the problem, related literature, mathematical formulation and results, a common set of terms will be defined. After these definitions common to the scheduling literature are established, three aspects of the problem, limited worker interchangeability, limited worker availability, and worker task qualifications will be discussed.

Definitions/Terminology. The first terms to be defined are "shift" and "work shift". These two terms may be used interchangeably. A shift is a period of work during a day. A work shift has a start and a finish time. The period of time between the start time and finish time is sometimes called a "work stretch". A work stretch may contain rest breaks or lunch breaks. In some scheduling environments "split" shifts exist. A split shift is a daily work shift that has a break of more than one half hour. A split shift would be defined by two sets of start and finish times for a single day. These two sets would not, of course, overlap.

The term "work tour" is used to describe the set of shifts scheduled for a worker over the scheduling horizon. A work tour would consist of the shifts scheduled as well as the days off, or "relief days". The "scheduling horizon" is the length of time for which a schedule is produced. For our research, the scheduling horizon is one week. In this research the scheduling process produces schedules where the shifts are generally not the same from workday to workday. Such a tour is said to contain "mixed shifts."

Limited Worker Interchangeability. The term limited worker interchangeability refers to existence of characteristic differences between workers that must be considered during the third step of the scheduling process. If there are no such differences, the workforce is called homogenous. A homogenous workforce has workers that are completely interchangeable. On the other hand, a heterogeneous workforce has workers

whose differences must be taken into account when the schedule is constructed. These differences might be due to differing times available during the day and/or week, and different tasks the worker is qualified to perform.

Limited Worker Availability. A worker's availability is defined as the specific hours during the week that the worker has agreed to be available for work. Because the variety of workers in service firms ranges from full-time professionals to part-time high school students, it can be reasonably inferred that the availability of workers would vary from one to another, being limited for some or all employees. School, family, transportation, and curfews are examples of conditions that can restrict a worker's availability.

A part-time employee will accept a schedule with mixed shifts and non-consecutive workdays in order to restrict his availability and to have the option of changing that availability to accommodate needed (or desired) changes in schedule. The manager, however, would like to have unrestricted availability to ease the scheduling process and to maintain the greatest degree of flexibility for use in the scheduling process. Some companies require unrestricted availability of their full time employees, in exchange for the benefits of full time employment, such as health insurance, unemployment insurance, guaranteed number of hours, etc.

Worker Task Qualifications. The particular tasks (or functions) that a worker is qualified to perform are referred to as the worker's task qualifications. Because of the cost of training, and the rate of employee turnover (sometimes high), the manager may not train every new employee on every task as quickly as possible. The manager may train the employee on those tasks which the manager deems most necessary, probably based on an assessment of task coverage needs. Task coverage is the degree to which the task required to be performed by the forecast can be covered by the employees available to be scheduled.

1.6. RESEARCH FOCUS AND OBJECTIVES

1.6.1. Critical aspects of research setting

The characteristic of service systems that output cannot be inventoried requires scheduling enough workers for adequate customer service or, alternatively, minimization of staffing shortage hours (too few workers scheduled can cause unacceptable service delivery times). When delivery of service is critical, such as in an health care environment, staffing shortage hours cannot be tolerated. On the other hand, high volume and low margins in food service require minimization of staffing surplus hours (scheduling too many workers does not improve service delivery times sufficiently to offset increases in costs). Changes in the relative importance of surplus and shortage hours can be accommodated in the problem formulation to be used in this research (chapter 3) by adjusting parameters associated with the decision variables.

Task specific, non-interchangeable workers exist in the problem settings considered in this research. Workers can be qualified in more than one task, however. This means that workers must be scheduled for specific tasks at specific times.

As noted before, in the food service environment demand patterns vary from hour-to-hour, day-to-day, week-to-week, and etc. This requires the scheduling to be performed with short time intervals in order to minimize surplus hours, and hence cost.

In order to accommodate part-time workers, to increase worker satisfaction levels and to reduce worker turnover, worker preferences must be taken into account. The worker preferences would include: minimum and maximum work hours per day and per week, preferable work times during the day (mornings, afternoons or evenings) as well as days off.

These characteristics suggest that the number of potential solutions to be considered to identify the "best" solution is large. As an example, consider a fast food restaurant with these characteristics: it is open 18 hours per day, 7 days per week, the worker shifts are from 3 to 8 hours in length and the number of days worked per week can vary from 1 to 5. In this example there are 74,747,846,439 possible combinations of shifts or work tours. For a calculation method for the theoretical number of work tours, please see appendix 1.

In addition, a mathematical characterization of the work tour scheduling problem leads to a combinatorial optimization problem, since many of the variables are integer valued. Because of the combinatorial nature of the problem and its large size, integer programming methods are currently not useful. There are two primary reasons integer programming solution methods are not useful [Glover, 1990]: First, solution times are unacceptably long, given the size of the problem for even small businesses, and second, most computer codes available for small computers will not accommodate the large number of variables necessary to model the problem.

1.6.2. New Methods Used For Solution Of This Problem

This research applied two new methods to this research problem, simulated annealing and tabu search. Their implementation in this research demonstrates that more general procedures can be applied to this labor scheduling problem and provides additional information about the relative merits of tabu search and simulated annealing.

1.7. ALGORITHMS PREVIOUSLY APPLIED IN THIS SETTING

There have been three algorithms discussed in the literature to solve the problem of work tour scheduling with limited worker interchangeability; Heuristic Programming [Glover, 1988], Multi-Objective Heuristic [Loucks, 1987], and Network Solution Methods [Love and Hoey, 1990]. A distinction might be made between the research problems used by Glover and Love and Hoey, and Loucks. In the Glover and Love and Hoey research the length and position of the shifts during the working day were fixed. In Loucks, neither the shift length nor the position of a shift is fixed within the day. The research describing the development and application of these three techniques are detailed in the literature review section 2.5.2.

1.8. ORGANIZATION OF THE DISSERTATION

The remainder of the dissertation consists of: a literature review comprises chapter 2, and the mathematical formulation of the research problem is given in chapter 3. The initial feasible solution generator is presented in chapter 4, and the implementation of tabu search is discussed in chapter 5. The subject of chapter 6 is the theory and implementation of simulated annealing. Chapter 7 is devoted to the experimental design. Results of the experiment are discussed in chapter 8, with the summary and conclusions presented in chapter 9.

CHAPTER 2. REVIEW OF RELATED LITERATURE

2.1. ORGANIZATION OF LITERATURE REVIEW

The organization of this review is an adaptation of Loucks' (1987) scheme for the categorization of the labor scheduling problem. He proposed two criteria for classification of labor scheduling problems, work tour set composition and worker interchangeability. A work tour set is the set of different work shifts (tours) from which a worker's schedule is selected. A work tour set can be multiple shifts or a single shift per day. The work tour set can be further characterized by its "days-off" patterns, whether there is a single days-off pattern or multiple days-off patterns.

Worker interchangeability is defined as the degree to which the workers being scheduled differ in their qualifications to perform the various tasks in the work environment, times available to work, or other relevant properties. Worker interchangeability, for the purposes of this literature review, can be split into two categories, limited and unlimited. Figure 2-1 shows a taxonomy which can be used to classify the existing body of literature. Figure 2-2 shows a representative sample of the literature classified into the taxonomy presented in figure 2-1.

2.1.1. Taxonomy of Literature Review

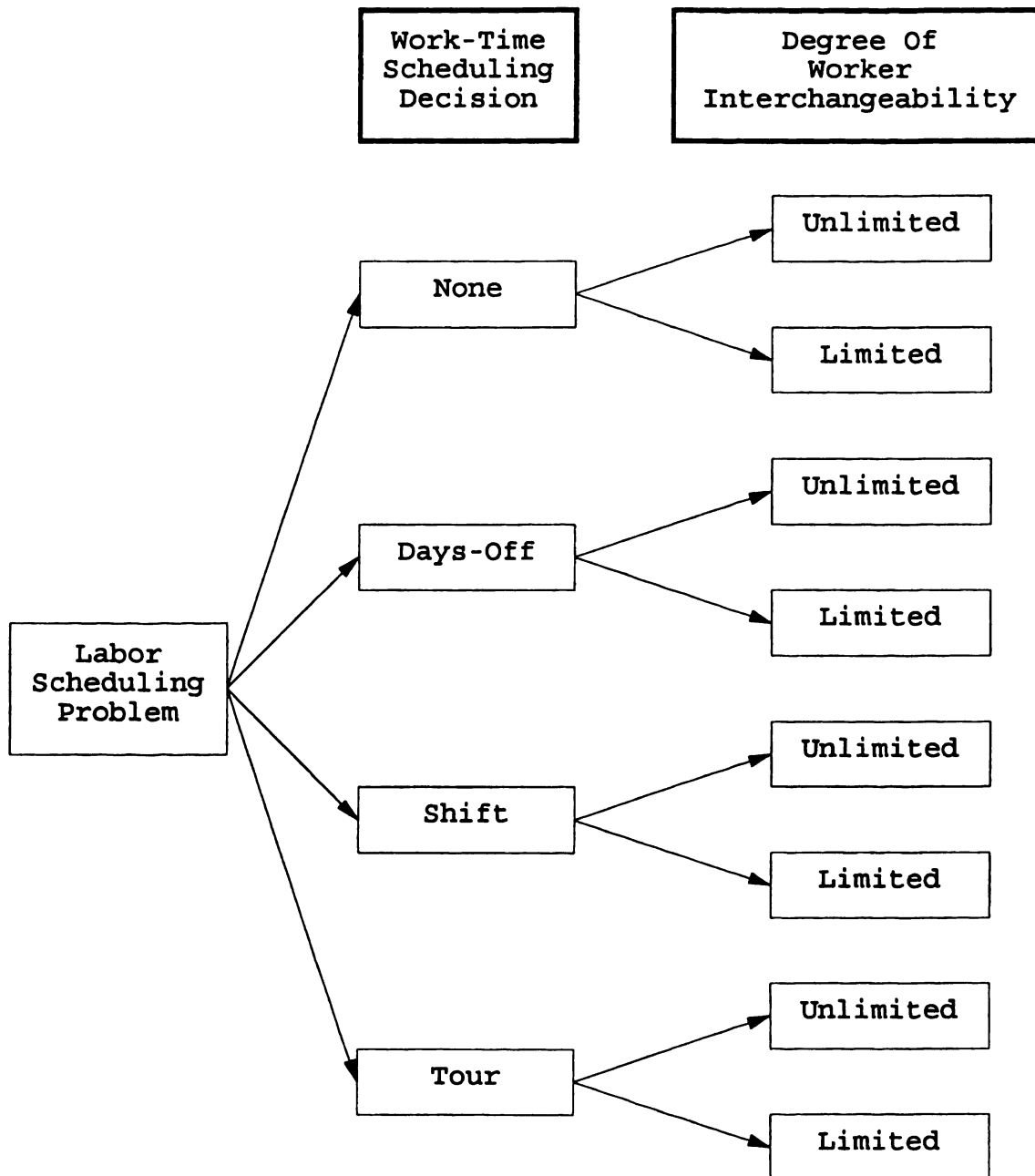


Figure 2-1, Taxonomy of Literature Review

2.1.2. A Classification of Labor Scheduling Research

Labor Scheduling Problem	Degree of Worker Interchangeability	
	Unlimited	Limited
None*	Holstein/Berry [1972] Job Shop Workers	Charnes/Cooper [1961] Job Shop Workers
	Church [1973] Telephone Service Reps.	
	Miller/Berry [1974] Job Shop Workers	
Days-Off	Baker [1974a] Generalized	
	Brownell/Lowerre [1976] Generalized	
	Monroe [1970] Baggage Handlers	
Shift	Baker/et al. [1973] Baggage Handlers	Pappas [1967] Railroad Personnel
	Henderson/Berry [1977] Telephone Operators	
	Keith [1979] Telephone Operators	
Tour	Showalter/et al. [1977] Mail Sorters	Warner/Prawda [1972] Hospital Nurses
	Morris/Showalter [1983] Generalized	Ritzman/et al. [1976] Mail Sorters
	Mabert/Watts [1982] Check Encoders	Loucks [1987] Fast Food

* This category is characterized by an absence of any work time decision, as described in the text. The decision is primarily one of job assignment.

Figure 2-2. A Classified Sample of Labor Scheduling Research

2.2. NO WORK-TIME SCHEDULING DECISION

This category represents the single shift - single days-off pattern. In the absence of any work-time scheduling decision, the decision is primarily job assignment. This scheduling decision is primarily used in an operation that operates one shift per day, five days per week.

2.2.1. No Work Time Decision - Unlimited Worker Interchangeability

Given a single work tour and manageable work loads, the scheduling problem becomes one of job assignment. Miller and Berry [1974] examine the problem of assigning workers to several semi-automatic machines. Two heuristics were developed for determining labor assignments: the "man-loading" and "labor-saved" heuristics. Their heuristics minimize the combined costs of idle labor and machine time. They compare their heuristics against the optimal solutions found by a branch and bound algorithm.

The scheduling of staff to answer phones and process paperwork was studied by Church [1973]. The problem he studied was to assign workers to one of two tasks over half hour time intervals. His heuristic assigned workers to tasks with the objective of equalizing work load among workers and keeping a worker's task the same through several consecutive periods and then switching tasks.

2.2.2. No Work Time Decision - Limited Worker Interchangeability

The scheduling problem with a single work tour and limited worker interchangeability is a restricted assignment problem. This type of problem has been dealt with in the linear programming literature and is efficiently solved by such methods [Charnes and Cooper, 1961].

2.3. DAYS-OFF SCHEDULING DECISION

This category represents the single shift - multiple days-off pattern. This scheduling decision is typically found in a business that functions 8 hours per day and more than 5 days per week. Thus the decision is categorized as a days-off problem or a days-off and task assignment problem, for unlimited and limited worker interchangeability, respectively [Loucks, 1987].

2.3.1. Days-Off Scheduling Decision - Unlimited Worker Interchangeability

The most common form of this problem is scheduling five day work tours with a seven day operation. The objective is to find the minimum labor force size while meeting the demand for the worker's services, and permitting two consecutive days off. If excess staffing is necessary to assure feasibility, the algorithms by Baker [1974a] and Tibrewala [1972] are optimal and simple to use.

Several variations of this problem can be found in the literature. Bartholdi and Ratliff [1978] examine the problem with several different days-off policies. Monroe

maximized consecutive days-off without employing part-time workers. Baker [1974b] relaxes the five day work tour constraint to minimize the number of part-time man-days to exactly meet the staffing requirements.

Brownell and Lowerre [1976] compare days off policies, with regard to the manpower requirements for each different policy.

Miller, Pierskalla and Rath [1976] use a cyclic coordinate descent algorithm to find near optimal solutions. They compared their algorithm to an optimal branch and bound algorithm, and found that their solutions were near-optimal and that the algorithm was much faster than the branch and bound. Their algorithm was also found to be better than the manual methods in use at the time.

Khan [1979] uses a capacitated network flow method for producing manpower schedules. Khan found that this method was substantially more efficient than the Simplex method in this application.

Baker and Magazine [1977] present a lower bound for workforce size and feasible schedule construction algorithms for 4 different days-off policies.

Baker, Burns and Carter [1979] Develop lower a bound on workforce size and manual methods for constructing schedules to meet these lower bounds.

Shepardson and Marston [1980] reformulate the two duty period scheduling problem as a one duty period problem with side constraints. Since the one duty period problem can be solved using a minimal cost network flow model, dualization is used with respect to the side constraints, forming a Lagrangean relaxation that is easily solved.

Bartholdi [1981] solves the cyclic staff scheduling problem using linear programming with a round-off procedure. He shows that his round off algorithm is better than those presented earlier.

Burns and Carter [1985] developed a simple, one pass algorithm to generate schedules and to compute lower bounds on the workforce size.

2.3.2. Days-Off Scheduling Decision - Limited Worker Interchangeability

There are undoubtedly scheduling environments with single work tours and limited worker interchangeability. There seems to be no published work in this area.

2.4. SHIFT SCHEDULING DECISION

This category represents the multiple shifts - single days-off pattern. The shifts assigned to different workers might overlap or be contiguous. Overlapping shifts are typical of a service environment. Labor costs can be reduced while maintaining customer service levels with overlapping shifts.

2.4.1. Shift Scheduling Decision - Unlimited Worker Interchangeability

Baker [1973] formulates the problem of scheduling 8 hour shifts with 4 hour overlaps as an integer program. He develops results that show that a linear program solution to his formulation will always produce an integer solution provided that the period requirements are integer valued.

As requirement periods are shortened and the number of shifts are increased, problem size grows. Problem size in business environments easily exceeds the limits of existing, commonly available integer programming codes. As an additional complication to the solution of the labor scheduling problem, LP solutions do not typically provide integer solutions. In that sense, Baker's [1973] LP solution is uncommon in that it provides integer solutions.

As a result of the limitations of integer programming codes, researchers have developed rounding heuristics for LP solutions. Keith [1979] developed a heuristic to find an integer solution close to the optimal solution. First he rounds the solution to the nearest integer, then he attempts to add or remove workers from the rounded solution.

For some large problems, such as those encountered in labor scheduling, even linear programming can prove to be inefficient. Segal [1974] developed a network-flow formulation which guarantees integer solutions for very large problems. The limitation of Segal's formulation is that lunch and breaks cannot be explicitly modeled, or scheduled.

Henderson and Berry [1976] develop heuristics to select "working subsets" of shift schedules from the "master set" of all possible shifts, without appreciably reducing solution quality.

Buffa [1976] outlines an integrated work shift scheduling system used to schedule telephone operators. A heuristic is used to schedule the operators. An outline of requirements for heuristics to solve manpower scheduling systems is given.

Mabert and Raedels [1977] compares two heuristic methods with an integer programming formulation for scheduling part-time workers to meet varying daily work loads. Computation time was the criterion for ranking methods.

Henderson and Berry [1977] have developed a branch and bound algorithm to efficiently solve problems having as many as 100 shifts per day. The large number of shifts allows breaks and lunch to be explicitly scheduled, assigning workers to both work and break shifts.

Mabert [1979] develops a non-deterministic model for scheduling workers in a bank. He also develops the idea of safety capacity, which is analogous to safety stock in inventory theory, to meet varying volume demands when forecast errors are present.

Wilson and Willis [1983] developed a network flow and a LP model for this problem, constrained by forecasted demand, shift sizes, number of supervisors and space limitations for workers. Although the network flow model produced optimal solutions,

the LP model was implemented because of the excessive input requirements for the network flow model.

Sinuany-Stern and Teomi [1986] develop both an optimal and heuristic algorithm to schedule security guards. The optimal algorithm was found to be inefficient for the problem, which was rather large. The heuristic produced nearly optimal solutions, as well as almost 50% cost savings over the manual system.

Burns and Koop [1987] present a way to develop optimal cyclic schedules (what they call master schedules). Their approach allows differing (deterministic) demand patterns for each shift.

2.4.2. Shift Scheduling Decision - Limited Worker Interchangeability

Pappas [1967] is one of few authors who have looked at limited worker interchangeability. In his study several scheduling restrictions exist: no employee should be scheduled to work on his preset rest days, and no morning shifts should be assigned to a worker that worked an evening shift the day before.

Warner [1976] poses the scheduling decision as a large multiple choice programming problem whose objective function quantifies preferences of individual nurses. The problem is solved using a modified version of Balintfy and Blackburn's algorithm for multiple choice programming problems.

Franz, et al [1989] develop a multi-objective integer linear program for scheduling an staffing multiple clinics. As we might expect optimal procedures for solving this problem were not found to be computationally efficient. The problem was

reformulated to resemble a generalized network flow model with substantially reduced computation requirements.

2.5. TOUR SCHEDULING DECISION

This is the labor scheduling problem that the proposed research focuses on. This category represents the multiple shifts - multiple days-off pattern. A business that would have this type of labor scheduling decision to make would typically be functioning more than 8 hours per day and more than 5 days per week. Shifts may or may not overlap, depending on demand patterns for workers and may change from day-to-day.

2.5.1. Tour Scheduling Decision - Unlimited Worker Interchangeability

The research referenced above has studied either shift scheduling or days-off scheduling, not both simultaneously. There are many problem environments where both decisions must be made such as in restaurants, banking, and retail operations .

Morris and Showalter [1983] treat this issue of integration, using two approaches, bottom-up and top-down. The bottom-up approach solves the two problems sequentially, while the top-down approach solves the two problems simultaneously. Their solution method for the top-down approach used a LP model with a heuristic to round the solution down. They found that the top-down approach was superior.

Mabert and Watts [1982] follow on the Henderson and Berry [1976] study of tour scheduling problem. They develop and compare six techniques for selecting a working subset of tours from the master set. They found that the techniques based on historical

demand patterns produced the lowest cost schedules. The other techniques tried were based on random selection and employee convenience.

Bechtold [1988] develops implicit optimal and heuristic methods for solving the problem in a multi-objective, multi-location environment. He explicitly models the tradeoffs between idle time, the number of employees required to work at multiple locations and the size of the total labor pool.

Li, Robinson and Mabert [1991] compare 3 different heuristics for the tour scheduling problem with unlimited worker interchangeability. They minimized total labor costs, rather than minimization of the number of hours scheduled or the minimization of the number of employees.

Bechtold, Brusco and Showalter [1991] compare 8 different heuristics with optimal solutions generated by solution of an integer linear program. The primary criterion for comparison was labor hours scheduled above optimum. Secondary criteria evaluated for each of the 8 heuristics were percentage of employees with two consecutive days off, number of active tours, mean number of daily shifts, and computational times.

Easton and Rossin [1991] develop a method for efficiently generating equivalent alternative solutions for the tour scheduling problem. They show that their method works best when it has a limited number of tours with which to work.

2.5.2. Tour Scheduling Decision - Limited Worker Interchangeability

Ritzman, Krajewski and Showalter [1976] developed a combined heuristic and simulation algorithm to construct tour assignments in a post office where workers differ by the work centers in which they are qualified to work. Their algorithm has several objectives; maximize service, minimize administrative costs and minimize night and overtime wage premiums.

A nurse scheduling problem with different skill levels was investigated by Warner and Prawda [1972]. The scheduling horizon used in the research was 3 or 4 days. The problem is formulated as a mixed integer quadratic programming problem and is decomposed into a linear 0,1 programming master problem with small quadratic programming subproblems.

Loucks [1987] and Loucks and Jacobs [1991] developed a heuristic to solve a goal programming formulation of this problem. His contention was that the problem was so large that it precluded solution by optimal methods. His research indicated that the heuristic produced high quality schedules but no comparison was made with other solution methods.

A general heuristic for solution of this problem was outlined by Glover et. al. [1984, 1988]. This procedure was implemented and tested in a fast food restaurant, and is unavailable for testing. The tests showed that the solutions were of high quality but the only comparison (informal) was against manual methods.

Love and Hoey [1990] presented an integer formulation for this scheduling problem, developed from a fast food setting. The integer problem formulation is amenable to a minimum cost network flow solution, after decomposition. The tests showed that the solutions were of high quality but the only comparison (informal) was against manual methods.

2.6. SUMMARY OF THE REVIEW OF LABOR SCHEDULING LITERATURE

As can be seen from reviewing the literature, real world labor scheduling can be a complex problem. A researcher or practitioner has to accommodate multiple objectives, multiple constraints, and many variables. Prior investigations have shown that the solution difficulty is dramatically affected by limited worker interchangeability. Reliance on specially developed heuristics is common because of the solution difficulty and problem size, causing a large number of heuristics to be developed, but no general methods. In addition, precedent exists for construction/improvement approaches, as will be used in this research.

2.7. OPPORTUNITIES FOR RESEARCH

2.7.1. Application of new algorithms

Two recently developed optimization techniques, tabu search [Glover, 1989] and simulated annealing [Kirkpatrick, 1983, van Laarhoven and Aarts, 1987], show promise in the area of combinatorial optimization. These algorithms have not been applied in any of the labor scheduling problems discussed above. Previous research [Knox, 1989 and Skorin-Kapov, 1990] has shown tabu search and simulated annealing to be superior to existing techniques in the problem areas tested (the traveling salesman problem and the quadratic assignment problem, respectively). Adaptation of these algorithms to the research problem considered in this proposal may yield a superior technique in terms of solution quality and solution speed. In addition, incorporation of heuristic information into these optimization techniques may provide some extensions to the theory of tabu

search and simulated annealing. These algorithms are explained in further detail in chapters 5 and 6.

2.7.2. Comparison of Previously Applied Algorithms

As noted in the review of the solution techniques above, no comparative study of different solution techniques for the work tour scheduling problem with limited worker interchangeability has been performed. A comparison of available techniques will be useful to guide future research in several ways.

Since the research problem is large and combinatorial, any research that compares algorithms may have general applicability in guiding further research into areas that appear to be more fruitful in terms of solution quality and solution speed.

Several algorithms have been suggested, apparently without knowledge of the others, for the solution of the work tour scheduling problem with limited worker interchangeability. A comparative study of the algorithms can provide guidance for improvements in solution quality and speed. In addition to the computation aspects, the new algorithms might enable the researcher to address the need for a different set of objectives for each firm and provide a tool that can be used for more than one problem type.

CHAPTER 3, MATHEMATICAL FORMULATION

3.1. CHOICE OF FORMULATION

There are several ways to formulate the problem of scheduling work tours with limited worker interchangeability [Love and Hoey, 1990], [Loucks, 1987], [Glover and McMillan, 1986], [Warner, 1976], [Warner and Prawda, 1972]. The formulation selected and described below is adapted from one presented by Loucks [1987].

The primary differentiator between formulations is that of fixed shifts or variable shifts. In the fixed shift formulations [Love and Hoey, 1990], [Glover and McMillan, 1986], [Warner, 1976], [Warner and Prawda, 1972] the manager defines, beforehand, the shifts to be used in scheduling the workforce, so the problem becomes assignment of workers to the predefined shifts. In the variable shift models [Loucks, 1986], the authors define their models in such a way that the shifts are generated simultaneously with the worker assignments to those shifts. The fixed shift model, because of the fixed shifts, is more constrained than the variable shift model. The more highly constrained fixed shift model has a smaller solution space and takes less time to solve, (compare, for example, the results of [Glover and McMillan, 1986] to the results of the tabu search implementation here).

The advantage of using a variable shift model is that the labor costs can be lower than those of a fixed shift model because that algorithm can more closely match the demand for workers with the assignment of workers. The primary disadvantage of these formulation is that they are more complex and time consuming to solve.

Loucks presents his formulation as a 0-1/integer goal programming problem. His formulation consists of ranked goals and constraints, and for solution, uses a two phase procedure; Phase 1) the construction of an initial feasible solution and Phase 2) improvement of the solution. Without loss of generality, and with the goal of reducing implementation complexity, Loucks' formulation will be modified to be one phase.

There are several reasons for using Loucks' formulation. First, his formulation has a large number of 0,1 variables, easing application of simulated annealing.

Second, his formulation uses variable rather than fixed shifts. This has the advantage of potentially lower costs than a formulation using fixed shifts, but may require more computation time.

Third, Loucks' formulation explicitly considers worker preferences, an advantage when workers are scarce.

3.2. *PROBLEM FORMULATION*

This particular formulation of the work tour scheduling problem with limited worker interchangeability was adapted from one presented by Loucks [1987]. His applications were in a fast food restaurant, a banking environment and synthetic problems. The mathematical formulation uses the following notation:

Definition of Notation:

A capital letter with one or more subscripts represents a finite set.

The number of elements of a finite set, say A , is denoted by $|A|$.

I = number of workers being scheduled/assigned in labor pool,

J = number of tasks in operation,

K = number of operation hours per day,

T = number of operating days in scheduling horizon,

c_i = wage rate for employee i ,

Dh_t = Demand maximum for workers on day t , by task hour,

Dl_t = Demand minimum for workers on day t , by task hour,

e_{jkt} = number of workers scheduled in excess of the number required for task j in hour k of day t ,

h_i = number of tour hours targeted for worker i ,

Q_j = workers qualified for task j ,

r_{jkt} = number of workers required for task j in hour k of day t ,

$s_{\min}$ = allowed minimum number of hours in any shift,

$s_{\max}$ = allowed maximum number of hours in any shift,

T_i = tasks for which worker i is qualified,

U_{it} = worker i 's available hours in day t ,

u_{it} = worker i 's first available hour in day t ,

V_i = worker i 's available days (partial or whole),

v_{it} = worker i 's last available hour in day t ,

$w_{\max}$ = maximum number of days worker can be assigned in scheduling period,

$x_{ijkt} = 1$ if worker i is scheduled to perform task j in hour k of day t ,

$= 0$ otherwise, and

$y_{it} = 1$ if worker i is scheduled to work any hours in day t ,

$= 0$ otherwise.

The objective of the work tour scheduling problem with limited worker interchangeability is to minimize the total man-hours of overstaffing and to match as closely as possible the workers desired hours and time off. So the objective function is

minimize:

$$W_1 * \sum_{jkt} e_{jkt} + W_2 * (\sum_{ijkt} x_{ijkt} - h_i).$$

In Loucks' formulation, the two terms in the objective function were stated as goals in his goal programming formulation. The weights W_1 and W_2 will be chosen to ensure that this integer program behaves as his goal program does. The first term has first priority and as such will have a large W_1 value, W_2 will have a relatively smaller value.

Subject to:

$$\sum_{i \in Q_j} x_{ijkt} - e_{jkt} = r_{jkt} \text{ for all } j, t \in V_{it}, k \in U_{it} \quad (1)$$

Constraint set (1) states that the labor schedule must meet, but preferably not exceed, the staffing requirement for each task in each operating hour of the week.

$$\sum_{j \in T_i} \sum_{k \in U_{it}} x_{ijkt} \leq 1 \text{ for all } i \text{ (where } |T_i| > 1, t \in V_i \text{ and } k \in U_{it} \quad (2)$$

Constraint set (2) restricts a worker assignment to not more than one task per hour.

$$\sum_{j \in U_{it}} \sum_{k \in U_{it}} x_{ijkt} - s_{\min}(y_{it}) \geq 0 \text{ for all } i \text{ and } t \in V_i, \quad (3)$$

$$\sum_{j \in T_i} \sum_{k \in U_{it}} x_{ijkt} - |U_{it}| y_{ik} \leq 0 \text{ for all } i \text{ and } t \in V_i, \quad (4)$$

Constraint sets (3) and (4) confines a worker's shift in any workday to be greater than or equal to the minimum allowable duration, $s_{\min}$. Constraint set (4) controls the value (0 or 1) of y_{it} .

$$\sum_{j \in T_i} \sum_{k \in U_{it}} x_{ijkt} \leq s_{\max} \text{ for all } i \text{ and } t \in V_i, \quad (5)$$

Constraint set (5) restricts the worker's shift length to be less than or equal to the maximum allowable shift length, $s_{\max}$.

$$\sum_{t \in V_i} y_{it} \leq w_{\max} \text{ for all } i \text{ where } |V_i| > w_{\max}, \quad (6)$$

Constraint set (6) restricts the number of days worked per week for any worker to less or equal to than the allowable maximum, $w_{\max}$.

$$\sum_{j \in T_i} (x_{ijkt} - x_{ij(k+1)t} + x_{ij(k+2)t}) \leq 1 \text{ for all } i$$

where $|U_{it}| > s_{\min}$, $k = (u_{it}, \dots, v_{it} - s_{\min} + 1)$, and $t \in V_i$. (7)

The last constraint set (7) ensures that a worker's shift will be contiguous.

3.3. ASSUMPTIONS OF FORMULATION

There are several assumptions inherent in this formulation. The first assumption is that the forecast of worker requirements for the scheduling period is given; there are no provisions in the algorithm or formulation to develop a forecast. The implications of this assumption are two-fold; 1) performance differences between algorithms will be due to the suitability of the algorithm to the problem and not superior performance of forecasting methods and 2) the staffing requirements are deterministic, that is, the number of workers needed for each task in each hour is assumed to be known.

Second, the solution is constrained by limited worker interchangeability, and specific task assignments. This assumption is derived from application to service businesses. It makes the problem larger and more difficult to solve.

Third, the schedule encompasses less than 24 hours, and no accommodation has been made for shifts to overlap days. Reformulation of the problem would be necessary to accommodate 24 hour schedules.

Fourth, the time period length has been set at 1 hour. There are two reasons for this length. 1) A 1 hour scheduling period will allow the manager enough flexibility to schedule lunch and breaks during lulls in customer demand without having to explicitly model such break times. Reducing the time period would allow us to model the lunch and rest breaks, but since demand is variable, even though it is assumed to be fixed, modeling with reduced time periods may cause reductions in customer service. 2) Managers may not care to adjust the staffing levels more often than once per hour, more frequent changes might reduce productivity.

Fifth, this formulation assumes that the operating days are short enough to ensure that there is sufficient rest time between shifts. So a worker may "close" the facility one day and "open" the next day.

Last, the model allows customer or client demand patterns that are independent from day-to-day. This assumption or feature of the model is a necessity for 7 day operations.

CHAPTER 4, GENERATION OF INITIAL FEASIBLE SOLUTION

4.1. INTRODUCTION

There are at least three approaches to the design and implementation of iterative improvement algorithms. The first method is to generate an initial solution, possibly feasible, using random methods, which is then acted upon by the improvement algorithm. The second method is to generate an initial solution that is known to be feasible, which is then acted upon by the improvement algorithm. The third method is to construct and improve the solution simultaneously, this is the method used by Loucks [1987]. An examination of the research literature of both simulated annealing and tabu search revealed that the prevailing approach is to generate an initial solution using random methods, which is then acted upon by the respective algorithm. The two predominant reasons for previous researchers using this method are these. First, according to van Laarhoven and Aarts [1989], simulated annealing is not sensitive to initial solution or configuration. No such results are available for tabu search. Second, the problems considered by previous research of both methods, simulated annealing and tabu search, are such that it is a relatively simple matter to use random methods to generate an initial, feasible configuration, because the problems previously considered have few constraints.

The use of a random approach was deemed inappropriate in this research for several reasons. First, no known, or easily conceived, methods are known for generating an initial, feasible configuration randomly, for this highly constrained problem. Second, after some investigation, it was determined that the algorithms, as implemented here, could not always (less than 50% of the time) remove the infeasibilities present in an infeasible initial configuration. Simulated annealing, given enough run time, tries all possible configurations and so would remove all infeasibilities. Tabu search, being a deterministic algorithm does not try all possible configurations and may never remove an infeasibility. Last, it was found that estimating the change in the objective function was an excellent way to improve the speed of the algorithms, as explained in the following chapters. It is relatively simple to estimate the effect of a feasible transition or move on the objective function, but it is not easy to estimate the effect on the objective function of causing an additional infeasibility, or partial removal of an infeasibility.

Previous authors have used the output from a previously developed algorithm as the input to simulated annealing and tabu search. One reason this approach was not taken is because Loucks' heuristic was shown to produce optimal results for a significant proportion of the problems tested by Loucks [1987]. It seemed foolish to apply an improvement algorithm to a heuristic that produces optimal solutions often. Additionally, Loucks presents evidence that his algorithm will not solve all problems produced by the problem generator he used. So, the possibility existed that improvements could be made in the number of problems that could be solved by developing a new method for generating initial, feasible configurations. These considerations prompted development of an initial configuration generator that, upon successful completion, produces feasible configurations or solutions.

4.2. STRATEGY OF INITIAL FEASIBLE SOLUTION GENERATOR

Several objectives and strategies for the development of an IFS generator were identified as a result of the considerations above. The objectives for the IFS generator were to have the algorithm work reliably and to produce a feasible solution quickly. This algorithm works more reliably than the only other method known for this problem, Loucks' heuristic. The method is also reasonably fast, it normally takes less than 10 seconds on a personal computer (33 MHz, 80386 processor) to produce the feasible solution. The strategies are detailed below.

4.2.1. *"Fat" solution, vs. "Lean" solution*

One consideration that guides the development of the IFS generator is that of the type of solution to be produced. In this case the strategy was to develop the IFS generator in such a way that a "fat" solution is produced. This means that there is not much attempt to keep from over scheduling task hours. This is direct contrast to Loucks' heuristic, which attempts to minimize over scheduling task hours at every opportunity. The result of this strategy is that the improvement algorithms, simulated annealing and tabu search, have more possible configurations to generate and choose from, hopefully improving the prospects of a good final solution. This strategy is also responsible for the improved ability of the IFS generator to produce feasible configurations.

4.2.2. *Margin on task hours*

Some method must be used to determine the priority in which to schedule task hours. While there are several methods to assign priority, such as starting at the beginning of the week and proceeding in a linear fashion to the end, the priorities could be assigned randomly, and etc., the method used in this research was to calculate the margin for each task hour and use that margin as the basis for assigning priority. The margin of a taskhour is defined as the difference between the number of workers required for the taskhour and the number of workers available and qualified for that taskhour. These margins are sorted from lowest to highest and the taskhours are assigned the highest priority from lowest to highest margin.

This method of assigning priority based on margin has the advantage of ensuring that when the scheduling assignments are made, those workers that were identified as available at the beginning of the algorithm are available when they are needed for assignment to the taskhour. The idea of margin is attributable to Loucks [1987].

4.2.3. *Choice of worker for task hour*

Another strategic issue is the order in which workers are chosen for scheduling in a particular taskhour. When a pool of workers for a particular taskhour is identified, some method must be used to pick among the workers in that pool for assignment to that taskhour. The method used in this research is to create a worker priority score. This score, then, is used to differentiate among the workers. See figure 4-1.

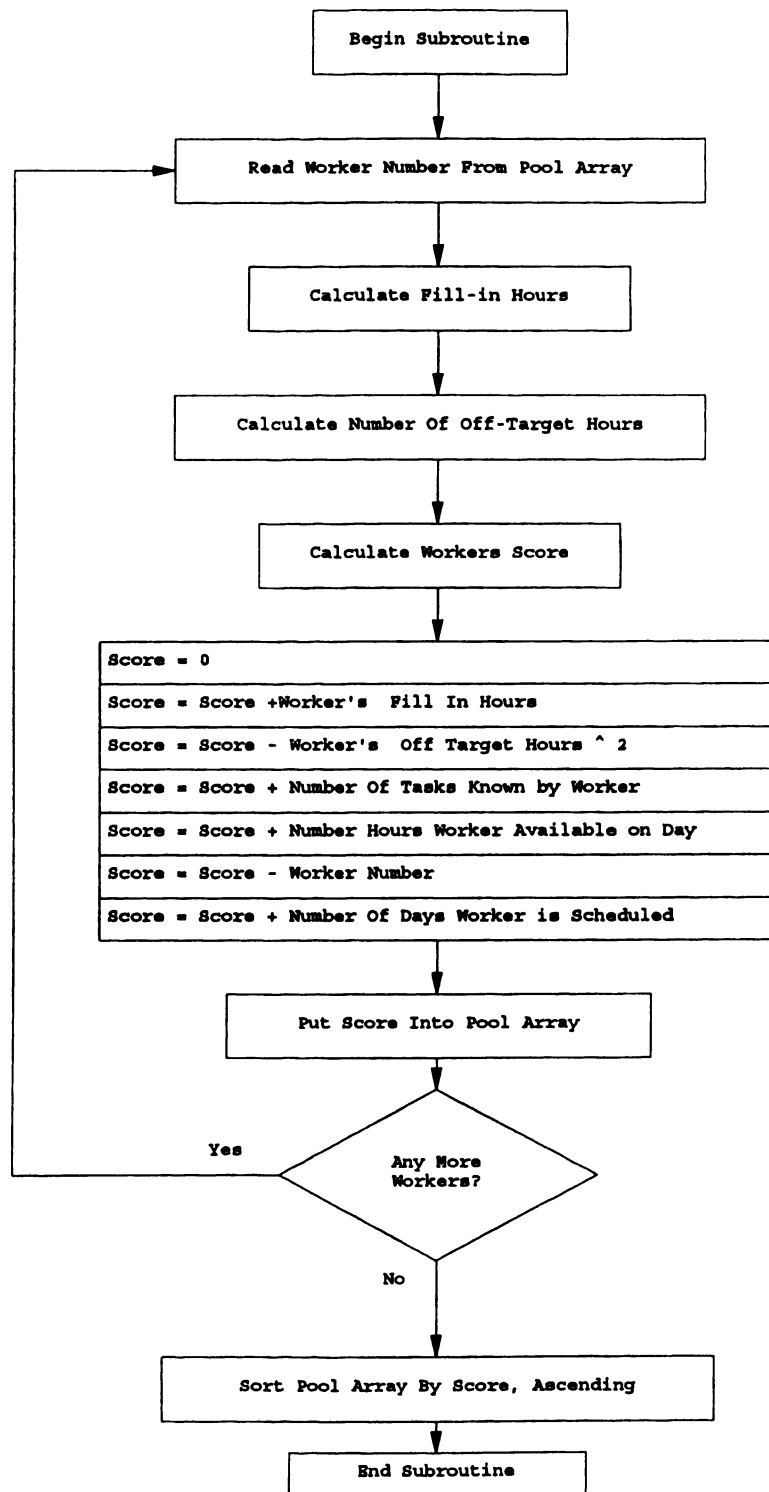


Figure 4-1, Selecting a Worker for Assignment

The first consideration in determining a worker score is the worker's scheduling flexibility. This flexibility is determined by the number of tasks known by the worker, the workers availability, and the number of days already scheduled. A worker is more flexible when that worker knows more tasks, when that worker is more available, and when that worker is scheduled for fewer days. Since any worker that is available and qualified may be chosen for assignment to the taskhour at hand, it makes intuitive sense to select the least flexible workers for assignment first, saving the more flexible workers for later in the assignment process when greater flexibility is needed.

The other consideration in choosing a worker for assignment is that of hours previously assigned to the worker. The method for calculating the score attempts to minimize the number of fill-in hours and the number of off-target hours.

The score considers the number of fill-in hours between the hour to be scheduled and the hours already scheduled, the number of hours a worker is under or over targeted hours, the number of tasks known by the worker, the number of hours the worker is available for the day, and the number of days the worker is already scheduled. The equation for calculating the score is:

$$\text{Score} = (\text{Number of Fill-In Hours}) - (\text{Number of Hours Off Target})^2 + (\text{Number of Tasks Known}) + (\text{Number of Hours Available on Day}) + (\text{Number of Days Already Scheduled}) + (\text{Worker Number})$$

The reader will note that the equation for worker score puts preference on lower worker flexibility and endeavors to minimize the number of fill-in hours and off target hours. The worker number is added to the score to alleviate potential ties among workers.

4.2.4. Elimination of Constraint Violations

The methods used to produce an initial schedule typically produce a number of constraint violations. In order to produce a feasible solution, these constraint violations must be eliminated. As elucidated in chapter 3, there are seven types of constraints in the mathematical formulation of this problem. Seven separate methods were developed to fix the various constraint violations. See figure 4-2.

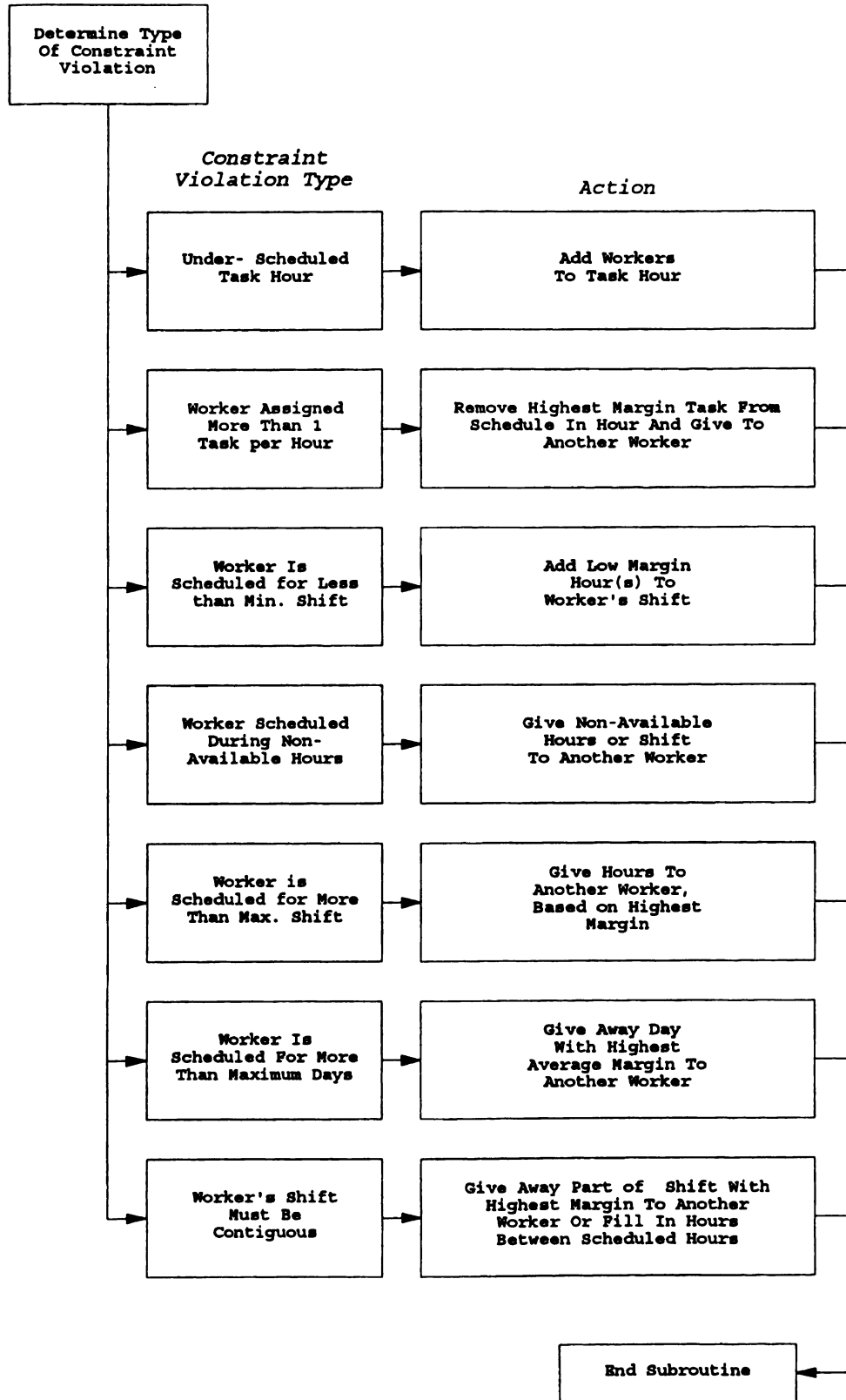


Figure 4-2, Eliminate Constraint Violations

Constraint Type 1: The labor schedule must meet or exceed demand requirements expressed in man-hours needed per taskhour. For constraint violations of this type the correction is to add a worker(s) to the taskhour in violation. If the Average Worker Utilization is low enough, this is an easy correction to make. If the AWU is too high, the procedure fails, and the algorithm fails to find a feasible solution.

Constraint Type 2: A worker must not be assigned to more than one task in an hour. Given the data structure of the programs representing the implemented algorithm, it is impossible to have a constraint violation of this type.

Constraint Type 3: A worker must not be scheduled for less than the minimum shift in any day. A rectification of this type of constraint violation requires that an hour(s) be added to one end of the shift in violation. The method used examines the taskhours on either end of the shift that the worker is available and qualified to perform, chooses the taskhour(s) with the lowest margin, and assigns the worker to that taskhour(s).

Constraint Type 4: The labor schedule must not exceed worker's available hours. In this case, the worker has been inadvertently assigned to hours that the worker is not available to work. If the shift length of the worker for the day, less the number of hours scheduled in error, leaves a shift equal to or greater than the minimum shift then the hours in error are given to another worker, available and qualified, of course. If no worker can take the add on hours, the another worker is found who is available and qualified and not scheduled for the day. This worker is then given the hours in error and a shift of minimum length is created around the hours given to the second worker. In the case that after subtracting the hours in error, the remaining shift is shorter in length than the minimum shift, the entire shift is given to another worker, who is available and qualified.

Constraint Type 5: A worker must not be scheduled for more than the maximum shift in any day. In this case, the worker has been inadvertently assigned to hours that

extend the shift beyond the maximum length. If the shift length of the worker for the day, less the number of hours scheduled in error, leaves a shift equal to or greater than the minimum shift, then the hours in error are given to another worker, that is available and qualified. If no worker can take the add on hours, the another worker is found who is available and qualified and not scheduled for the day. This worker is then given the hours in error and a shift of minimum length is created around the hours given to the second worker. In the case that after subtracting the hours in error, the remaining shift is shorter in length than the minimum shift, the entire shift is given to another worker, who is available and qualified.

Constraint Type 6: A worker must not be scheduled to work more than the maximum number of days per week. In case of a constraint violation of this type, the shift(s) with the highest average margin is given to another worker(s). The reason the highest average margin shift is given away is that it is easier to find a worker to take the shift for higher margin task hours than for lower margin task hours.

Constraint Type 7: The worker's shift must consist of contiguous hours. This type of constraint violation has the worker working several hours in a day, but with a gap of a non-scheduled taskhour(s) somewhere in the shift. If the total shift length (including the non-scheduled taskhours), is less than the maximum shift, then the gap is filled in by adding taskhours with the smallest margin. If the aforementioned shift length is longer than the maximum shift, then the part of the shift with the highest average margin is given to another worker.

Experience with the constraint violation correction routine showed that for some constraint violations, especially violation of constraint 7, that not only was the intended violation corrected, but that another violation(s) was corrected at the same time. For example, let us suppose that the constraint violation was of constraint 7. If the shift length was over the maximum shift length, then there would be a violation of constraint 5 also associated with the same scheduling error. Correction of the violation of constraint 7

would fix the violation of constraint 5, but not vice versa. After some experimentation, the order that the constraints were corrected, 6-1-5-4-3-7 (by number of constraint type), was found to be the most effective in correcting violations. It was also found that often, two iterations through the violations in the above order would completely correct the constraint violations where one iteration would not. This behavior may be attributable to the interactions between methods for fixing the violations.

4.3. STEP BY STEP EXPLANATION OF IFS GENERATOR

A brief description of the steps that comprise the initial feasible solution generator is given below. A flow chart, figure 4-3, is presented as illustration of the algorithm.

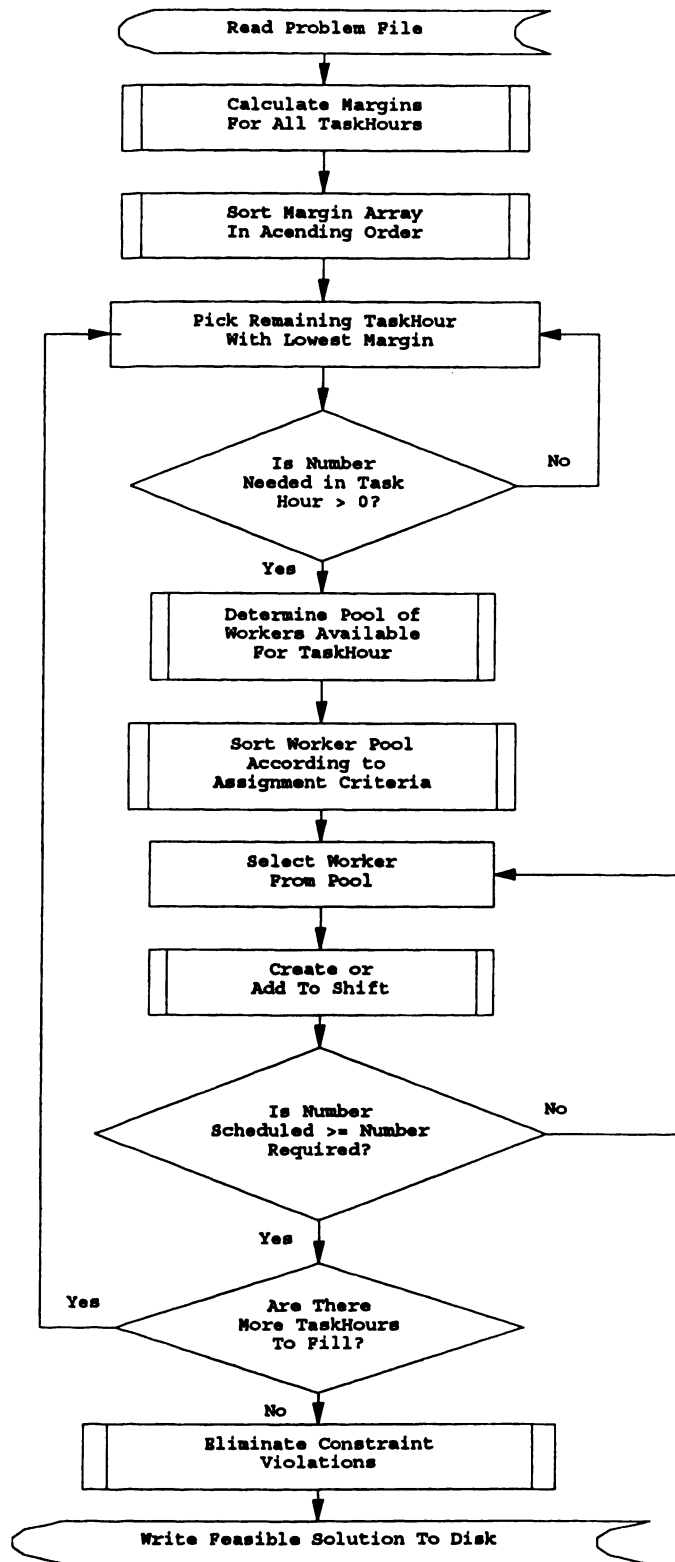


Figure 4-3, Procedure to Generate Initial Feasible Solution

4.3.1. *Explanation of Steps in Flow Chart*

The description of steps below follow the flow chart in figure 4-3. The titles of the sections below match the text in the boxes in the flow chart.

Step 1. Read Problem File. This step reads a problem from a disk (generated previously and stored on disk) and puts the problem data into the appropriate data structures.

Step 2. Calculate Margin For All Task Hours. Step 2 loops through all taskhours for the scheduling period, tallying the total number of workers available and qualified for each taskhour and calculating the margin for each taskhour according to the following equation:

$$\text{Margin}(\text{Hour}, \text{Task}, \text{Day}) = \text{NumberOfWorkersAvailable}(\text{Hour}, \text{Task}, \text{Day}) - \text{NumberOfWorkersRequired}(\text{Hour}, \text{Task}, \text{Day}).$$

These margins are stored in an array for future reference.

Step 3. Sort Margin Array in Ascending Order. The purpose of Step 3 is to sort the array in which the margins are stored in ascending order so that the taskhours may be selected from lowest margin to greatest.

Step 4. Pick Remaining Task Hour With Lowest Margin. This step begins a loop that, when finished, assigns workers to all taskhours. This loop steps through the sorted margin array from beginning to end, picking the taskhour with the lowest margin that has not yet been scheduled.

Step 5. Check To See if This Task Hour Needs to Be Scheduled. During the assignment of hours, performed in step 9, workers usually are assigned more to than one taskhour. As these assignments occur, some or all of the taskhour requirements in neighboring hours to the taskhour currently being scheduled are being filled. Occasionally early, and especially late in the algorithm, a taskhour may not need any further worker assignments. If a particular task hour's requirements are already met by

previous worker assignment, there is no need to execute the loop, and this step skips the remainder of the loop, returning processing to step 4.

Step 6. Create Pool of Available Workers for Task Hour. Once the taskhour is chosen for scheduling, workers must be found that are available and qualified for the taskhour. This step creates a pool of these workers by looking at all worker's availability and task qualifications, and putting the worker numbers of all workers available and qualified for that taskhour into an array for further processing.

Step 7. Sort Worker Pool According to Assignment Criteria, Figure 4-1. After the all available and qualified workers are identified, a method for choosing a worker from that pool must be used. While such methods as random selection might be used, the method used, shown in figure 4-2 and detailed above, selects workers based upon their flexibility and schedule thus far in the algorithm. This method considers the number of fill-in hours between the hour to be scheduled and the hours already scheduled, the number of hours a worker is under or over targeted hours, the number of tasks known by the worker, the number of hours the worker is available for the day, and the number of days the worker is already scheduled. As stated above the workers are assigned a score based on the aforementioned criteria, the scores set into the worker pool array and then the worker pool array is sorted into ascending order.

Step 8. Select Worker From Pool. Workers are selected from the pool, starting at the beginning of the array. If more than one worker is needed, the workers are taken in the order in which they were sorted in step 7.

Step 9. Create New Shift or Add to Existing Shift, Figure 4-4. This step assigns the worker chosen in step 8 to the taskhour. If the worker is already scheduled for to work on the day of the taskhour, the hour(s) between the taskhour and the previously scheduled shift are filled in with task assignments. If the worker was not scheduled to work on the day of the taskhour, then a shift is created around the taskhour being

scheduled. A more complete discussion is given above and the reader may also refer to figure 4-4.

Step 10. Check to See if Enough Workers have Been Scheduled for Task Hour.

Frequently more than one worker is required for a particular taskhour. This step determines if the taskhour requirements have been met. If not, this step redirects the processing to step 8.

Step 11. Check to See if there are more Task Hours to Schedule. The algorithm schedules workers into taskhours according to the order in the margin array. This step checks whether any taskhours remain to be scheduled. If there are more taskhours to be scheduled, this step directs the processing to step 4.

Step 12. Eliminate Constraint Violations, Figure 4-2. After all the taskhours have been scheduled, a schedule has been created for the problem read in step 1. Typically the schedule created by the preceding steps will violate the constraints detailed in chapter 3. This step rectifies constraint violations. A more complete description is given above, and a graphical description of the procedure to eliminate constraint violations is given in figure 4-2.

Step 13. Write Feasible Solution to Disk. This step saves the initial feasible solution in a disk file for improvement by the simulated annealing and Tabu search algorithms.

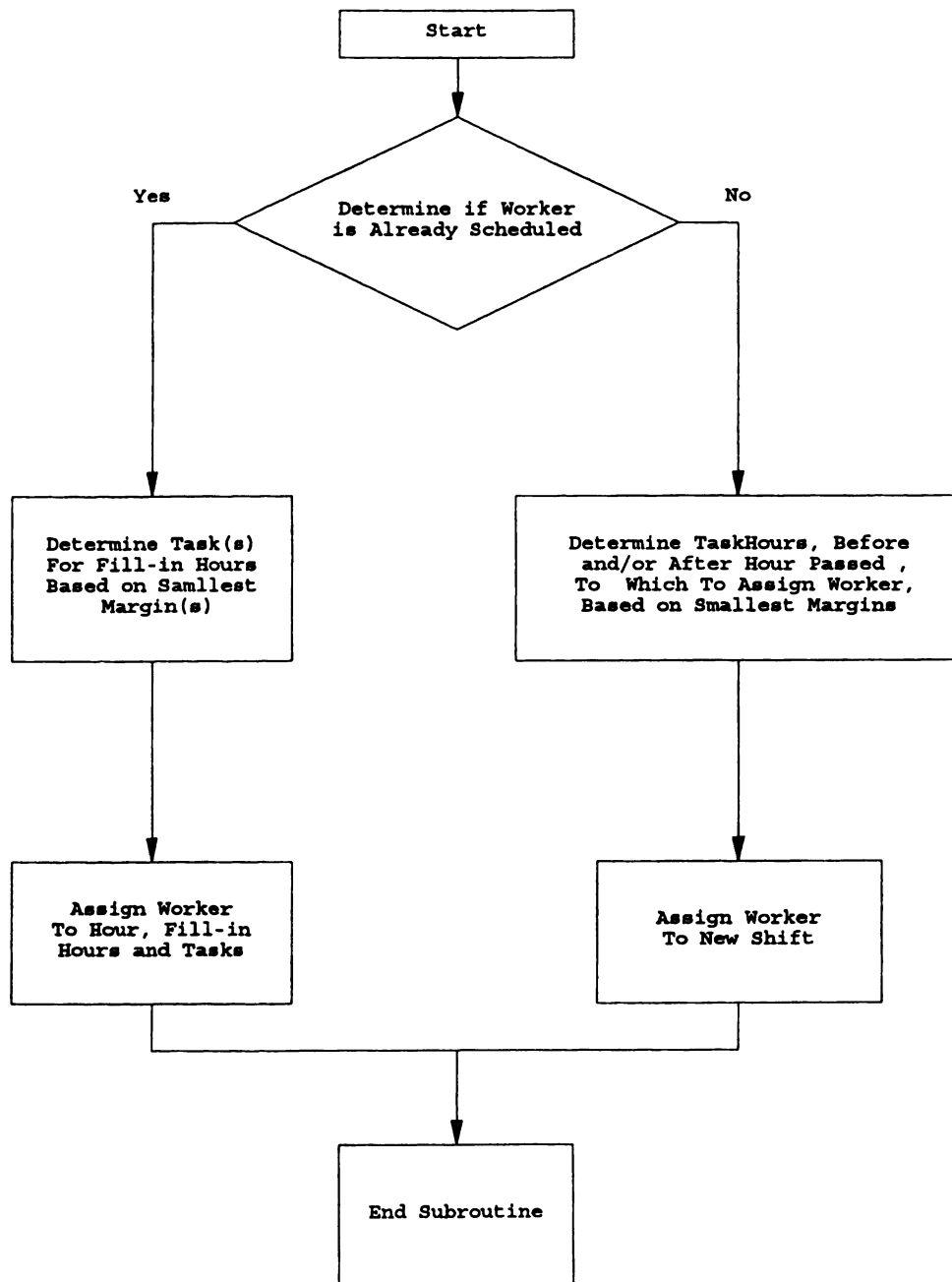


Figure 4-4, Create or Add to Shift

4.3.2. *Pseudo Pascal Description of IFS Generator Algorithm*

The following is a description of the algorithm in pseudo pascal.

Procedure Initial Feasible Solution

Begin

Initialize;

Read Problem File;

Repeat

Calculate Worker Margin For Task Hour;

Until All Task Hours Are Done;

Sort Margins in Ascending Order;

Repeat

Pick Remaining Task Hour with Lowest Margin;

If Number Needed > 0 then

Repeat

Determine Worker Pool Available For Task Hour;

Sort Worker Pool According to Assignment Criteria;

Select Worker From Pool;

If Worker Is Already Scheduled Then

Add To Current Shift;

ElseIf

Create Shift Around Task Hour;

Endif;

Endif;

Until Number Scheduled >= Number Needed For Task Hour;

Until All Task Hours Have Been Scheduled;

Eliminate Constraint Violations;

Write Feasible Solution To Disk;

End:

4.3.3. Conclusion

An algorithm for generating feasible solutions to the work tour scheduling problem with non-interchangeable workers, as formulated in chapter 3, has been presented. As discussed before, the performance of this algorithm exceeds previous algorithms in terms of the number of problems for which it is able to produce a feasible solution. This initial configuration generator produces initial, feasible configurations on problems for which Loucks' heuristic produces an abnormal stop. The initial, feasible configuration (IFS) generator produces feasible configurations on all problems that Loucks' heuristic would run to completion, but Loucks' heuristic would not run to completion on all the problems that the IFS generator would successfully run to completion.

CHAPTER 5 - TABU SEARCH: THEORY AND IMPLEMENTATION

5.1. INTRODUCTION

Tabu search is a strategy to solve combinatorial optimization problems that uses an adaptive mechanism to guide other algorithms, such as linear programming or other specialized heuristics, to avoid the limitations of local optimality. Tabu search was initially developed by Glover [1986] and has since been the subject of considerable research.

5.1.1. Overview of Tabu Search

Tabu search may be considered a supervisory heuristic imposed on another heuristic in order to more rapidly solve combinatorial optimization problems. In general, tabu search uses another heuristic(s) to generate a number of potential intermediate solutions (moves) from which it picks the "best". This leads to rapid arrival at a local or global optima. The approach endeavors to overcome local optima by imposing a strategy of forbidding certain moves and forcing moves away from the local optima. Reversal of these moves or directions are forbidden or tabu for a number of iterations so that cycling

might be avoided. The flow chart and related discussion following will make the operation of the tabu search algorithm clear. The tabu search described below is an adaptation of that presented by Glover [1989].

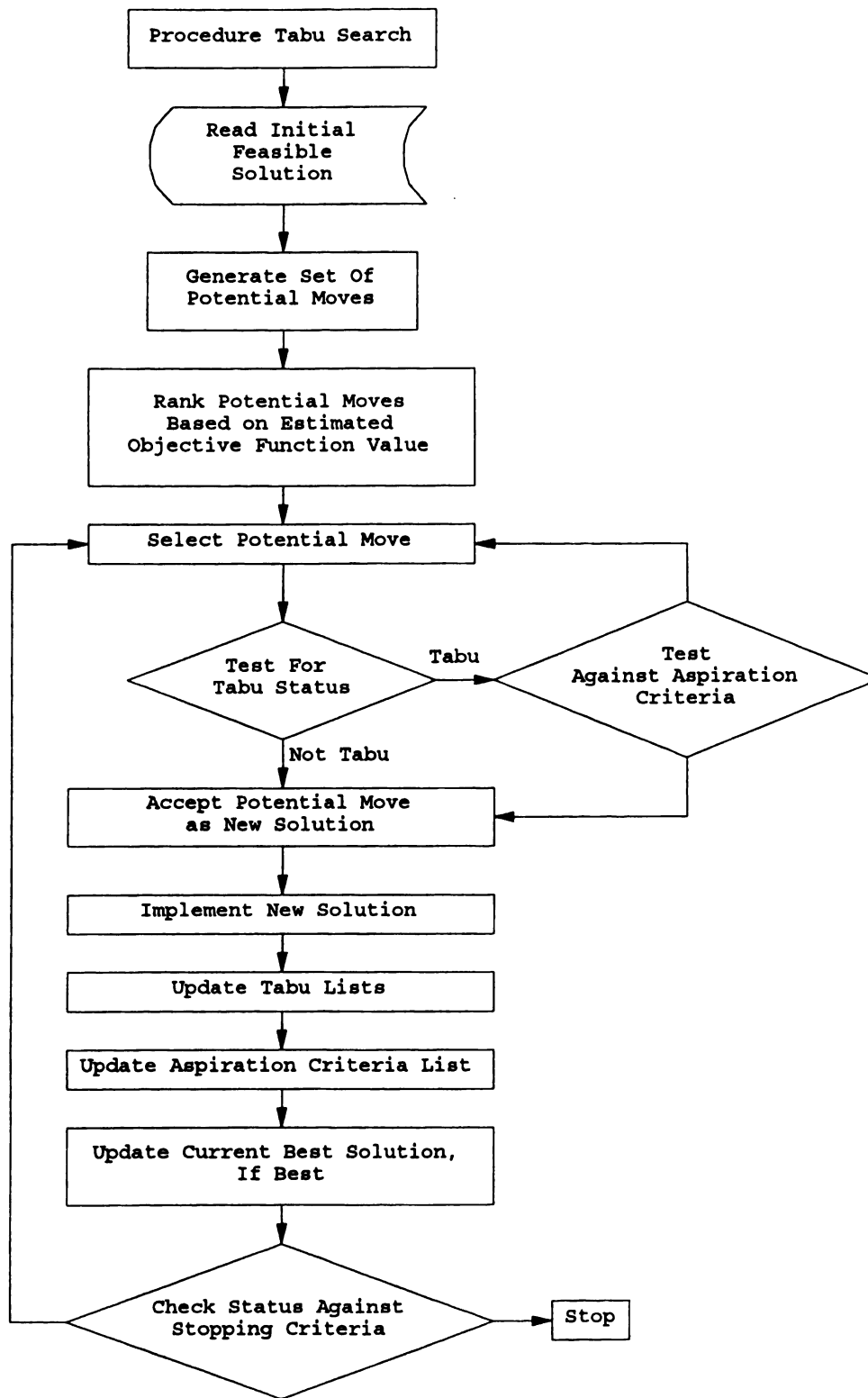


Figure 5-1, Tabu Search Algorithm

As shown in the flow chart, the first step is to generate an initial feasible solution. This initial solution is generated using the procedure explained in chapter 4. The current best solution is set to the initial solution. The next step is to generate a set of potential moves. As mentioned before, these moves can be generated by other heuristic methods, by optimization methods used on partitions of the problem, etc. The move generators used in this investigation are detailed below. The potential moves are then sorted or ranked. Ranking of potential moves might be done in a number of different ways, such as, on objective function value, estimates of objective function value, change in objective function value, or some other criterion. The ranking of potential moves for this research is based on an estimate of the change in objective function value.

Testing the solution for tabu status is the next step. If the solution is not tabu, the solution is accepted as the new solution. The tabu list, aspiration criteria list (to be described shortly) and current best solution would be updated as necessary. If the prior solution was a global or local optima, it is very likely that all of the potential solutions in the current iteration would be worse than the current best solution. In this situation, there are two items of interest; First, this is an example of the algorithm moving away from an (local) optima in search of a better (local or global) optima and Second, in this situation the current best solution as well as the aspiration criteria would not be updated.

If the solution is tabu, then the solution is tested against the aspiration criteria. An aspiration criteria is a change in objective function value of sufficient magnitude to override the tabu status. The aspiration criteria is set in such a manner so as to prevent cycling of the algorithm. For example: if the algorithm was moving away from an optima, the aspiration criteria must be set in such a way that it would not override the tabu status and cycle back to the previously found optima. The aspiration criteria used in this implementation of tabu search is to require a move to produce a solution better than the current best solution, in order to override tabu status. Glover [1989] covers aspiration criteria in considerable detail. If the solution meets or exceeds the aspiration criteria, tabu

status is overridden and the tabu list, aspiration criteria and current best solution would be updated as necessary.

At each iteration, the algorithm checks against the stopping criteria. The stopping criteria used in this research was based on two rules. First, a limit of one thousand iterations was set, based on trial runs of the algorithm. Glover [1989] recommends that such an overall iteration limit be set. This limit seems to be somewhat past the "point of diminishing returns". The second rule used was to stop the algorithm if there had been no improvement in the current best solution in 300 iterations. Again this rule is based on Glover [1989] and trial runs of the algorithm.

5.1.2. *Description of Tabu Search Algorithm In Pseudo-Pascal*

```

Procedure Tabu Search
Begin
  Initialize;
  M := 0;
  repeat
    Generate {Potential Solutions}
    Sort {Potential Solutions}.....(based on Objective Function Value,
                                          Best First)

    repeat
      N := 0;
      TabuTest (Solution N, Tabu Status);
      If (Solution N) = Tabu Then
        AspirationTest (Solution N, Aspiration Criteria);
        If (Solution N) > Aspiration Criteria Then
          Accept (Solution N);
      Else
        Accept (Solution N) ;
        Update (Tabu List, Aspiration Criteria, Current
                  Best Solution)
        Accept Flag := 1;
      Endif;
    Until Accept Flag = 1;
  Until Stop Criterion = True;
End.

```

5.1.3. Example of Tabu Search Algorithm

Consider maximizing the function $f(x) = x^2$, where x is an integer and permitted to vary from 0 to 31. For this example the move generator will be a random number generator that generates integers from 0 to 32. Furthermore, we will arbitrarily set the number of moves to be generated at 4 and the tabu list length at 3.

Initialize: We select an initial solution at random, say $x=3$. The objective function value is 9. We set the current best solution to $x=3$. The tabu list is empty as is the aspiration criteria list.

Generate: We pick 4 potential solutions from our random number generator, 1, 8, 19, and 6.

Sort: We sort the values from high to low, 19, 8, 6, 1.

TabuTest: We test the 1st solution (19) for tabu status. Since the tabu list is empty the solution is not tabu.

Accept: We accept the solution (19) as the new solution.

Update: We update the tabu list, (it includes 19 now), the aspiration criteria list, (a solution must beat 19), and the current best solution (set to 19).

Generate: We pick 4 potential solutions from our random number generator, 12, 7, 19, and 5.

Sort: We sort the values from high to low, 19, 12, 7, 5.

TabuTest: We test the 1st solution (19) for tabu status. Since the tabu list contains 19 the solution is tabu.

AspirationTest: The solution does not meet the aspiration criteria that the solution must beat 19, so we go the next potential solution (12).

TabuTest: We test the 2nd solution for tabu status. The tabu list does not contain 12 so the solution is not tabu.

Accept: We accept the 2nd solution as our new solution.

Update: We update the tabu list (it includes 12 and 19 now). The aspiration criteria list and the current best solution remain unchanged.

The algorithm would continue until a stopping criteria is met.

Because the example is simple for clarity's sake, the aspiration criteria is almost trivial in its simplicity. Normally, the aspiration criteria would allow a tabu variable back into the solution if two conditions are met; 1) the magnitude of the change in objective function value is sufficiently large, and 2) there is some change in the solution, i.e. the solution has a different variable set than it had when the tabu variable was in the solution previously or equivalently, the solution is not identical to any previous solution.

5.2. TABU SEARCH IMPLEMENTATION ISSUES

5.2.1. Tabu List Types

The basic idea of a tabu list is to restrict previous solutions (or some aspect of a previous solution) from being used for a specified number of iterations. Each move that is selected as a potential move is checked against the tabu list. If that move is found on the tabu list, that potential move is eliminated from consideration, and another potential move is chosen. In prior implementations tabu search, Glover [1989], Knox [1990], used tabu lists that prohibit reentry of a solution based on move type. Using the traveling salesman problem as an example (that is what Knox researched), if a move had changed the connection between two edges, the tabu list would not accept a move that reconnected the two edges in the original order, for a certain period of time (based on the length of the tabu list). In the implementation of tabu search for the work tour scheduling

problem with non-interchangeable workers, the move types used precluded the use of such conceptually simple tabu lists.

The tabu lists that are kept for this implementation of tabu search are based on the types of move generators used. There are three types of move generators used; ones based on hours (i.e. GiveAnHour, IncreaseOverScheduling, GiveAwayPartShift and ReduceOverScheduling) , ones based on shifts (i.e. GiveADay, TradeTours, and CrossTradeTours) and hybrids of the two (i.e. GiveAwayMinShift and GiveAwayPartShift). As mentioned before, in prior research, tabu lists were used for each different type of move. In this research, however, separate lists for each type of move would be very cumbersome to maintain and check for tabu status. For these reasons, four lists were created that would preserve the idea of tabu lists and also allow for easy maintenance and checking. These four lists track hours deleted from a workers schedule on a particular day, hours added to a worker's schedule on a particular day, shifts deleted from a workers schedule on a particular day and shifts added to a worker's tour for a particular day. As a particular move is implemented, the appropriate list is updated with the relevant information from the move.

These four tabu lists were found to be adequate to prevent cycling, and to force the algorithm out of local optima, which are the two objectives of using tabu lists.

5.2.2. Aspiration Criteria

The tabu lists, when properly specified, prevent the tabu search algorithm from cycling. As was mentioned before, this is accomplished by prohibiting a previous solution or some aspect of a previous solution from reentering the solution, for some number of iterations. Aspiration criteria allow parts of a previous solution back into the

so

G

ir

fu

so

P

a

5

th

d

th

5

5

L

solution, providing that the reentry of this part of the solution does not cause cycling. Glover [1987] suggests that aspiration criteria allow a tabu variable, or solution, back into the solution if two conditions are met; 1) the magnitude of the change in objective function value is sufficiently large, and 2) there is some change in the solution, i.e. the solution has a different variable set than it had when the tabu variable was in the solution previously or equivalently, the solution is not identical to any previous solution. These are the aspiration criteria used in this investigation.

5.2.3. Selection of Parameter Values

The only parameter value to be determined is the length of the tabu list. Although there is some guidance given in the literature [Glover, 1989], the tabu list size is usually determined by experimentation. Experimentation was used in this research to determine the tabu list length. Results of the experimentation is shown in Figure 5-2, and in Table 5-1.

5.2.4. Comparison of Results for Different Tabu List Lengths.

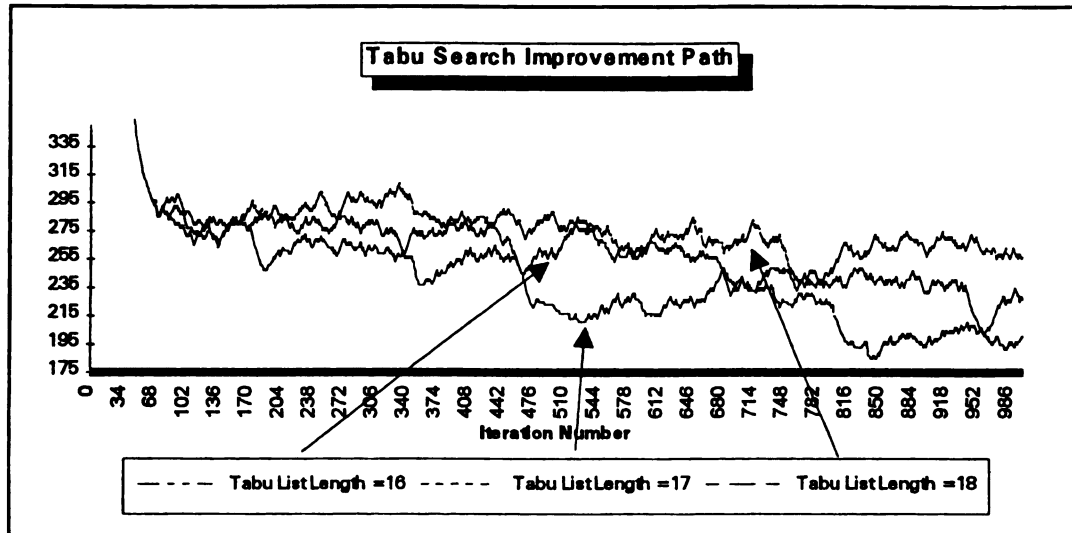


Figure 5-2, Tabu Search Improvement Path for Different Tabu List Lengths

As can be seen from Figure 5-2, the tabu list length of 17 seems to outperform either 16 or 18. Table 5-1 shows that for other lengths of the tabu list, the best objective function value is always higher than when the length is 17.

Table 5-1, Comparison of Tabu List Length and Objective Function Value

Tabu List Length		7	11	13	15	16	17	18	19	30
Minimum		221	225	225	209	203	185	233	215	219

As a demonstration that the tabu list length choice is important in eliminating the threat of cycling, Figure 5-3 is offered. Cycling is clearly evident after about 600 iterations.

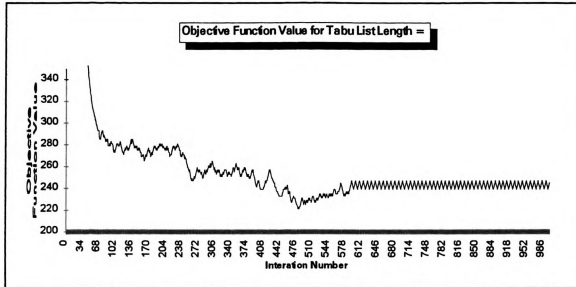


Figure 5-3, Improvement Path for Tabu List Length = 7

5.2.6. *Generation of Moves or New Solutions*

The generation of moves in the implementation of tabu search has been approached in a problem dependent manner in the past [Glover, 1989]. Moves or potential solutions have been generated by mechanisms based on salient aspects of the problems to be solved or by utilizing portions of heuristics previously applied to the problem [Knox, 1987]. As a result, there are no "generic" move generators, so for the labor scheduling problem in this research, the methods below are used for generating moves. In the computer program written to implement tabu search for this research, each of the move generators was contained in a separate subroutine and each of those subroutines are explained below.

There are several strategies that guided the development of these move generators. First, each of the move generators was designed to produce a move without

creating constraint violations. The reasons behind this strategy are; 1) There is no good way to estimate the change in objective function without assurance that the move is feasible, and 2) If moves that produce constraint violations are allowed, there is no assurance that a constraint violation will be rectified without a routine with which to fix that violation. Therefore, move generators were designed so that no constraint violations were generated along with the move.

The second strategy utilized was to design move generators that gave the algorithms the greatest opportunity to arrive at a good solution. While adherence to such a strategy, with a given set of move generators, probably cannot be proven, it is posited that the set of move generators utilized in this research do, in substantial measure, adhere to this strategy.

The last strategy utilized for designing move generators for this research was to use moves that had the same underlying concepts for both tabu search and simulated annealing. This strategy was rigidly adhered to because of the overriding policy of giving no advantage to one algorithm over another.

5.2.7. Move Generators

Those subroutines that implement the move generators are explained in the following sections. As each of the move generator subroutines are executed, all possible potential moves of the respective type are generated. All these potential moves are then evaluated and one is chosen for implementation.

Reduce OverStaffing. This subroutine produces potential moves or transitions that eliminate an hour from one end of a worker's ($worker_1$) shift. A flow chart of this subroutine is given in figure 5-4. First, a list of all taskhours that are overstaffed is generated. Then, for each overstaffed taskhour, a list of workers ($worker_2$) is generated that meet the following criteria: 1) $worker_2$ is scheduled for that taskhour, 2) the taskhour is either the beginning or end of $worker_2$'s shift on that day, and 3) $worker_2$ is scheduled for more than the minimum shift on that day. Finally, this list is combined with the lists from all other overstaffed taskhours and added to the potential move list for further processing.

The motivation behind this move is to reduce the cost of the schedule, and secondarily to increase the availability, and hence flexibility of the worker, so that other moves might be more easily made.

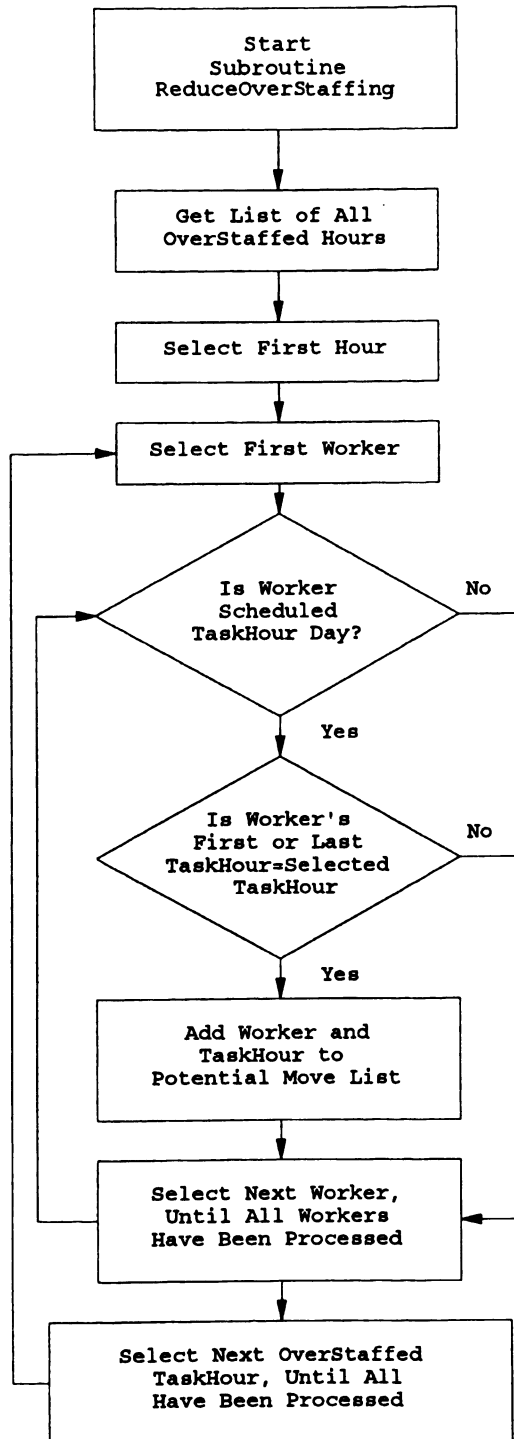


Figure 5-4, Reduce Over Staffing

Increase Over Staffing. This subroutine produces potential moves that add an hour to one end of a worker's shift. A flow chart for this subroutine is given in figure 5-5. For each day, potential moves are generated by adding an hour to either end of the shift for workers that are scheduled. The particular task the to which the worker would be assigned is determined at the time of assignment, based on the taskhour margin, as discussed in chapter 4. All potential moves generated are combined with the potential move list for additional processing.

This move type is motivated by the idea that adding to a worker's shift might decrease his deviation from targeted hours, and may provide additional opportunities for the other move generators.

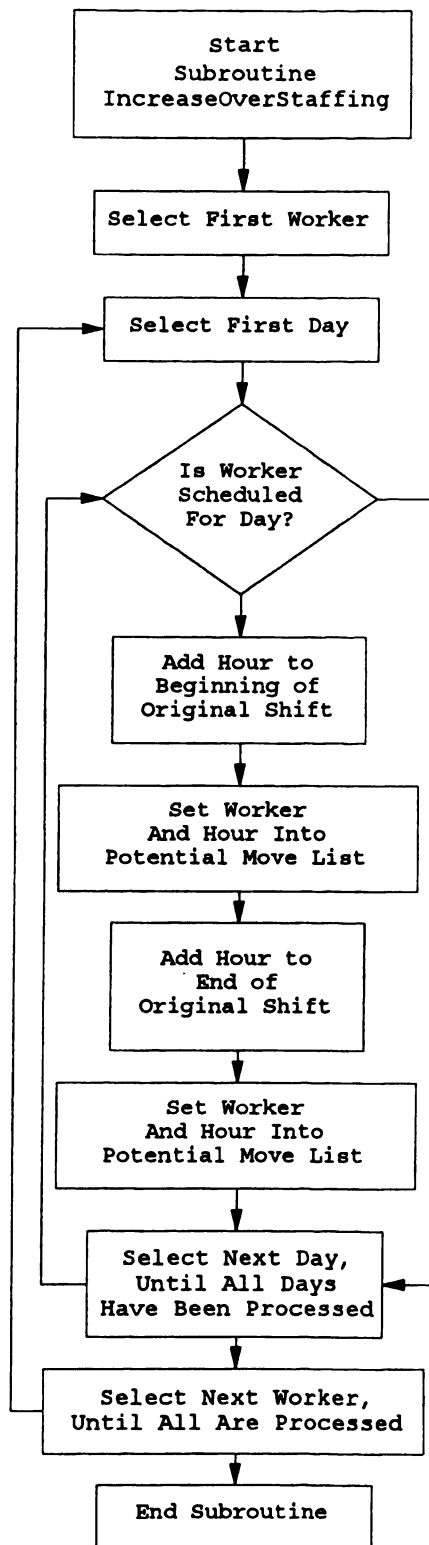


Figure 5-5, Increase Over Scheduling

Give An Hour To Another Worker. In this subroutine, potential moves are generated by moving a taskhour assignment from one worker to another worker. The flow char for this move is given in figure 5-6. First, the shifts for each worker are considered one-by-one and for each shift that is longer than the minimum shift, a second worker (worker₂) is sought for the first or last hour of the shift. A candidate for worker₂ will be considered only if the taskhour to be given away is either the hour immediately prior to worker₂'s existing shift, or the hour immediately following worker₂'s existing shift, and if worker₂ is available and qualified for the taskhour. All worker₂'s and taskhours so identified are added to the potential move list for further processing.

This type of move is meant to simultaneously decrease two worker's deviation from targeted hours. It may also decrease the cost of the schedule, if the two workers have different wage rates.

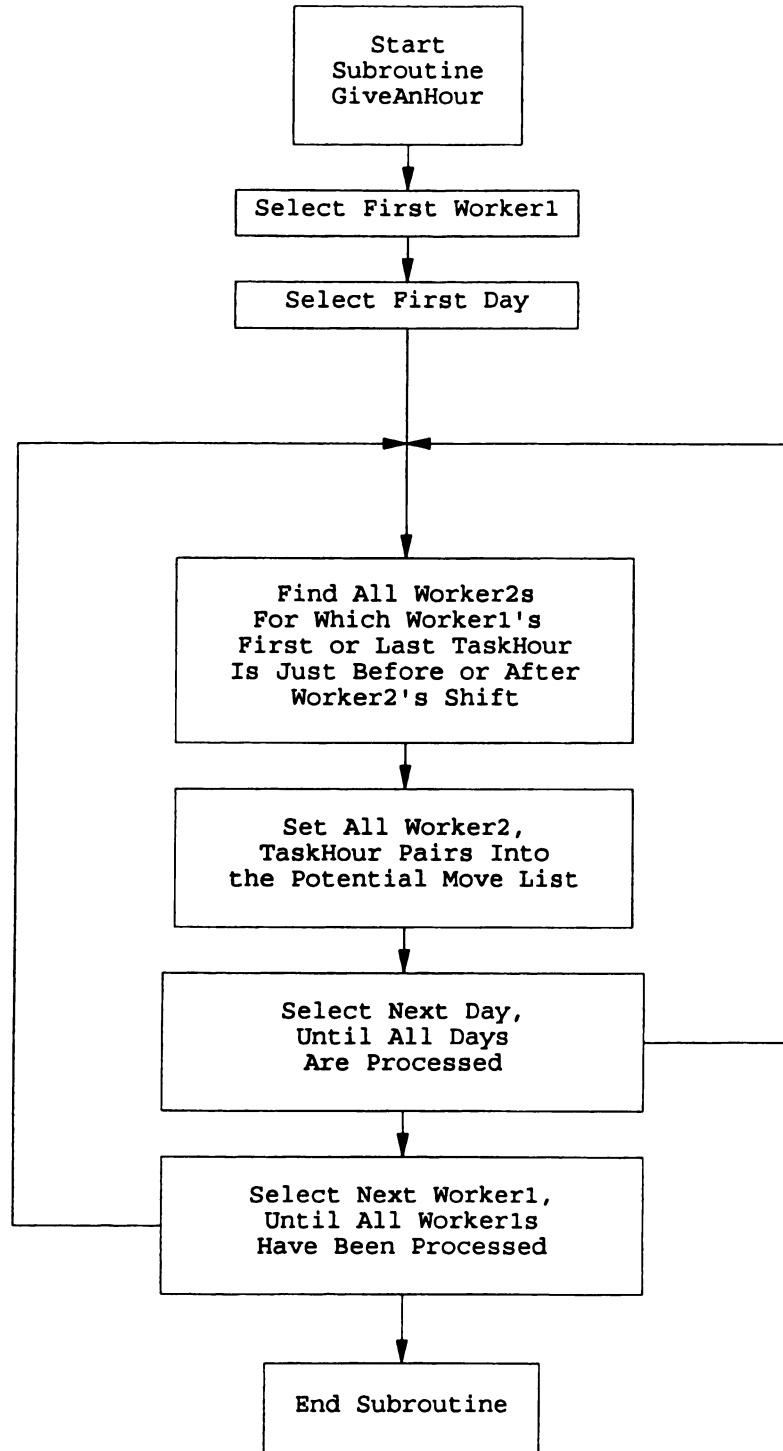


Figure 5-6, Give An Hour to Another Worker

Give A Day To Another Worker. The subroutine that generates these potential moves gives away one workers (worker_1) shift to another (worker_2). The flow chart for this subroutine is given in figure 5-7. First, the shifts for each worker_1 are considered one-by-one and for each shift, a worker_2 is sought for the entire shift. A candidate for worker_2 will be considered for the potential move only if worker_2 is available and qualified for the taskhour. All workers and taskhours so identified are added to the potential move list for further processing.

This type of move is meant to simultaneously decrease two worker's deviation from targeted hours. It may also decrease the cost of the schedule, if the two workers have different wage rates.

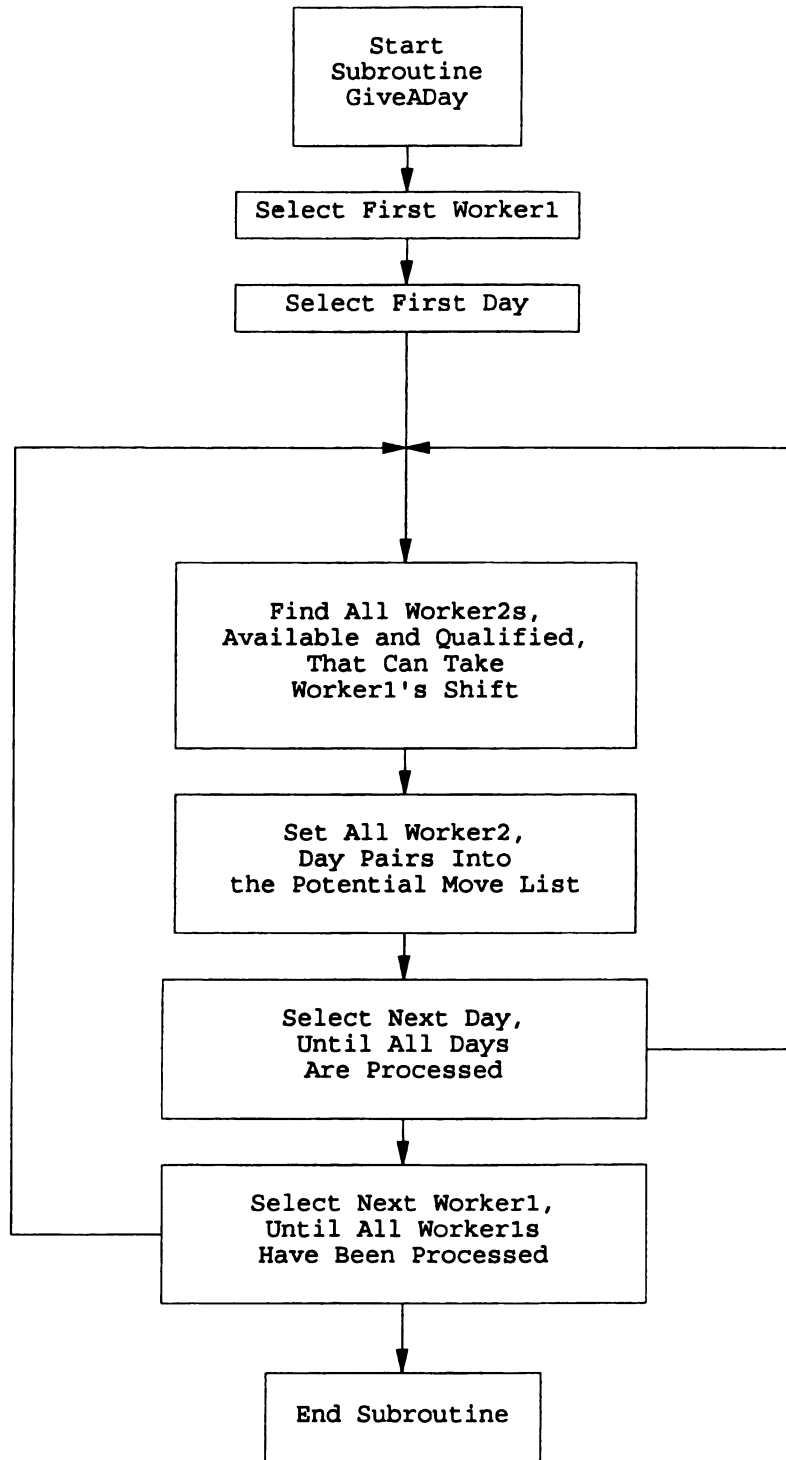


Figure 5-7, Give A Day to Another Worker

Trade Shifts With Another Worker. In this subroutine, potential moves are generated by trading two worker's shifts on the same day. A flow chart for this move is given in figure 5-8. For each worker and each shift for that worker, all other workers are tested to see whether they are eligible to trade shifts. Both workers must be scheduled on the same day, available for the hours of the other worker's shift and be qualified to perform the task on the other worker's shift. If all these conditions hold, then a potential move has been identified and is added to the list of potential moves.

Moves of this type are useful for reducing both worker's deviation from targeted hour, and may also reduce the total schedule cost if the wage rates of the two workers are different.

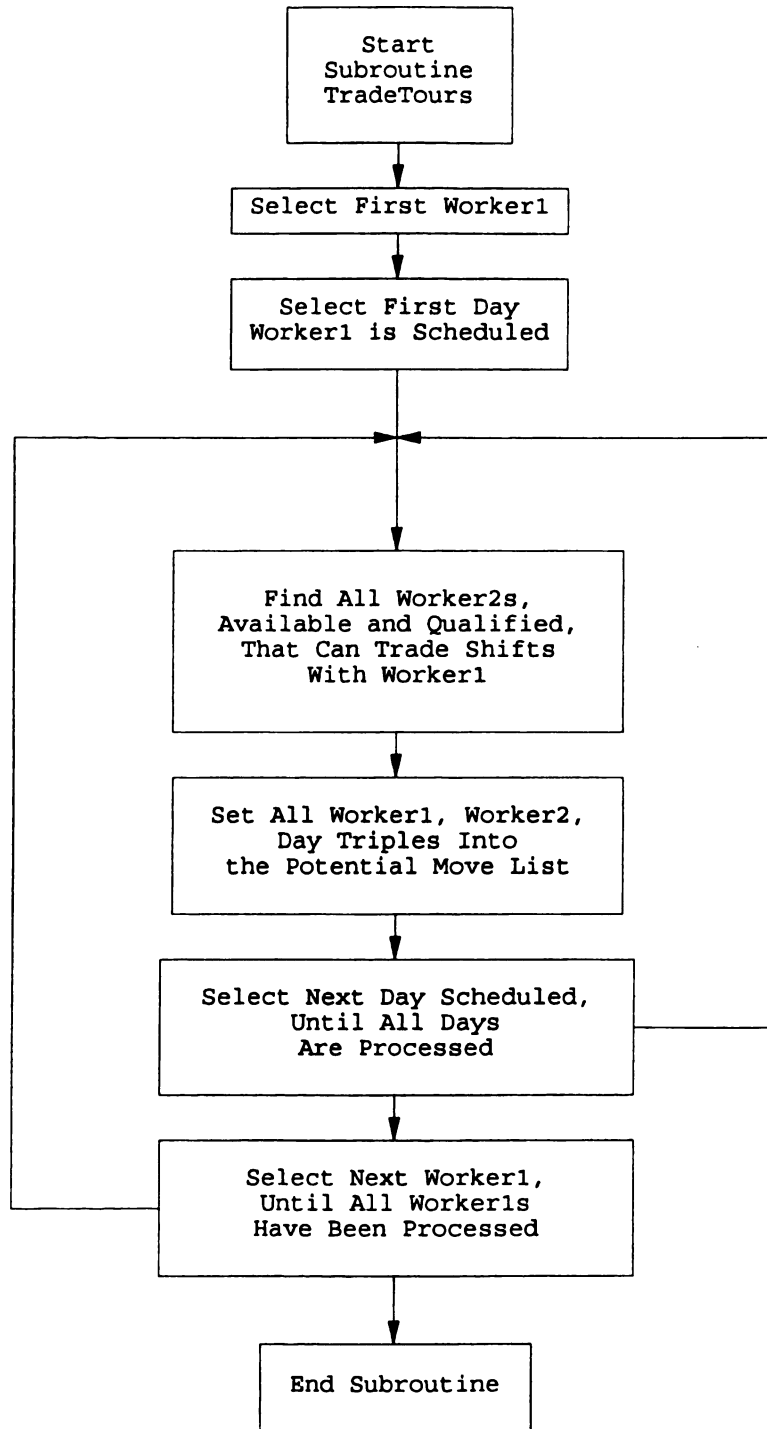


Figure 5-8, Trade Shifts With Another Worker

Cross Trade Shifts With Another Worker. In this subroutine, potential moves are generated by trading two worker's shifts on different days. This move is illustrated with flow chart 5-9. For each worker₁ and each shift for that worker₁, all other workers (worker₂) are tested to see whether they are eligible to trade shifts. Both worker₁ and worker₂ must be scheduled on the different days, available for the hours of the other worker's shift and be qualified to perform the task on the other worker's shift. If all these conditions hold, then a potential move has been identified and is added to the list of potential moves.

Moves of this type are useful for reducing both worker's deviation from targeted hour, and may also reduce the total schedule cost if the wage rates of the two workers are different.

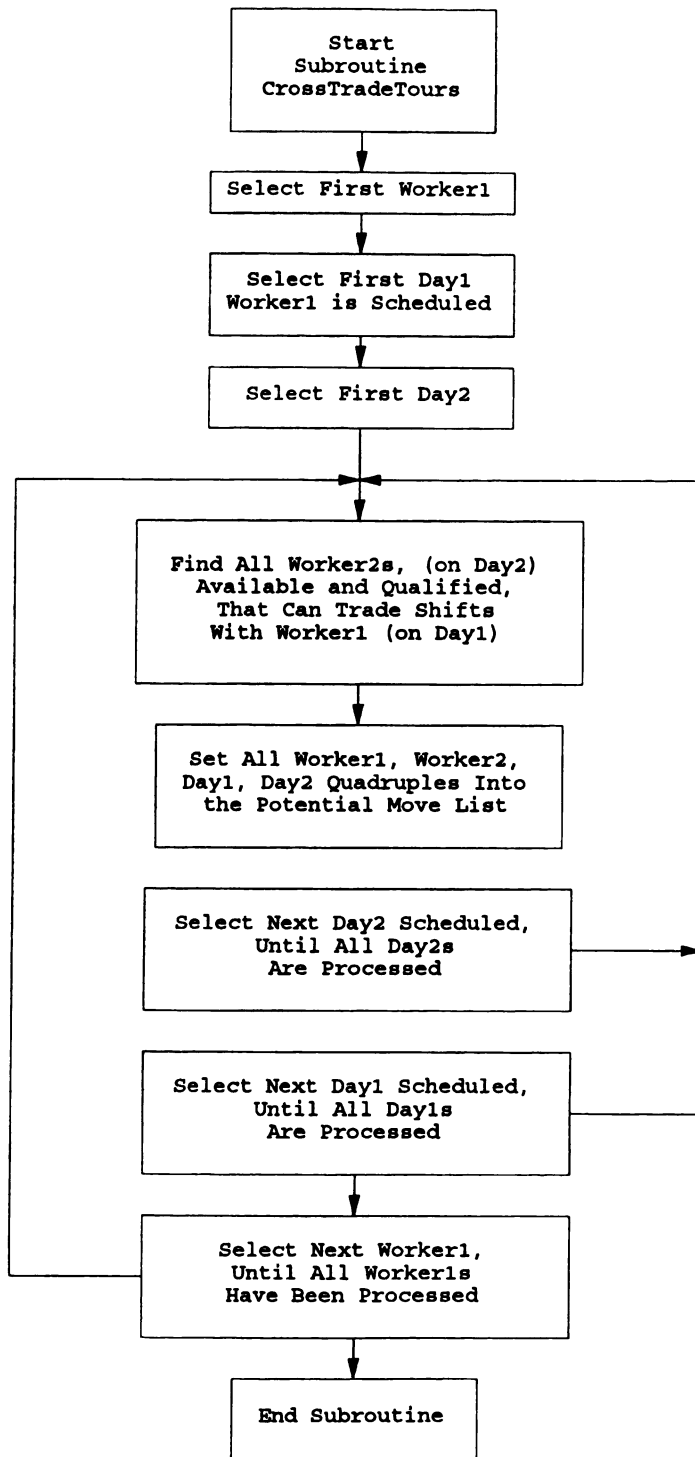


Figure 5-9, Cross Trade Shifts With Another Worker

Give Away Part Of A Worker's Shift. This subroutine gives away part of a worker's ($worker_1$) shift to other workers ($worker_2$) on an hour-by-hour basis. This move is depicted in figure 5-10. First, the shifts for each $worker_1$ are considered one-by-one and for each shift that is longer than the minimum shift, $worker_2$ s are sought for the first and last hours of the shift. The number of hours to be given away is either the worker's shift length less the minimum shift or the number of hours off target for the worker, whichever is less. A candidate for $worker_2$ will be considered only if the taskhour to be given away is either the hour immediately prior to $worker_2$'s existing shift, or the hour immediately following $worker_2$'s existing shift, and if $worker_2$'s is available and qualified for the taskhour. Once $worker_2$'s have been identified for all taskhours to be given away, all possible combinations of those $worker_2$'s and taskhours are generated for the appropriate end of the shift of $worker_1$. All $worker_2$ s and taskhours combinations so identified are added to the potential move list for further processing.

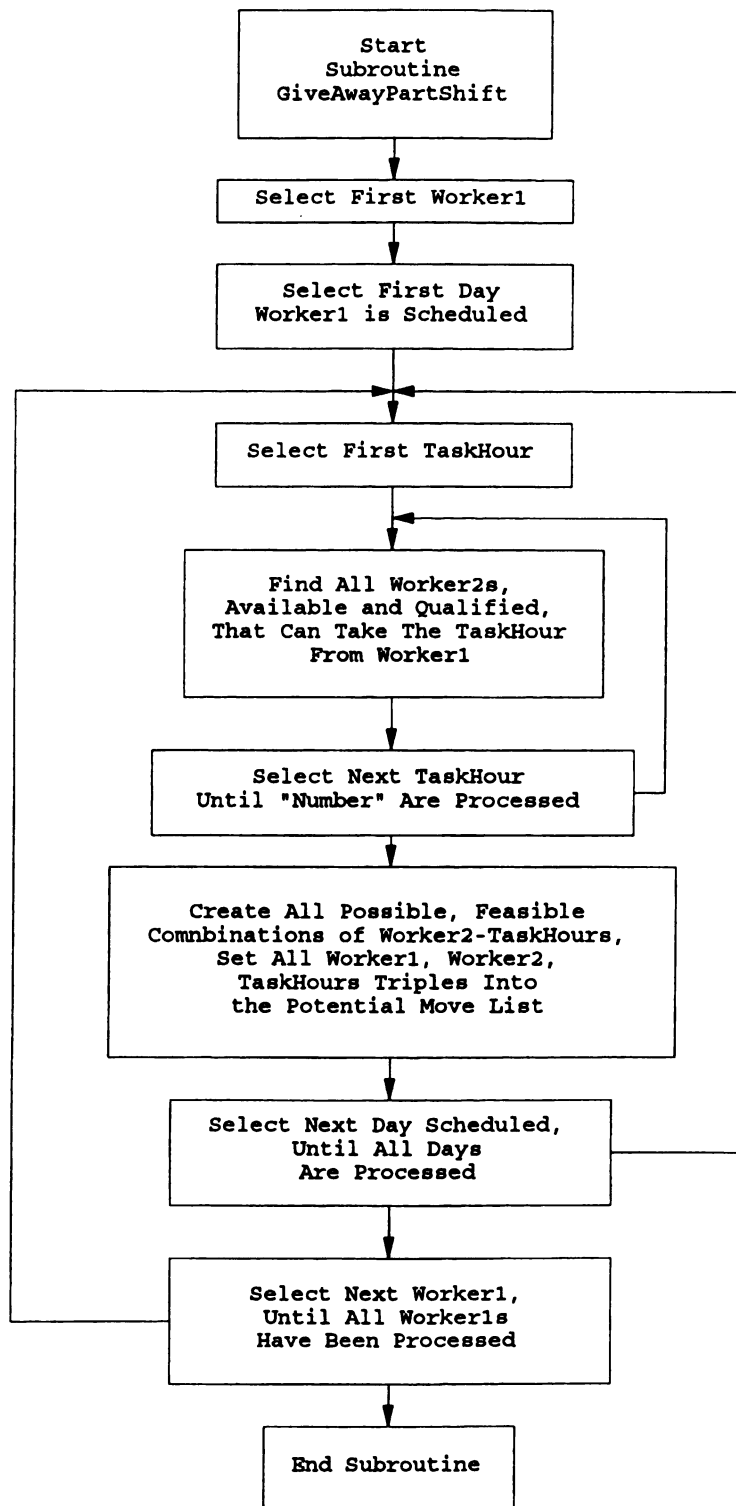


Figure 5-10, Give Away Part of a Worker's Shift

Give Away A Short Shift, Hour By Hour. This subroutine gives away a worker's (worker₁) short shift to other workers (worker₂) on an hour-by-hour basis. A flow chart depiction of this move is given in figure 5-11. First, the shifts for each worker₁ are considered one-by-one and for each shift that is equal to or less than the minimum shift plus one hour, worker₂s are sought for the hours of the shift. The worker₂ will be considered only if the taskhour to be given away is either the hour immediately prior to worker₂'s existing shift, or the hour immediately following worker₂'s existing shift, and if worker₂'s is available and qualified for the taskhour. Once worker₂'s have been identified for all taskhours to be given away, all possible combinations of those worker₂'s and taskhours are generated for the appropriate end of the shift of worker₁. All worker₂s and taskhours combinations so identified are added to the potential move list for further processing.

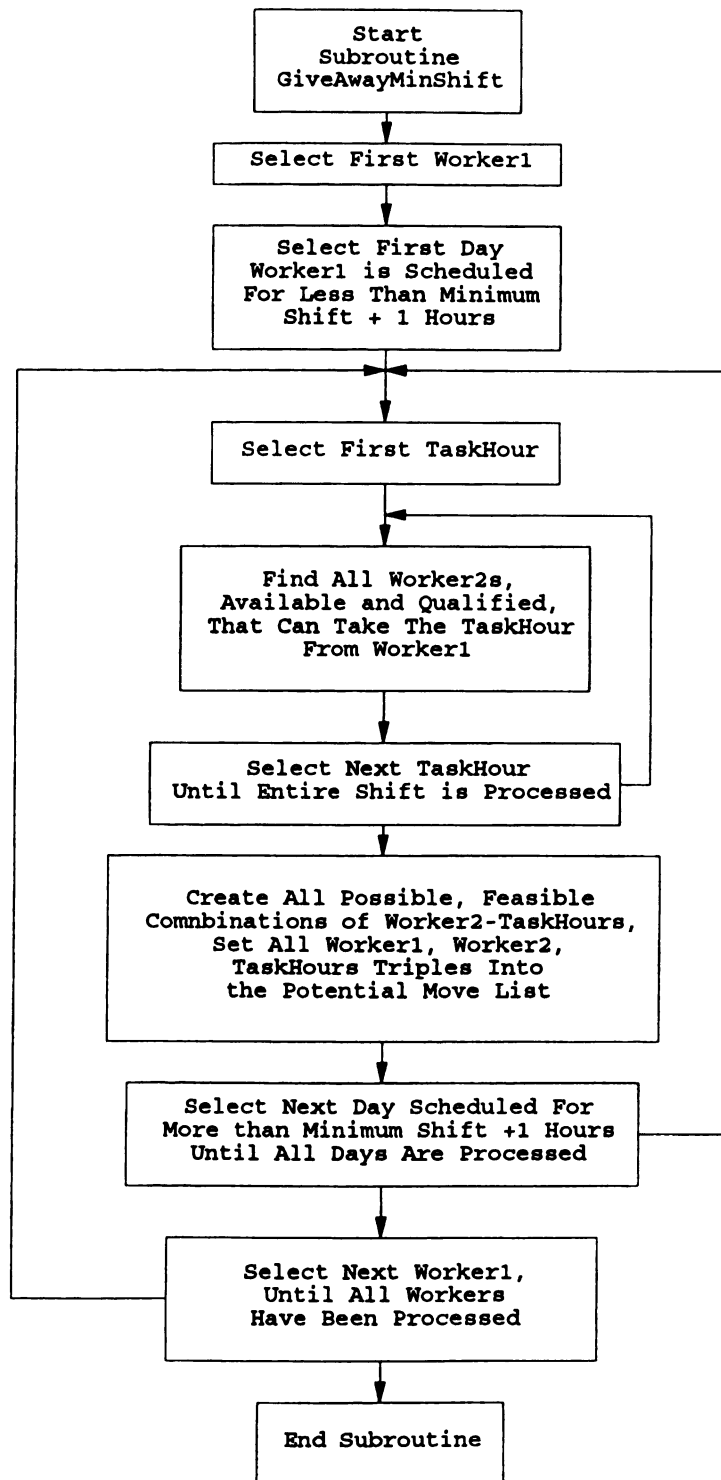


Figure 5-11, Give Away A Short Shift, Hour By Hour

5.2.8. *Formulation of constraints.*

Formulation of the problem for solution with tabu search is shown in appendix 2. Although a complete exposition of the single function for the objective function and constraints of the problem is given in appendix 2, the implementation of tabu search utilized in this research does not require the problem to be in single function form. In fact, the because the constraints are not violated when a potential move is implemented, they are not required to be checked or calculated with the objective function. The routine that calculates the objective function does, however, check for constraint violations every time the objective function is calculated.

5.2.9. *Objective Function Estimator*

A routine to estimate the change in objective function value was developed and implemented in the tabu search algorithm used in this research. An estimator was used for several reasons. First, calculating the entire objective function is a time consuming procedure. Second, computer memory constraints allowed only one copy of the schedule in memory at a time (calculating the objective function requires a complete schedule), so calculating the objective function rather than estimating would cause a great deal of disk access, also very time consuming. Last, the change in objective function can be estimated very accurately. The estimator that was developed determines the changes to be made to the schedule, given the move type and information about the workers, and calculates the changes to the objective function value based on differences in targeted hours, before and after the move, and differences in overstaffing, before and after the move. These

estimates are then used to rank the potential moves, best first. The estimator also allows the potential moves to be evaluated without any changes to the master schedule.

5.3. CONCLUSION

This chapter has described the implementation of tabu search used in this investigation. The overall flow of the algorithm was shown and explained as were the various move generators. The methods used to set parameters was explained and results of those experiments were shown. A demonstration of the fact that the algorithm can cycle with incorrect parameter settings for the tabu list length was demonstrated. Contributions based on the implementation of tabu search include new types of tabu lists, ranking the potential moves, and estimation of objective function value.

CHAPTER 6 - SIMULATED ANNEALING; THEORY AND IMPLEMENTATION

6.1. INTRODUCTION

Simulated annealing is a stochastic optimization technique derived from statistical mechanics. It is used for finding (near) globally minimum cost solutions to wide variety of large, combinatorial optimization problems. Kirkpatrick et al [1983] were the first to propose and demonstrate the application of simulation techniques from statistical mechanics to problems of combinatorial optimization. For a complete discussion of simulated annealing the reader is referred to van Laarhoven and Aarts, [1987].

6.1.1. Statistical Mechanics

An understanding of statistical mechanics is necessary to comprehend the relationship between the techniques of statistical physics and the solution of large combinatorial optimization problems. Statistical mechanics is a body of methods for analyzing aggregate properties of the large numbers of atoms to be found in samples of liquids or solids, in thermal equilibrium at finite temperatures. Because the number of atoms is on the order of 10^{23} per cubic centimeter, only the most probable behavior of

the system in thermal equilibrium is observed. This behavior can be characterized by the average and small fluctuations about the average behavior of the system, when the average is taken over the set of such systems defined by the configurations of the systems. Suppose that the configuration (config.i) of the system is analogous to the set of spatial positions of the components (atoms or molecules). If the system is in thermal equilibrium at a given temperature c , then the probability $p(\text{config.i})$ that the system is in a given configuration (config.i) depends on the energy $E(\text{config.i})$ of the configuration. This probability follows the Boltzmann distribution: $p(\text{config.i}) = \exp(-E(\text{config.i})/k_B c)$, where $E(\text{config.i})$ is the energy of the configuration, c is the temperature, and k_B is Boltzmann's constant ($k_B = 1.38 \times 10^{-23}$).

6.1.2. The Simulation of Particles at Thermal Equilibrium

The behavior of a system of particles or components in thermal equilibrium can be simulated using a stochastic relaxation technique developed by Metropolis et al [1953]. Suppose that at time t , the system is in configuration (config.a). A candidate (config.b) for the configuration at time $t+1$ is generated using a random process. The test for selecting or rejecting configuration (config.b) as the configuration at time $t+1$ is based on the difference between the energy levels of configurations (config.a) and (config.b). If the energy level of (config.b) $<$ (config.a) then (config.b) is accepted as the new configuration. If the energy of (config.b) $\geq$ (config.a) then (config.b) is accepted with probability $p = \exp(-(E(\text{config.b}) - E(\text{config.a}))/k_B c)$. It must be noted that higher energy states can be attained (with probability p), but that the general trend of configuration energy levels will be toward the lower energy states. This trend is also confirmed by noting that the temperature, c , is decreasing periodically (as described in

the definition of annealing, below). This decreases the probability that a configuration with a higher energy level would be accepted.

6.1.3. The Relationship to Combinatorial Optimization

A primary question in statistical mechanics is the nature of the system at low temperatures, for example, whether the atoms remain fluid or solidify, and if they solidify, whether they form a crystalline solid or a glass. Very low energy states predominate at low temperatures, because of the nature of the Boltzmann distribution. To achieve low-energy configurations, simply lowering the temperature is not sufficient, because of the possibility of the production of glasses (localized higher energy states). To avoid these glasses or higher energy states, an annealing process must be used. In an annealing process the temperature is raised, and then slowly lowered, spending enough time at each intermediate temperature level to achieve thermal equilibrium.

The problem of finding the low-temperature state of a system when a formula for its energy level calculation is given is similar to the optimization of a combinatorial problem. This similarity comes from the fact that annealing attempts to transform the entire solid into its lowest energy state, avoiding the formation of glasses in any part of the solid. Such a solid is, of course, composed of very many molecules, that individually, must be cooled evenly to avoid the formation of glasses.

For the analogy combinatorial optimization, the variables may be thought of as molecules. When we simulate annealing we are attempting to reduce the objective function value (energy level) by manipulating the variables (molecules and temperature). The objective function value must be lowered evenly and slowly to avoid local optima (glasses) as described above.

6.1.4. Requirements for Simulated Annealing Optimization

Simulated annealing as applied to optimization problems involves several steps:

- Identification of the analogs of the physical process in the optimization process itself:
 - the energy function becomes the objective function,
 - the configurations of atoms becomes the configuration of parameters,
 - finding a low energy configuration becomes finding a near-optimal solution
 - temperature becomes a control parameter for the simulation.
- Formation of an annealing schedule comprised of the following parts:
 - a set of steadily decreasing temperatures or control parameters designated c in the description of the algorithm. The c parameters control the probability of a higher temperature configuration being accepted as the new configuration at each iteration.
 - the amount of time (number of iterations) spent at each temperature. The objective is to spend enough time at each value of c to minimize the formation of glasses or locally optimal solutions.
- The development of a method(s) of generating new configurations.

6.1.5. Description of Simulated Annealing Algorithm in Pseudo-Pascal

Procedure Simulated Annealing

Begin

Initialize;

$M = 0;$

repeat

repeat

Perturb(Config.i $\rightarrow$ Config.j , ΔC_{ij})

if $\Delta C_{ij} \leq 0$ **then accept else**

if $\exp(-\Delta C_{ij}/c) > \text{random}[0,1]$ **then accept;**

if accept then Update(configuration j);

until equilibrium is approached sufficiently closely;

$c_{M+1} := f(c_M);$

$M := M+1;$

until stop criterion = true;

end.

Where C_i is the objective function value for configuration i and $\Delta C_{ij} := C_j - C_i$.

6.1.6. Flow Chart of Simulated Annealing

A flow chart for the simulated annealing algorithm is given in figure 6-1.

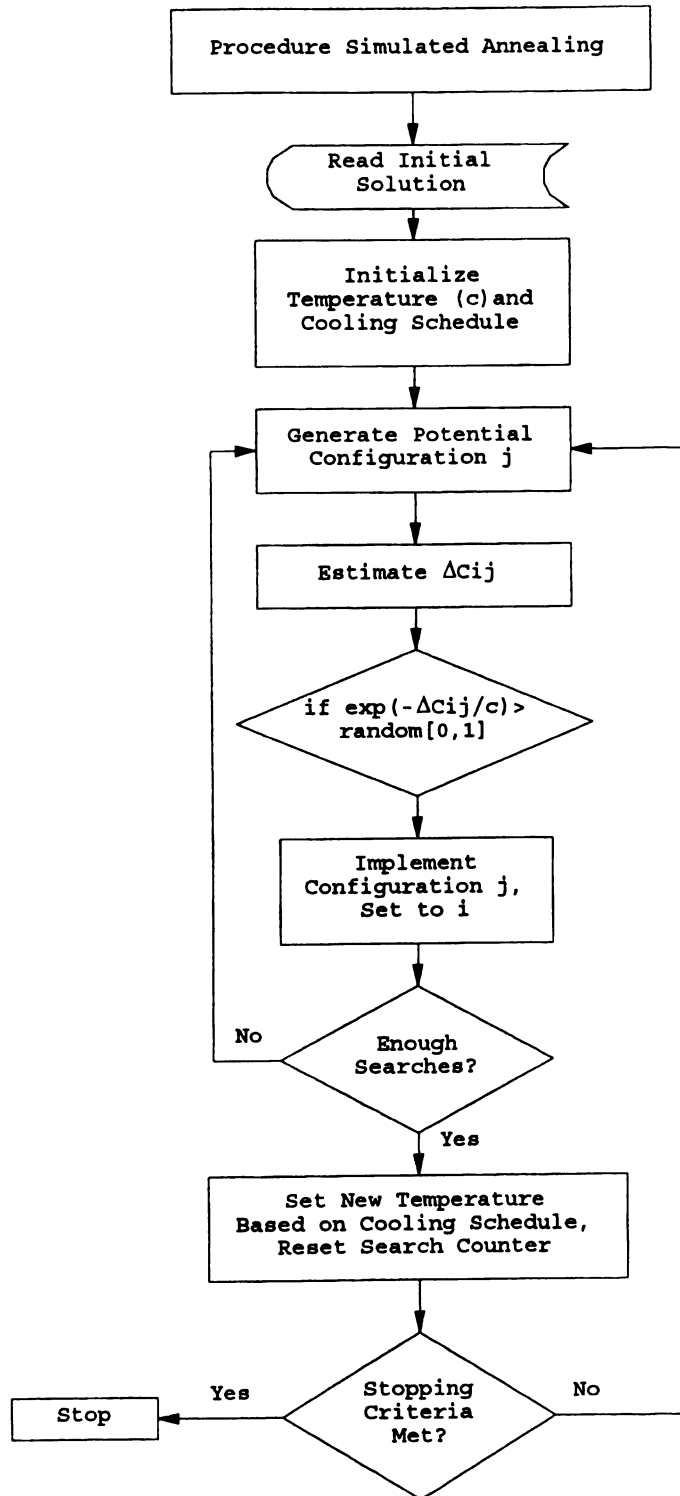


Figure 6-1, Simulated Annealing Flowchart

6.1.7. Example of The Simulated Annealing Algorithm

Consider maximizing the function $f(x) = x^2$, where x is an integer and permitted to vary from 0 to 31. To maximize the function using simulated annealing, we must first code the decision variable as a finite length string. For this problem we will code the variable as an unsigned binary integer string of length 5 ($2^5=32$). We also note that the algorithm is inherently a minimization algorithm. For the purposes of this example we will define ΔC_{ij} to be $C_i - C_j$ (rather than $C_j - C_i$), and we will arbitrarily set c at 2 (c is the control parameter or temperature in the annealing schedule).

Initialize: We select an initial configuration at random, say 00101. The C_i value for this configuration is 5.

Perturb: We select a bit to change at random, in this case the bit at position 3. The C_j value is then 1.

$\Delta C_{ij} = 4$ ($5-1$). This configuration fails the first if-then test. $\exp(-\Delta C_{ij}/c) = 0.135$, a random $U[0,1]$ pick (re-picked at each generation) gives .729, so we do not update and loop back to perturb.

Perturb: We select a bit to change at random, bit 2, giving 00111, with $C_j = 7$.

$\Delta C_{ij} = -1$ so we accept the new configuration.

Following the pseudo-PASCAL description above, the algorithm would continue the inner loop until equilibrium is reached (determined by some stopping rule), when the next c value is used and the process would repeat itself until the algorithm loops with the final value of c . At that point, an optimal or near optimal solution would have been reached.

6.2. *SIMULATED ANNEALING IMPLEMENTATION ISSUES*

6.2.1. *The Annealing Schedule*

The tradeoff in developing a cooling schedule is between solution speed and solution quality. A faster annealing schedule will usually give a poorer solution. Several types of cooling schedules are classified in van Laarhoven and Aarts [1987]. They distinguish between two general classes of cooling schedules:

Class A: A variable number of searches, the number of configurations generated and tried at a given temperature and a fixed decrement of the control parameter, or temperature, and

Class B: A fixed number of searches and a variable decrement of the control parameter.

Both class A and class B cooling schedules are theoretically sound, although most prior research has used class A cooling schedules with good success. Prior research on scheduling problems (Vakharia and Chang [1990] and Brusco and Jacobs [1993]), has also shown good results with class A cooling schedules. Given the prior research and some preliminary experimentation, a class A cooling schedule was selected and implemented for this research. The method for setting the number of searches per value of the control parameter (iteration) was determined in accordance with theoretical concerns. Those concerns are: 1) the algorithm must have a finite probability of accepting all possible configurations, and 2) the probability of accepting a configuration decreases as the value of the control parameter decreases. One method for increasing the likelihood of acceptance for new configurations is to increase the number of configurations tried as the control parameter decreases. A rather elaborate method due to Romeo [1985] was tried first and then compared to a simple method that multiplies the

initial number of searches by the iteration number. The comparison of final objective function values indicated that the simpler method was preferred. The simpler method was used in the remainder of this investigation, to multiply the initial number of searches by the iteration number. After selecting the type of cooling schedule, experimentation was performed to determine good cooling schedule parameters, as explained in the next section.

6.2.2. *Setting the parameter values*

The cooling schedule used for this implementation of simulated annealing has three parameters that require setting; the initial value of the control parameter, the decrement of the control parameter, and the initial number of searches per iteration.

The initial value of the control parameter, c_0 , was set using the method of White [1984]. In this method, a number of configurations are generated and the mean and standard deviation of those configurations are calculated. The mean was calculated to be 1.38 and the standard deviation, 11.94. White proposes that $c_0 \geq \sigma$, so c_0 was set at 12. Setting the parameter in this manner allows most configurations to be accepted early in the execution of the algorithm, which, theoretically, improves the performance of the algorithm.

The rule for the decrement of the control parameter was first proposed by Kirkpatrick [1982], and used in most of the applications published since then. This rule is: $c_{k+1} = \alpha \cdot c_k$, $k = 1, 2, \dots, n$. α is usually set in the range of .5 to .99. After some experimentation it was found that $\alpha = .8$ produced satisfactory results. A more elaborate method for decrementing the control parameter, c , due to Huang, et al [1986] was tried, but the results were not as good as the simpler decrement rule.

The initial number of searches per iteration was set using experimentation. Several values for the initial number of searches were tried ranging from 75 to 300. Analysis of the final objective function values from the range of values indicated that there was no significant difference among the values. The implication is that this implementation of simulated annealing is insensitive to the setting of the initial number of searches per iteration. The initial number of searches per iteration was set at 75 for the remainder of the research.

6.2.3. Stopping Rules

As is shown in the general description of simulated annealing given above, stopping rules are necessary for termination of the algorithm. Two types of stopping rules were implemented in this research. The first rule is based on a final value of the control parameter, c . The final value of the control parameter can be set in a number of different ways. The method chosen for this research is due to Lundy and Mees [1986]. The calculation yields a final value close to .05, so that was the value used for the final value of the control parameter. The other stopping rule used was to stop execution of the algorithm if there had been no improvement in the objective function value after some number of iterations. The rule developed after some experimentation was to stop the algorithm when there had been no improvement for two iterations if at least 10 iterations had been done (this would correspond to at least 1575 configurations tried with no improvement).

6.2.4. *Perturbation Using Configuration Generators*

There are several methods for generating new configurations explained in van Laarhoven and Aarts [1987]. These methods were deemed unsuitable because there was no protection from the generation of constraint violations along with the new configuration. In order to ensure that each new configuration was feasible, and to keep the comparison between simulated annealing and tabu search as fair as possible, the same move generating methods used in tabu search were used in generating perturbations, or new configurations (or moves), for simulated annealing.

The configuration generators used in the implementation of simulated annealing use the same concepts as those used in tabu search. The primary difference is that the move generators for tabu search generate all possible moves of the particular type each time the generator is used, while the move generators used in simulated annealing generate only one move of the particular type each time the generator is used.

Descriptions of each of the move generators follows.

Reduce Over Staffing. The subroutine used to produce a transition to reduce overstaffing is shown in figure 6-2. This subroutine finds the task hour that is most over staffed. Next, a pool of workers scheduled to work during that task hour and whose first or last hour in the shift is the hour most over staffed are identified. From this pool of workers, the worker most over targeted hours is selected. If that worker is not scheduled in the over staffed task for that hour, an attempt to swap the workers task for the over staffed one is made. If the swap is not successful, then the next most over target worker is selected. When a worker from the pool is found that is scheduled for the over staffed task hour (whether the worker swapped for the task or not) that worker, and task hour is returned to be tested for acceptance by the main program.

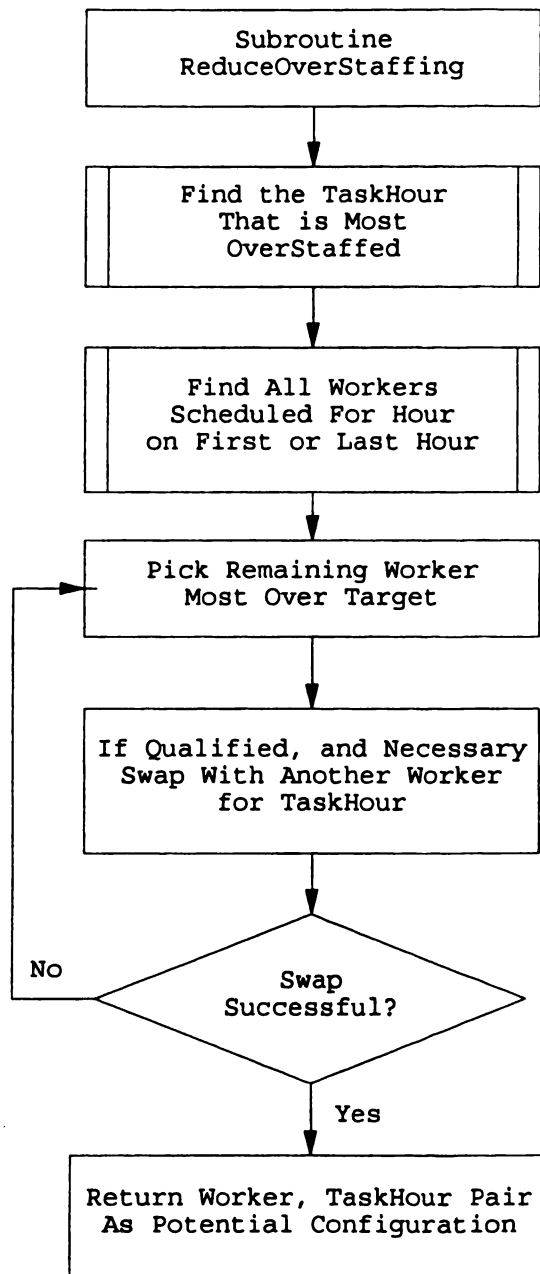


Figure 6-2, Reduce Over Scheduling

Increase Over Staffing. The flow chart for the transition generator is given in figure 6-3. The first step is to identify all workers under targeted hours, sort them according to the number of hours under target and select the worker most under target. Next that worker's shortest shift is found. Once the shortest shift is known an hour either just before the shift or just after the shift is picked randomly. The task is set to be the same as for assignment of the hour immediately before or after the hour picked, as the case may be. This task hour and worker combination is returned to the main program for acceptance or rejection according to the metropolis equation.

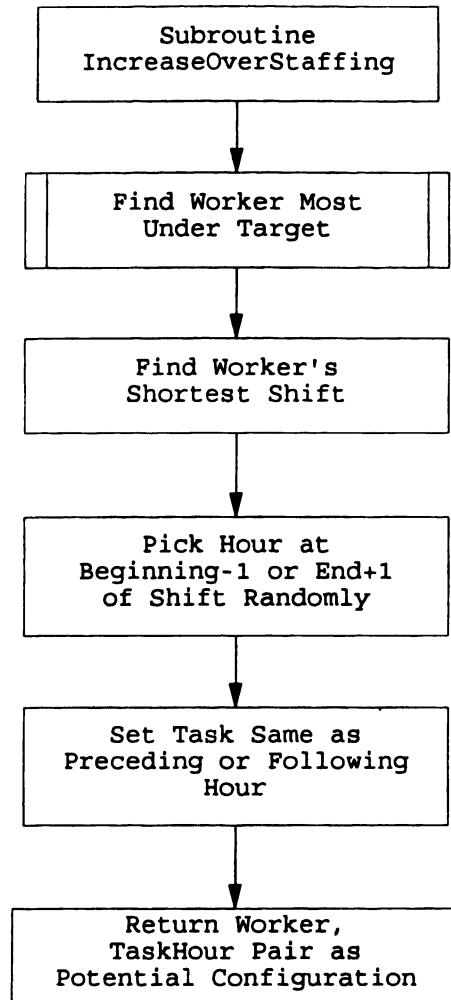


Figure 6-3, Increase Over Scheduling

Give An Hour To Another Worker Randomly. Figure 6-4 gives the flowchart of the subroutine that creates a potential configuration that gives an hour to another worker randomly. The first step is to pick a day in the scheduling period randomly, from a uniform distribution. Next the pool of all workers scheduled for the day is created. From the pool, a worker is picked randomly, and the rest of the pool is searched for all other workers that can take for the first worker's first or last hour, considering availabilities and task qualifications. From the pool of all workers that can trade, a second worker is chosen, based on the shortest shift length. Based on the availability of the second worker, the task hour is determined, and the first and second worker and the task hour are returned to the main program for further processing.

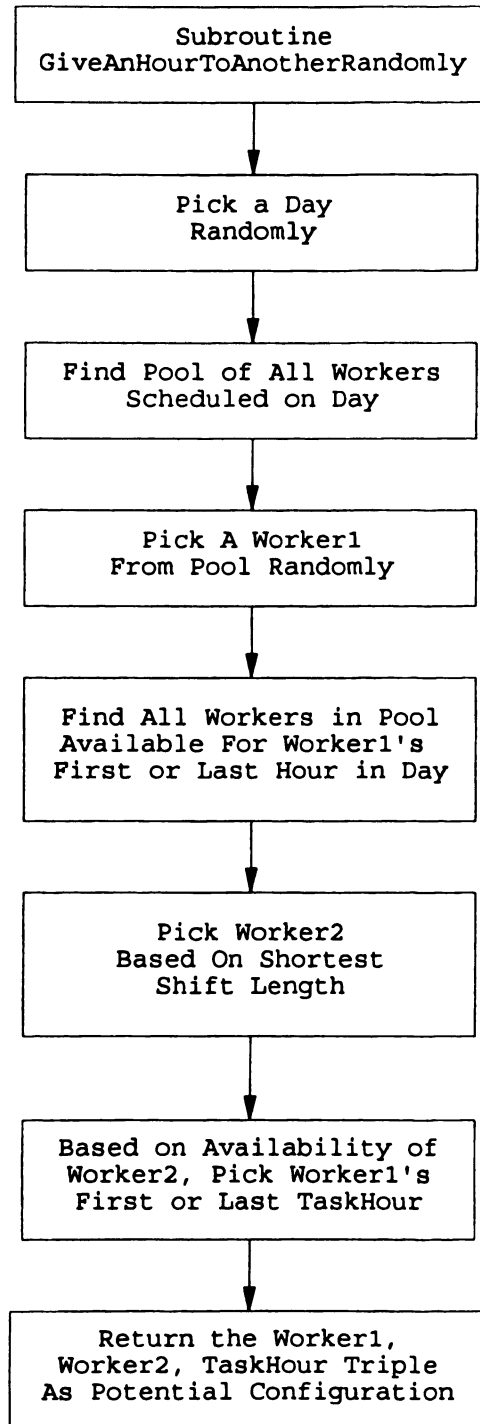


Figure 6-4, Give An Hour to Another Randomly

Give A Day To Another Worker Randomly. The flowchart of the subroutine that creates a potential configuration that gives a day to another worker randomly is given in Figure 6-5. The first step is to pick a day in the scheduling period randomly, from a uniform distribution. Next the pool of all workers scheduled for the day is created. From the pool, a worker is picked randomly. A second pool of worker is formed that can trade for the first worker's shift , considering availabilities and task qualifications. From the pool of all workers that can trade, a second worker is chosen, based on the most hours under target. Finally, the first and second worker and the shift are returned to the main program for further processing.

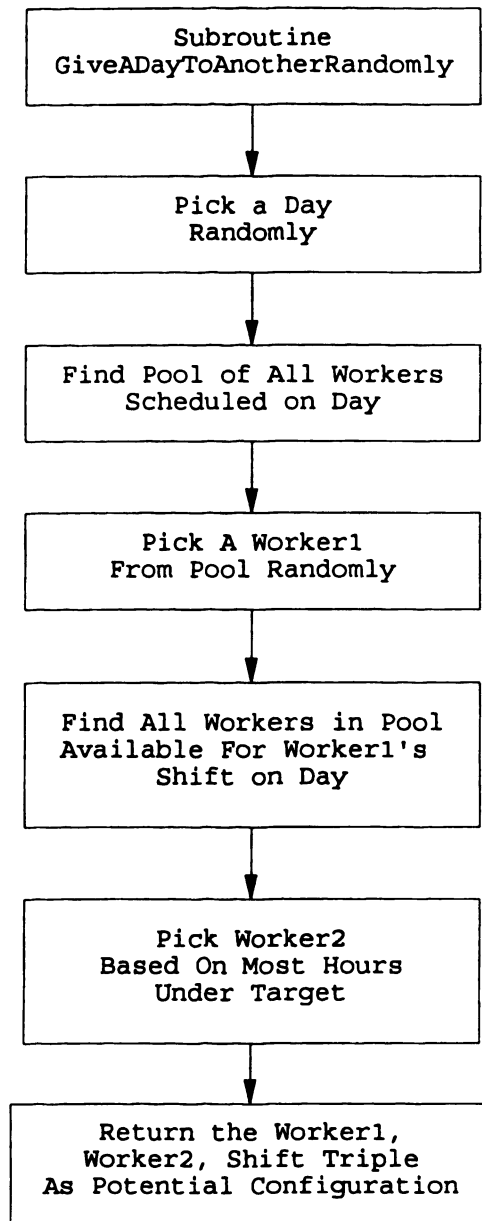


Figure 6-5, Give A Day To Another Randomly

Trade Tours Between Workers Randomly. The subroutine that generates a new configuration that trades tours between two workers randomly is given in figure 6-6. The first step is to pick a day in the scheduling period randomly, from a uniform distribution. Next the pool of all workers scheduled for the day is created. From the pool, a worker is picked randomly. A second pool of worker is formed that can trade for the first worker's shift (and that the first worker can trade with), considering availabilities and task qualifications. From the pool of all workers that can trade, a second worker is chosen, based on the most hours under target. Finally, the first and second worker and the shift are returned to the main program for further processing.

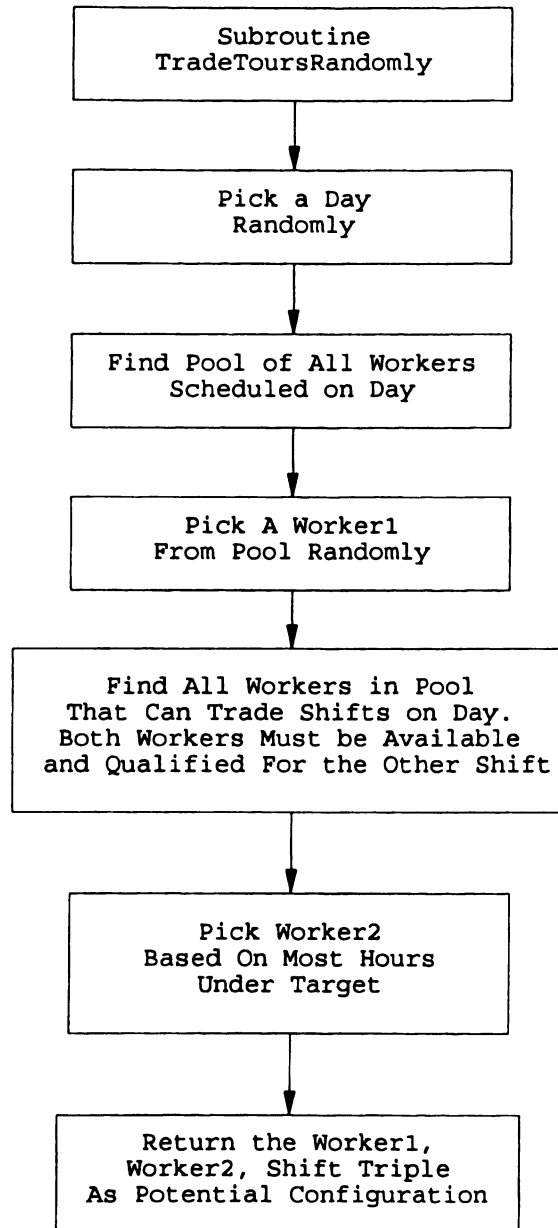


Figure 6-6, Trade Tours Randomly

Cross Trade Tours Between Workers Randomly. Figure 6-7 gives the flowchart of the subroutine that creates a potential configuration that cross trades tours randomly. The first step is to pick a day in the scheduling period randomly, from a uniform distribution. Next the pool of all workers scheduled for that day is created. From the pool, a worker is picked randomly. Another pool of workers is created with workers that are under target and are available and qualified to take the first worker's shift. The worker most under target is picked as the second worker. All the days the second worker is scheduled to work are checked to determine whether the first worker is available and qualified to take the second workers shift on that day. If so, a match has been found and the first worker, the second worker and the two shifts are returned to the main program for further processing. If not, another second worker is chosen and the process repeats.

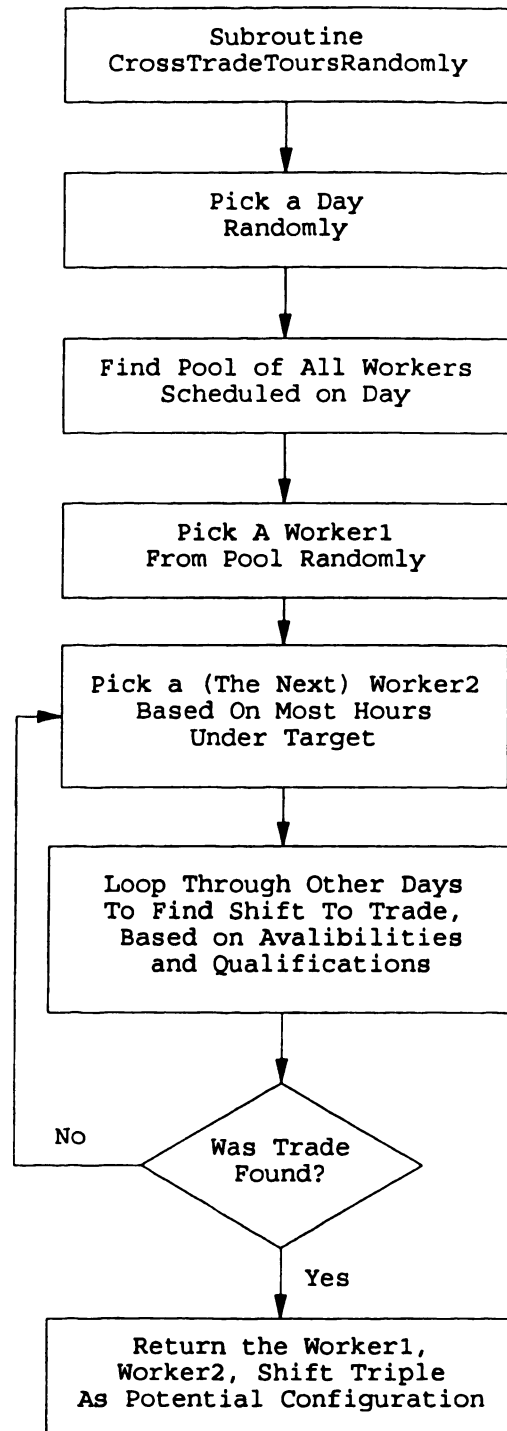


Figure 6-7, Cross Trade Tours Randomly

Give An Hour To Another Worker. The reader may refer to the flowchart shown figure 6-8 for a depiction of the subroutine that generates a configuration based on the method of give an hour to another worker. This subroutine first creates a pool of workers that are over targeted hours. The worker with the most hours over target is picked from the pool. From the worker's tour, the longest shift is selected. A second pool is created of workers that are available and qualified for either the first or last hour of the first worker's longest shift. This second pool's workers are required to have the shift they are scheduled for on the day in question adjacent to the first worker's shift, so the second worker might take either the first or last hour of the first worker's shift, without fill-in hours. Once a second worker is found, the subroutine returns the first and second workers and the task hour to the main program for further processing.

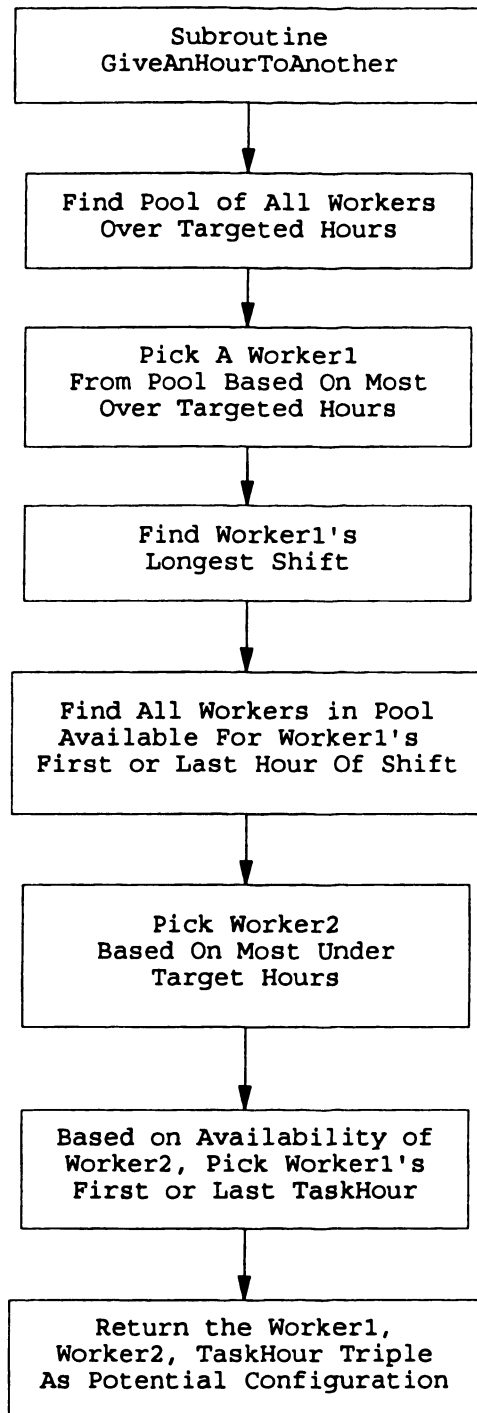


Figure 6-8, Give An Hour To Another

Give A Day To Another Worker. Figure 6-9 illustrates the subroutine that generates a new configuration based on the method of give a day to another. This subroutine first creates a pool of workers that are over targeted hours. The worker with the most hours over target is picked from the pool. Next, the worker's tour is searched for the shift whose length most closely matches the number of hours over target. Then a pool of workers, available and qualified for the first worker's shift, is created. A worker with the most hours under target is picked from this second pool. The first worker, the second worker and the shift are returned to the main program for further processing.

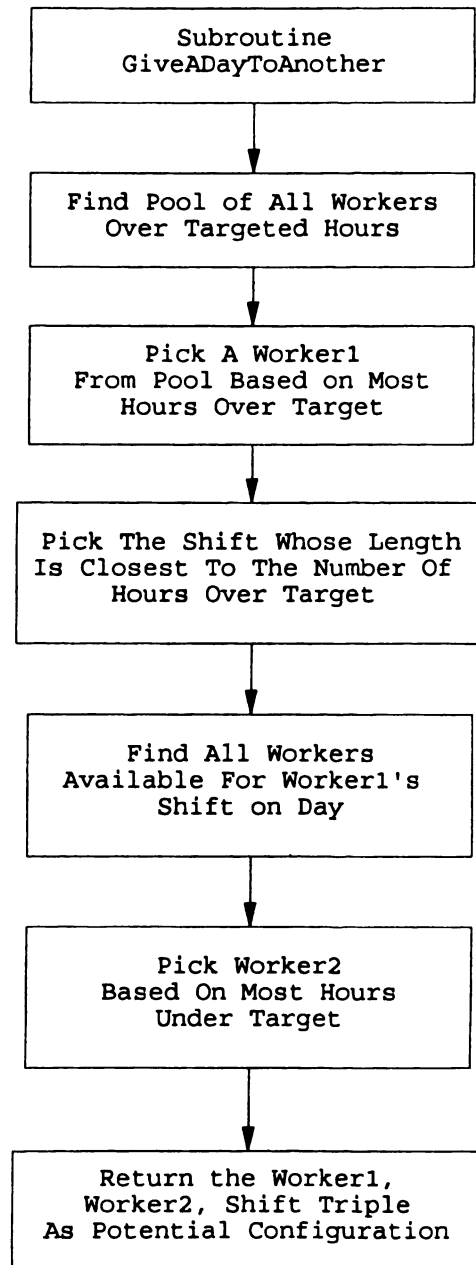


Figure 6-9, Give A Day To Another

Trade Tours Between Workers. Figure 6-10 illustrates the subroutine that generates a new configuration based on the method of trade tours. This subroutine first creates a pool of workers that are over targeted hours. The worker with the most hours over target is picked from the pool. Next, the worker's tour is searched for the shift whose length most closely matches the number of hours over target. Then a pool of workers, available and qualified for the first worker's shift, is created. A worker with the most hours under target is picked from this second pool. The second worker's shift is tested to determine whether the first worker is available and qualified to take it. If so, a match has been found and the first worker, the second worker and the shift are returned to the main program for further processing. If not, another second worker is picked and the process repeats.

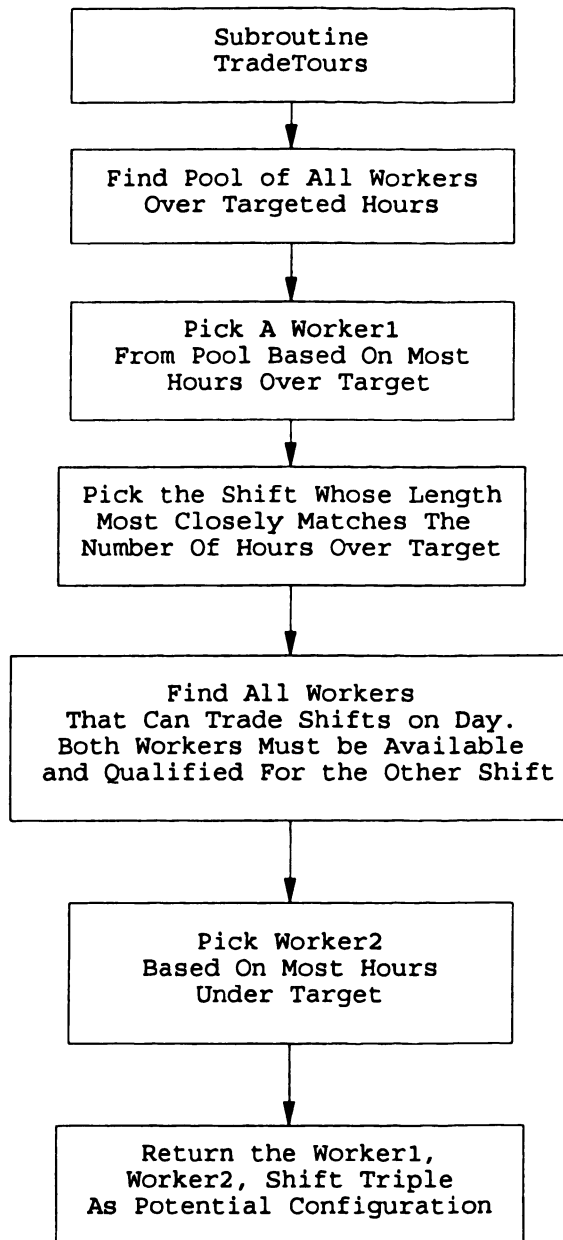


Figure 6-10, Trade Tours Between Workers

Cross Trade Tours Between Workers. The flowchart in figure 6-11 shows the method for generating configurations by cross trading tours. This subroutine first creates a pool of workers that are over targeted hours. The worker with the most hours over target is picked from the pool. Next, the worker's tour is searched for the shift whose length most closely matches the number of hours over target. Then a pool of workers, available and qualified for the first worker's shift, is created. A worker with the most hours under target is picked from this second pool. All the days the second worker is scheduled to work are checked to determine whether the first worker is available and qualified to take the second workers shift on that day. If so, a match has been found and the first worker, the second worker and the two shifts (on different days) are returned to the main program for further processing. If not, another second worker is chosen and the process repeats.

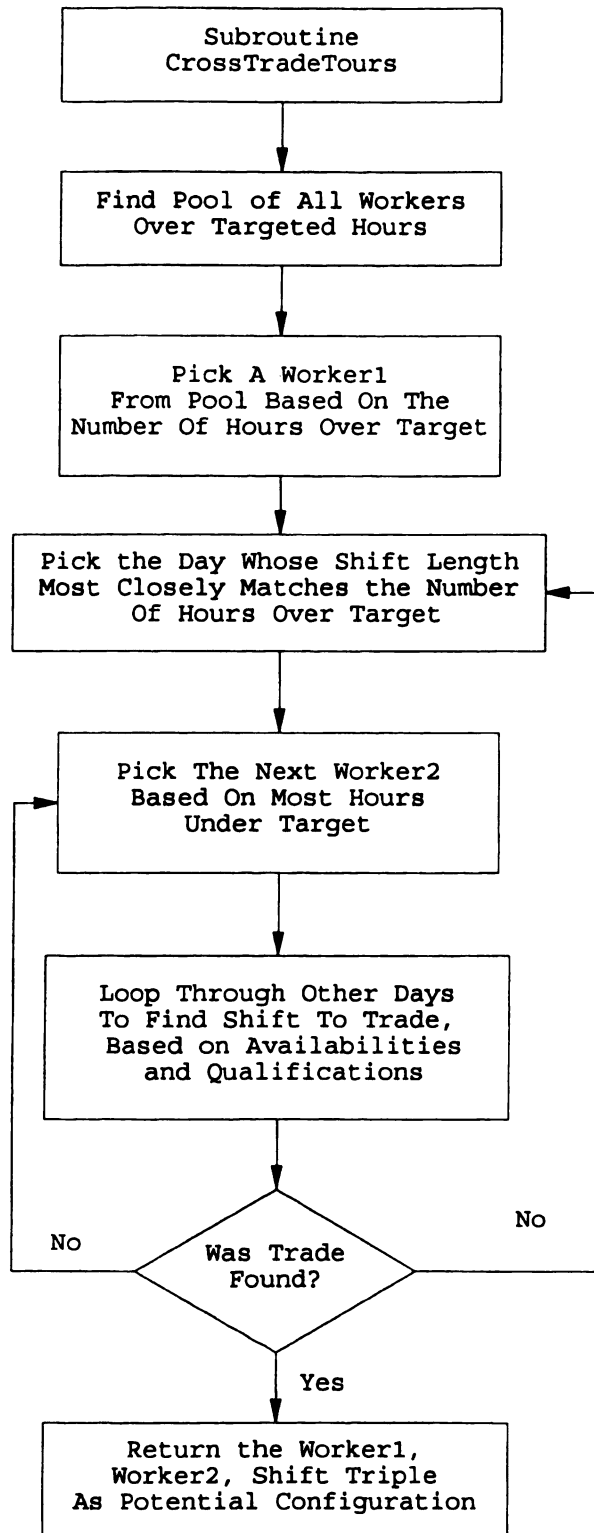


Figure 6-11, Cross Trade Tours Between Workers

Give Away Part Of Shift To Other Workers. The subroutine used to produce a transition to give away part of a worker's shift is shown in figure 6-2. This subroutine first finds the task hour that is most over staffed. Next, a pool of workers scheduled to work during that task hour is created and the worker that is most over targeted hours is chosen as the first worker. The task hour chosen in the first step lies somewhere in the first worker's shift. The portion and length of the shift to be given away are determined by the location of the task hour in the shift and the number of hours the worker is over target. Once the portion of the shift to be given away has been identified, a second worker, available and qualified, and most under targeted hours is found. The first worker, second worker, and portion of the shift are returned to the main program for further processing.

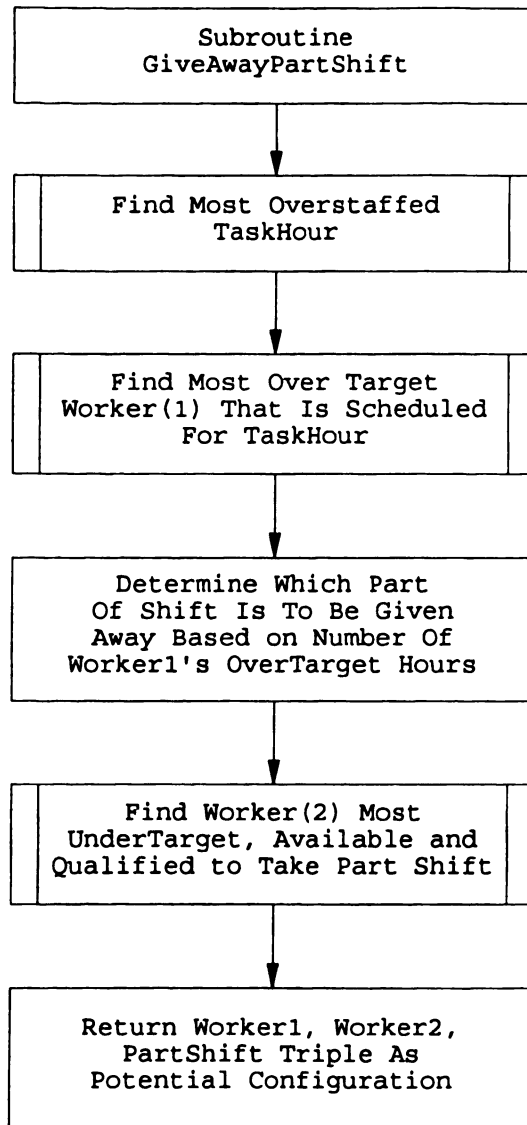


Figure 6-12, Give Away Part Of Shift To Other Workers

Give Away A Short Shift To Other Workers. An illustration of the subroutine to give away a short shift is given in figure 6-13. This subroutine gives away a worker's shortest shift (either minimum hours or minimum hours + 1), to other workers, one hour at a time. First the worker with the most over targeted hours is identified. Next that workers shortest shift is identified. Other workers are identified that can take the hours, one hour at a time. Then lists of the hours, and workers are returned to the main program for further processing.

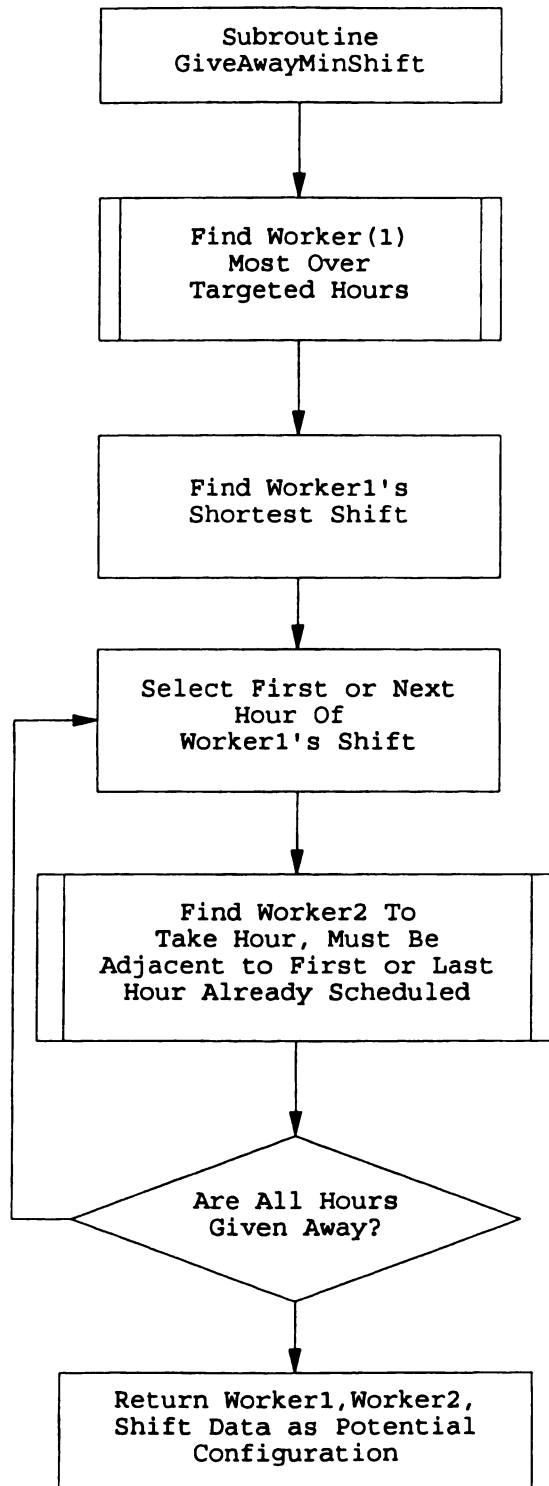


Figure 6-13, Give Away A Short Shift

6.2.5. Inclusion of Both Improvement And Random Configuration Generators

The type of configuration generators described in the simulated annealing literature have been what might be termed random configuration generators. These random configuration generators produce a perturbation in a random manner. Configuration generators of this type have been used for several reasons, the first of which is that annealing, in a physical sense, is a random process driven by the brownian motion of the molecules in the solid. Additionally, the problems to which simulated annealing has been applied lend themselves to random perturbations. As the configurations generators for this problem were being developed, it became evident that both random and "improvement" configuration generators could be developed. An improvement configuration generator produces potential configurations that are chosen to improve the objective function (solution), as apposed to the random configuration generators that may or may not improve the objective function. Theoretically, there was no reason why improvement configuration generators should not be utilized. Practically, it was not known, however, whether improvement configuration generators would be useful. As the reader will have noted from the descriptions of the configuration generators, both random and improvement configuration generators were used in this implementation of simulated annealing. Further, the reader will have noted that several of the configuration generators had both a good and a random version. Statistical justification for use of both types follows.

Separate runs on 15 test problems were made for the simulated annealing algorithm with the set of both random and improvement configuration generators (designated SA in tables 6-1 and 6-2), the set of configuration generators without the random versions (SANR), and the set of configuration generators without the improvement versions (SARO). An ANOVA was calculated as the statistical test for the hypothesis that all sets of configuration generators perform similarly, vs. the alternative

that there are significant differences among the sets of configuration generators. Results of the ANOVA are given in table 6-1.

Table 6-1. ANOVA for Objective Function

ANOVA for Objective Function Value					
VAR\$: SA, SANR, SARO					
DEP VAR: Objective Function Value					
MULTIPLE R: 0.759					
SQUARED MULTIPLE R: 0.577					
ANALYSIS OF VARIANCE					
SOURCE	SUM-OF-SQUARES	DF	MEAN-SQUARE	F-RATIO	P
VAR\$	1922542.178	2	961271.089	28.596	0.000
ERROR	1411841.467	42	33615.273		

Table 6-2, TUKEY HSD Multiple Comparison Test For Objective Function Value

TUKEY HSD Multiple Comparison Test For Objective Function Value			
VAR\$ SA, SANR, SARO			
USING LEAST SQUARES MEANS.			
USING MODEL MSE OF 33615.273 WITH 42. DF.			
MATRIX OF PAIRWISE MEAN DIFFERENCES:			
	SA	SANR	SARO
SA	0.000		
SANR	479.000	0.000	
SARO	381.533	-97.467	0.000
TUKEY HSD MULTIPLE COMPARISONS.			
MATRIX OF PAIRWISE COMPARISON PROBABILITIES:			
	SA	SANR	SARO
SA	1.000		
SANR	0.000	1.000	
SARO	0.000	0.322	1.000

If we take the objective function value as our measure of solution "goodness", we find that there are significant differences between the three sets of configuration generators. So we reject the null hypothesis that there are no significant differences among the three sets of configuration generators. Table 6-2 shows the Tukey HSD multiple comparison test for differences between the means of objective function values for the three different sets of configuration generators. The ANOVA results in table 6-1 show that there are significant differences among means of the three sets of configuration generators at $p < .001$. The Tukey HSD multiple comparison test shown in table 6-2 indicates that there are significant differences between SA and SANR and SARO but not

between SANR and SARO. Further, the mean for SA is smaller than both SANR's and SARO's. The implication is that the combination of both random and improvement configuration generators is better than either alone, at least for this implementation of simulated annealing. Full justification for the use of the Tukey HSD multiple comparison test is given in chapter 8.

6.2.6. Probabilities used for selection of configuration generators

As is mentioned before, there are 12 different configuration generators. With 12 generators, a question naturally arises: "Would the algorithm perform better if some configuration generators were used more frequently than others, or would choice of configuration generator by equal probability random choice be better?". To answer this question some experimentation was done. Several schemes were tried, equal probabilities, adaptive probabilities based on percentage of configurations accepted, unequal probabilities produced randomly, and probabilities based on expectations of what configuration generators would work best at the beginning and end of the run. The results of the experimentation indicated the probabilities based on expectations of what configurations would work best at the beginning of the run and what configuration generators would work best at the end of the run produced better solutions, in terms of objective function values. All configuration generators had positive probabilities at every stage of the optimization procedure. Those configuration generators that were thought to perform better at the beginning of the run had relatively higher probabilities of being used at the beginning of the run, with decreasing probabilities each iteration. Those configuration generators that were thought to perform better at the end of the run had

relatively lower probabilities of being used at the beginning of the run, with increasing probabilities each iteration.

6.2.7. *Improvement Path for Simulated Annealing*

The improvement path of the objective function value is shown in figure 6-14. This path has marked similarities to those shown in other investigations, see Knox [1991], and Brusco and Jacobs [1993].

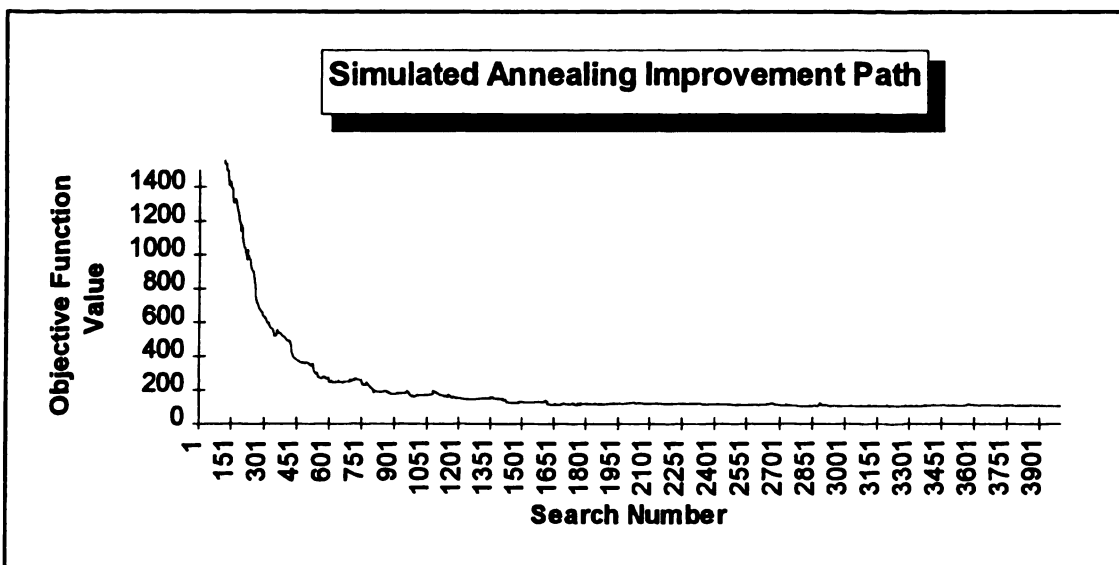


Figure 6-14, Simulated Annealing Improvement Path

6.3. *CONCLUSION*

The theory and implementation details of simulated annealing have been given. Among the contributions of this investigation are the introduction of "improvement"

configuration generators, and the use of weights to determine how often a configuration generator is used at a particular stage of the algorithm.

CHAPTER 7 - RESEARCH METHODOLOGY

7.1. INTRODUCTION

The objectives of this research were to adapt and implement two new algorithms, simulated annealing and tabu search, and to compare the performance of these two algorithms with an existing heuristic, Loucks', for solving large scale labor scheduling problems. The comparison was to be made on three areas of performance; speed to solution (measured in seconds), manpower cost of final schedule (measured in dollars), and quality of solution, measured by the number of hours overstaffed, by the sum of squared deviations from workers targeted hours and by the objective function value (a linear combination of the number of hours overstaffed and the sum of squared deviations from workers targeted hours).

The algorithms were tested across a spectrum of problems varying in difficulty as measured by three factors; Average Worker Utilization, Average Percentage of Number Of Tasks Known, and a measure of peak-to-trough taskhour demand variation, designated Max-Min Ratio.

Average worker utilization is the ratio, averaged across all workers, of the number of hours scheduled to number of hours targeted, expressed as a percentage. This factor is intended to measure the flexibility that a scheduler would have in assigning

workers to particular task hours. As average worker utilization increases, the problem becomes more difficult to solve, because there are fewer workers available to schedule for each task hour.

The measure, average percentage of number of tasks known, is the ratio, averaged across all workers, of the number of tasks known by a worker to the total number of tasks, expressed as a percentage. This factor also is intended to measure the flexibility that a scheduler would have in assigning a worker to a particular taskhour. As the measure, average percentage of tasks known by worker, is increased the scheduler has more flexibility to schedule workers, making the problem easier to solve.

The min-max ratio is the maximum demand in task hours for the day (summed across all tasks) divided by the minimum demand for task hours (summed across all tasks) within a four hour period. This factor was included in order to capture the difficulty a scheduler might encounter in minimizing the total number of task hours scheduled in the presence of peakedness in customer demand for task hours and a constraint on the minimum number of hours scheduled in a shift for a worker (in this case, 3). A higher level of this ratio indicates a more difficult problem.

This investigation was conducted in three phases. First a set of hypotheses were developed, based on the objectives of the research as stated above. Second, an experimental design was chosen to accomplish the objectives of the research. And third, the new methods as well as Loucks' heuristic were used to solve the labor tour scheduling problems associated with the worker flexibility and labor requirements characteristics specified in this research.

7.2. CHOICE OF ALGORITHM FOR COMPARISON

The heuristic solution method of Loucks [1987] was chosen for comparison for several reasons; the method has found optimal solutions in research settings [Loucks 1987], software is available for comparisons between algorithms, and the underlying formulation used in the software is similar enough to the one chosen for the research problem that it can be used for comparative purposes.

7.3. HYPOTHESES

As stated earlier, an objective of the research was to compare the three algorithms, Loucks' heuristic, tabu search, and simulated annealing. To accomplish that objective several hypotheses were developed relating to the speed, cost and quality of solution. In the following discussion of the hypotheses, the subscript attached to the dependent variable of interest refers to the solution method. For example, Time₁ refers to the Time for Loucks' heuristic, Time₂ refers to the time for simulated annealing, and Time₃ refers to the Time for tabu search.

7.3.1. Time - Speed of Algorithm

$$H_0 : \text{Time}_1 = \text{Time}_2 = \text{Time}_3$$

$$H_1 : \text{Time}_1 \neq \text{Time}_2 \neq \text{Time}_3$$

Speculation at the outset of this research was that tabu search and simulated annealing will outperform the Loucks' method on this measure [Knox, 1989]. Further,

related evidence [Knox, 1989, Skorin-Kapov, 1990], shows that tabu search outperforms simulated annealing in solution time for small problems. However, since the problems to be used in this research are large, the a priori expectation was that simulated annealing will outperform tabu search in Total CPU Time. Accordingly, the alternative hypothesis states that solution speeds are unequal.

7.3.2. *Cost - Total Manpower Cost of the Schedule.*

$$H_0 : \text{Cost}_1 = \text{Cost}_2 = \text{Cost}_3$$

$$H_1 : \text{Cost}_1 \neq \text{Cost}_2 \neq \text{Cost}_3$$

There was no evidence at the outset of this research to suggest that there is a cost difference between schedules developed by the different algorithms.

7.3.3. *Quality Of Solution*

TOS - Total man-hours of overstaffing.

$$H_0 : \text{TOS}_1 = \text{TOS}_2 = \text{TOS}_3$$

$$H_1 : \text{TOS}_1 \neq \text{TOS}_2 \neq \text{TOS}_3$$

There is no evidence to suggest any a priori ranking on this measure. This does not diminish its importance, however, because as overstaffing increases so do costs. An algorithm with a higher TOS will produce higher cost schedules.

7.3.4. TSD - Sum of squared deviations between scheduled and targeted work hours.

$$H_0 : TSD_1 = TSD_2 = TSD_3$$

$$H_1 : TSD_1 \neq TSD_2 \neq TSD_3$$

The measure TSD is an important indicator of a worker's perception of schedule quality. There is no evidence to suggest any rankings, however. Since simulated annealing can be shown to converge to an optimal solution, given sufficient time [van Laarhoven and Aarts, 1987], it may outperform the other algorithms on this performance measure.

7.3.5. OBJ - Objective Function Value

$$H_0 : OBJ_1 = OBJ_2 = OBJ_3$$

$$H_1 : OBJ_1 \neq OBJ_2 \neq OBJ_3$$

The measure OBJ is a linear combination of TOS and TSD. This measure is important because it is a measure of the overall level of quality of the schedule produced by the algorithm, from the perspectives of both management and labor. Again there is no evidence to suggest a priori rankings.

7.3.6. Interactions among variables

Because there were several factors intended to measure problem difficulty, and because the set of problems reflected those factors, it was deemed necessary that the data be tested for the presence of significant interactions. At the time of the research, there was no prior evidence to suggest which interactions might be significant.

7.4. SELECTION OF EXPERIMENTAL DESIGN

7.4.1. Factor Levels

As discussed above, the problem difficulty factors are Average Worker Utilization (AWU), Average Percentage of Tasks Known (PCTNTN) and the Max-Min Ratio (MMR). Eighteen managers of firms in the service industry were interviewed to determine the number of levels and the value of levels to be used in the experiment and to substantiate assumptions basic to this research. These firms were chosen to be representative of the different types of food service and retail establishments in the locality. Based on the interviews, three levels were chosen for each factor, with the values of the levels based on the average of the responses. The factor "Algorithm" has three levels, Loucks, simulated annealing and tabu search. Those values are shown in table 7-1.

Table 7-1 - Factors and Factor Levels

Factor	Level 1 (Low)	Level 2 (Med)	Level 3 (High)
Average Worker Utilization	$\leq .3$	$> .3$ and $\leq .5$	$> .5$
Average Percentage of Number Of Tasks Known	$\leq .35$	$> .35$ and $\leq .625$	$> .625$
Max-Min Ratio	≤ 2	> 2 and ≤ 5	> 5
Algorithm	Loucks	Simulated Annealing	Tabu Search

7.4.2. *Choice of Design*

Because of the interest in investigating the interactions among factors and because there were no prior expectations regarding interactions, other than that some would be significant, a full factorial design was selected, in order to investigate all possible interactions. Given three levels for each of the factors, the design is $3 \times 3 \times 3$, for a total of 81 treatments.

7.4.3. *Sample size of experiment*

As is widely known, sample size determination is a tradeoff between power of the test, that is the probability of a type II error, and the amount of resources necessary to carry out the experiment. For this investigation, the probability of a type two error was desired to be no more than .05.

A number of methods exist for determining sample size for an experiment, some quite general or "guesstimates", some quite analytical, see Kirk [1982], Larsen and Marx [1981] and others. Montgomery [1984] outlines several analytical methods for

determining sample size using operating characteristic curves. The method of determining sample size chosen for this research relies on estimates of treatment means and overall variance.

7.4.4. Test data

Because no estimates of means or variances were available from past experiment or experience, problems generated for use in the experiment were used to estimate the means and variances required for the calculation of Φ , a variable used in the analytical method for determining sample size using operating characteristic curves.. Four problem types (a total of 12 treatments) were chosen at random to provide estimates of the variance and treatment means. Each of the algorithms was run against 15 problems of each type. Means and variances were calculated for each treatment. Further, treatment effects were calculated for each treatment by subtracting from the treatment mean the grand mean. Using the results of these calculations values of Φ were calculated for each of the five performance measures.

For the purposes of this analysis, the variable chosen was TSD - Sum of squared deviations between workers scheduled and targeted work hours. This variable was chosen because, based on the sample data used, the value of Φ had the largest magnitude of all the performance measures, except for the performance measure time-to-solution (or time). Since the performance measures associated with solution quality were deemed more important than that of time-to-solution, TSD was used for the subsequent analysis. The use of the performance measure with the largest value of Φ provides the most conservative results from the analytical procedure, i. e. the largest sample size.

7.4.5. Analysis

After analysis, the data calculated and given in table 6-2 indicated that a sample size of 6 would be adequate to ensure that the probability of committing a type II error would be less than .05.

Table 7-2 Results of the analytic procedure for determination of sample size.

n	Φ^2	Φ	a(n-1)	β	Power(1- β)
4	2.09882	1.44873	36	0.10	.90
5	2.623525	1.61973	48	0.06	.94
6	3.14823	1.774325	60	0.03	.97

7.4.6. Choice of sample size

Although 12 treatments out of 81 represents a relatively high percentage of total treatments for determination of sample size, some uncertainty remained about how representative were the observed values of both the treatment means and the variances. Because of that uncertainty, and because of additional uncertainty regarding the ability of the algorithms to come to solution on the various problem types, it was decided to increase the sample size to 15. In this investigation, increasing the sample size merely increases the amount of personal computer time spent on solutions, so increasing the sample size was relatively cost free. It was felt that the tradeoff of increasing the total run time was well worth the effort since a larger sample size would increase the power of the experiment.

The total number of computer runs was set at, sample size * number of cells, or $15 * 81 = 1215$.

7.4.7. *Replications for Simulated Annealing*

As discussed in chapter 6, simulated annealing is a randomized search process, that requires the use of a probability distribution associated with the "transition" and therefore produces a different solution each time the algorithm stops (if it is stopped before reaching optimality). Since optimality for the simulated annealing algorithm can only be guaranteed with an infinite number of transitions, the implementation used in this research cannot be guaranteed to find the globally optimal solution. Since global optimality is not assured, and since the algorithm produces a different solution, with a different objective function value, for each run on the same problem, some method to characterize the performance of the algorithm was sought. Both Loucks' heuristic and tabu search are deterministic, that is, the solution generated is the best one for each problem solved, so they do not require repetition.

There are two ways to evaluate simulated annealing's performance - with no repetitions and with repetitions. Using one run, the first solution and associated performance measurement values, is an attractive suggestion. This has the benefit of reduced computer time for the research, but leaves the question of "Is this the best solution that this algorithm will generate?" open. It also rejects the probability that a manager would run the algorithm at least twice to find the better solution. After some discussion, the decision was made to run two or more replications of simulated annealing on each problem for two reasons. First, a single run would not capture indications of simulated annealing's best performance, and second, it is likely that in a firm the scheduler would run the algorithm at least twice, picking the best solution.

The number of replications for simulated annealing for each problem was set at 5, based on two considerations. First, the total time required for 5 replications of simulated annealing is just less than or approximately equal to the time that the tabu search

algorithm takes for completion, that is, five is the largest integer multiple of simulated annealing's run time that does not exceed tabu search's run time, on average. Second, the analytical procedure for determining sample size indicated that for simulated annealing, five replications would give a power of about .95, which was deemed sufficient for this research.

The data used for the statistical analysis for simulated annealing is the best solution of the five replications. The best of the five was chosen on the basis of the lowest objective function value. Ties were broken based on the fastest run time. All the associated values of performance measures of the best solution were used in the statistical analysis.

Some might argue that the average of the five solutions be used in the subsequent analysis, rather than the best solution. This argument was considered and rejected because; 1) scheduling managers were very unlikely to run the algorithm five times and then choose the average solution, and 2) while it is possible to average performance indicators, it is impractical to average two (or more) schedules together. One of the schedules, of the five, will have to be implemented. The obvious choice is the schedule corresponding to the best objective function value. So the idea of an average schedule represented by averages of performances is unworkable.

7.5. GENERATION OF TEST PROBLEMS

The generation of test problems followed the procedure of Loucks. For a full discussion of his procedure, see Loucks [1987], but in general terms, his procedure is to generate a work tour for each worker, to generate a set of available hours for each worker that are inclusive of the work tour, to sum across workers to find the task hours required for each hour of each day, and to use those hour by hour, task by task sums to construct a scheduling problem. As you will note, since the schedule is constructed from an actual schedule, Loucks' procedure provides a problem for which we know the optimal answer. It was felt, however, that Loucks' procedure should be modified for the following reasons.

First, actual demand patterns for workers, in terms of task hours, probably would not have the implicit characteristic of a minimum shift of 3 hours. When the Loucks procedure creates a scheduling problem, it sums the workers scheduled by task hour, with the restriction that all workers shifts will be at least the minimum shift (3 hours). This abnormally restricts the range of the Max-Min Ratio (MMR), or alternatively, reduces the peakedness of the demand pattern. A review of demand patterns published in the literature indicates that frequently the MMR will exceed 10. Measurement of the MMR from the unmodified Loucks procedure revealed a maximum of seven. After the restriction on minimum shift was lifted (by setting the minimum shift = one hour) the modified procedure showed MMR values up to 12. Unfortunately, the modification causes the procedure to produce problems for which the optimal solution is not known.

Last, modification of Loucks' procedure, by removing the minimum shift constraint, would also eliminate any favorable bias toward one algorithm or another because no algorithm would possess prior knowledge about the characteristics of the problem to be solved.

As specified, Loucks' procedure produces problems in 6 of the 27 problem types. To generate the other 21 problem types, the cumulative probability distributions for number of hours scheduled, number of tasks known and window of availability were adjusted, simultaneously and singly. Adjusting the cumulative probability distribution of the number of hours scheduled to increase the average of the number of hours scheduled produces problems with increased average worker utilization. Adjusting the cumulative probability distribution of the number of tasks known to decrease the mean of the number of tasks known produces problems with a lower average number of tasks known. Adjusting the cumulative probability distribution of the window of availability to reduce the average size of the availability window produces problems that have higher average worker utilization. These adjustments were performed to produce the problems for all 27 problem types.

This use of adjusting cumulative probability distributions as a method of producing problems, coupled with the relaxation of the minimum shift constraint(as explained above), causes the problem generator to produce a significant proportion of infeasible problems. Depending on the problem type, as many as 60 problems were generated to find 15 that were feasible. The problems were tested for feasibility by using the program used to generate initial feasible solutions for the simulated annealing and tabu search algorithms as described in chapter 4.

7.6. DATA GATHERING

Once the problems were generated, running the algorithm on the problems commenced. After initial calculation of approximate run times for the algorithms and in view of the total number of problems, additional computing power was arranged for this research. A total of 17 computers were used to collect the data, at three locations. All but five of these computers ran at different speeds. Several test problems were run on all computers to establish the factors by which to adjust the run time of the algorithm on each problem by computer on which it was run.

Loucks' heuristic would not run to completion on 20 of the feasible problems. Upon examination of Loucks' computer code, it was found that the method by which the workers were assigned to hours in Phase 1 of his algorithm, caused unresolvable constraint violations, which in turn caused the algorithm to terminate prematurely. The specific cause of the failures is thought to be related to the emphasis Loucks' algorithm places on minimizing unnecessary over scheduling of workers during the early stages of the algorithm. This caused a reduction of sample size in some cells to 13. To ensure a balanced design, data points were eliminated randomly from cells to equalize the sample size at 13 for all cells. As discussed previously above, the original sample size was set at 15 so that adequate power could be preserved in such a case as this.

The total time for computing on problems spanned 1014 hours of (adjusted to 33MHz, 386 standard time) computer time over the period of about 10 days.

CHAPTER 8 - RESULTS OF EXPERIMENTATION

8.1. INTRODUCTION

Before discussing the results of experimentation based on ANOVA, tests of the underlying assumptions of analysis of variance that were carried out are explained, so that there might be confidence in the results and the ensuing discussion. After the validity of using analysis of variance is established, results of significance tests will be presented and discussed. The level of significance used throughout the remainder of this discussion of statistical tests is $\alpha = .05$. All statistical tests were carried out using SYSTAT for Windows.

8.2. TESTS FOR ASSUMPTIONS OF ANOVA

To have confidence in the results obtained from an analysis of variance, verification that the data conform to the assumptions underlying the analysis of variance is necessary. The three assumptions of analysis of variance to be tested are; 1) that the error residuals are normally distributed, 2) that the error residuals are independently

distributed, and that all treatments have constant variance or that all treatments display homogeneity of variance.

First, normality of error residuals was tested by using normal probability plots of the error residuals. Second, first order autocorrelations are given to test for independence of the distribution of the residuals, in tabular form. Last, homogeneity of variance was tested by plots of residuals against estimates and by Bartlett's test for Homogeneity of Variance.

8.2.1. Normality of error residuals

Montgomery [1984] suggests that a useful procedure for checking the normality assumption is to construct a normal probability plot of the residuals. If the plot resembles a straight line, then the underlying error distribution is normal. Figures 8-1 through 8-5 show normal probability plots of residuals corresponding to each of the performance measures.

8.2.2. Normal Probability Plots of Residuals

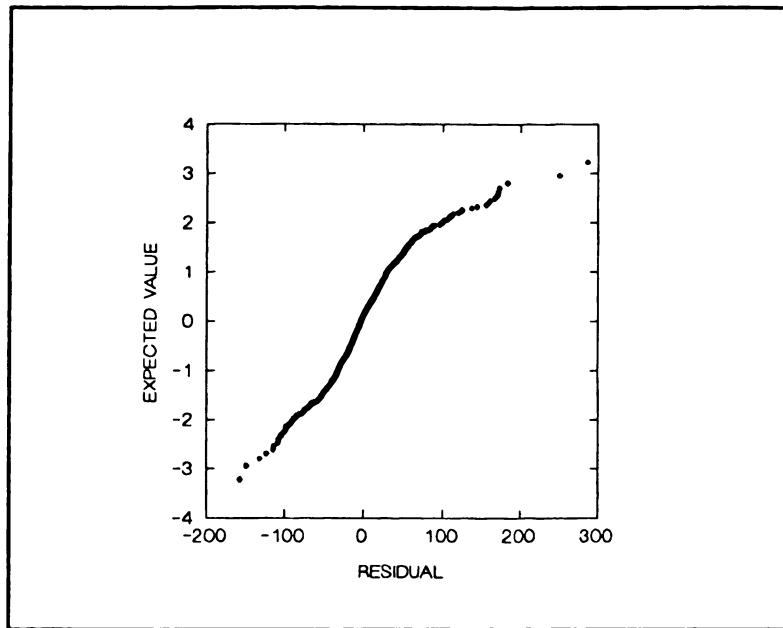


Figure 8-1, Normal probability plot of TOS residuals

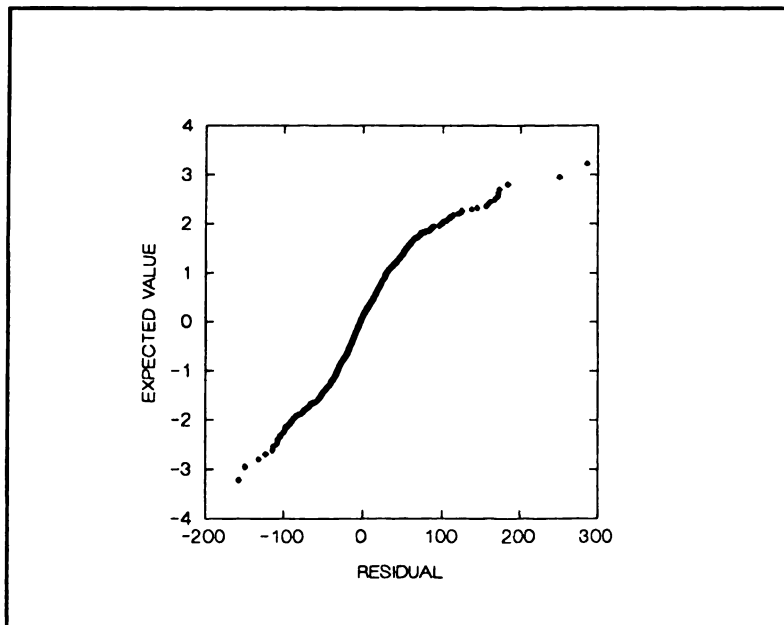


Figure 8-2, Normal probability plot of TSD residuals

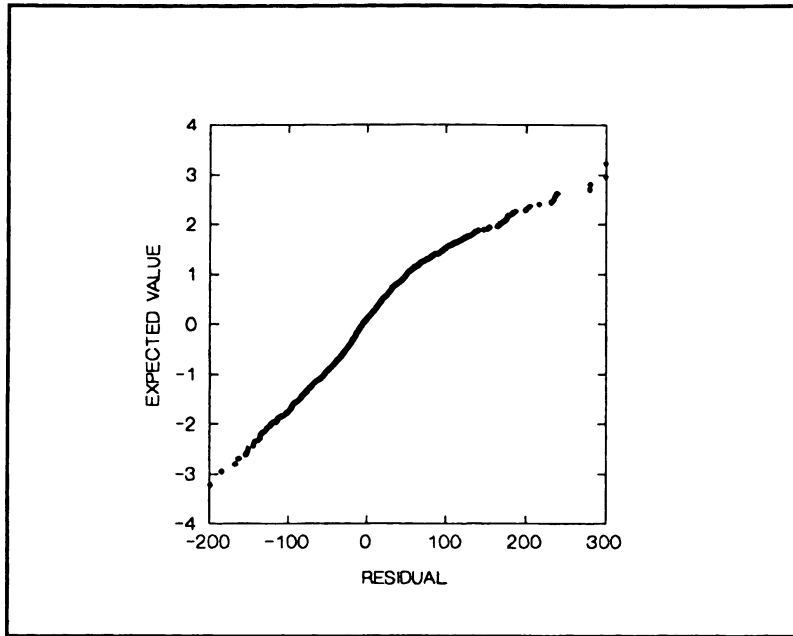


Figure 8-3, Normal probability plot of OBJ residuals

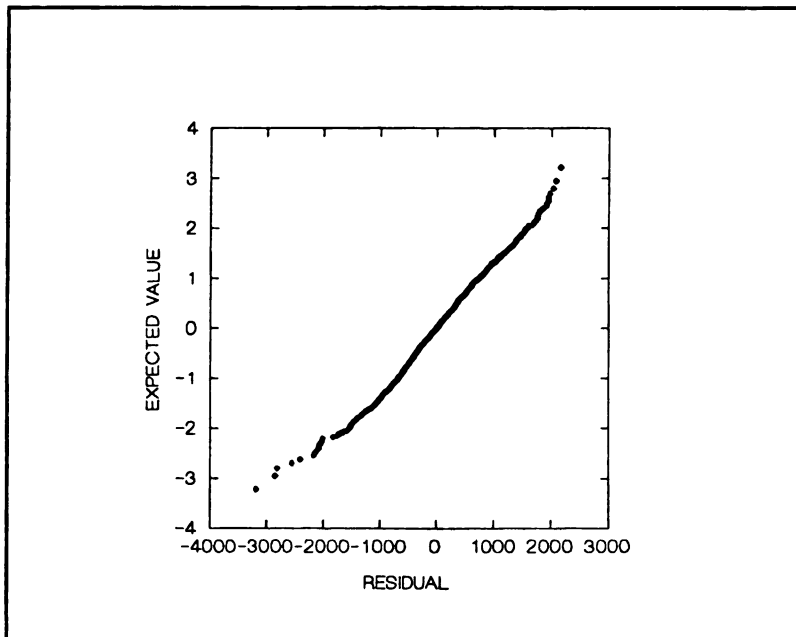


Figure 8-4, Normal probability plot of COST residuals

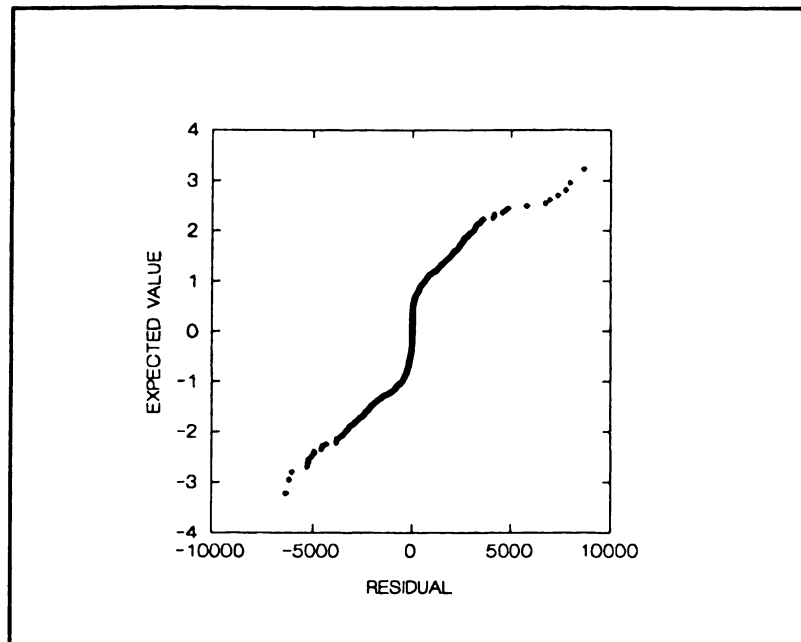


Figure 8-5, Normal probability plot of TIME residuals

As can be seen from figures 8-1 and 8-4, the residuals plot seems to approximate a straight line. However, figures 8-2, 8-3 and 8-5 indicate that the residuals may not be normally distributed since the plots differ appreciably from a straight line. The non-normal distributions for TSD, OBJ, and TIME may indicate that a transformation is desirable.

8.2.3. *Independent distribution of error residuals*

The independent distribution of error residuals can be assessed by measuring the autocorrelation of the data. This is done automatically by SYSTAT. The values of autocorrelation are listed in the following table. There seems to be no significant

autocorrelation, so it may be concluded that the error residuals were independently distributed.

Table 8-1, Autocorrelation Values For Dependent Variables

	TOS	TSD	OBJ	COST	TIME
Autocorrelation	-0.057	-0.002	-0.030	-0.068	0.041

8.2.4. Homogeneity of variance

The plots of residuals vs. estimate for all dependent variables are shown in figures 8-6 through 8-10. Homogeneity of variance is indicated by a random dispersal of residuals around a mean value of zero. There should be no apparent patterns in the data.

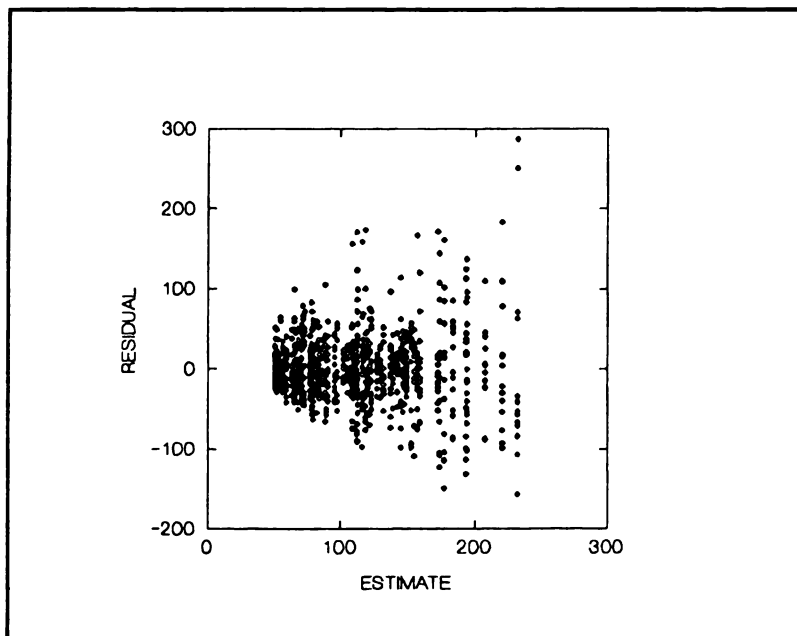


Figure 8-6, Plot of Residual vs. Estimate - TOS

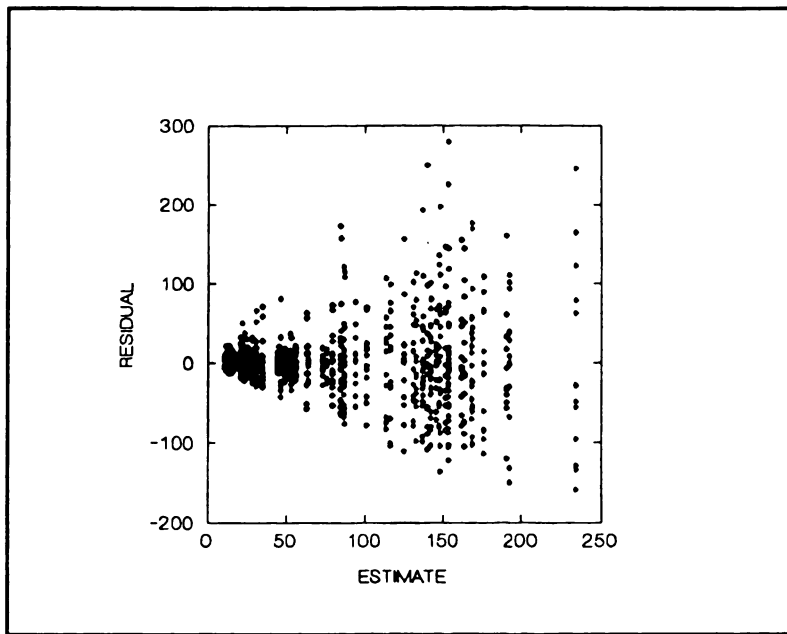


Figure 8-7, Plot of Residual vs. Estimate - TSD

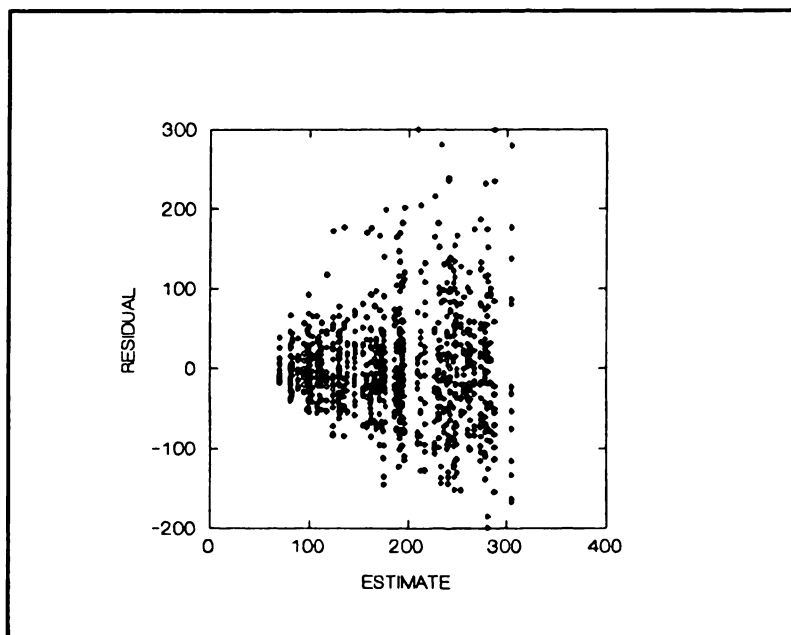


Figure 8-8, Plot of Residual vs. Estimate - OBJ

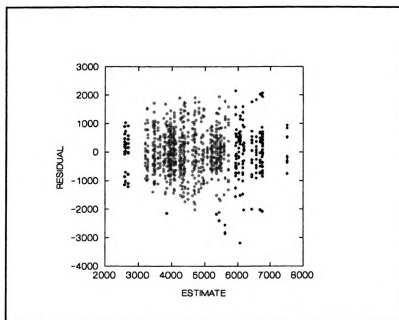


Figure 8-9, Plot of Residual vs. Estimate - COST

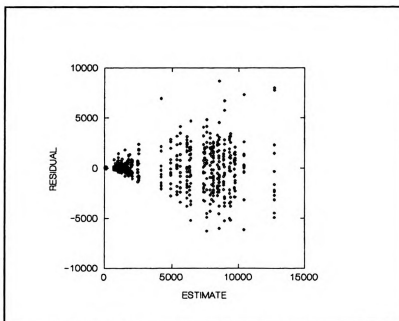


Figure 8-10, Plot of Residual vs. Estimate - TIME

An examination of the figures 8-6 through 8-10 reveals that the plots of residual versus estimate for all dependent variables exhibit a funnel shaped distribution. The presence of funnel shaped distributions is indication that a transformation of the data is desirable.

8.2.5. Bartlett's Test

Bartlett's test for homogeneity of variance tests the hypothesis that the variances of all treatments are equal for each performance measure. This hypothesis is rejected if the test statistic exceeds the appropriate chi-square value.

The test statistics for each of the dependent variables is shown in Table 8-2.

Table 8-2, Bartlett's Test For Homogeneity of Variance

	TOS	TSD	OBJ	COST	TIME
Chi-Square, 80df	60.39	60.39	60.39	60.39	60.39
Test Statistic, 80df	600.381	1334.183	445.358	242.852	3650.298

As can be seen from the table, each of the test statistic values exceeds the chi-square value, so we reject the hypothesis that all variances are equal for each dependent variable.

8.3. BOX AND COX ANALYTICAL METHOD FOR DETERMINING TRANSFORMATION

The evidence presented indicates that the data violate the assumptions of normality of residuals and homogeneity of variance. The usual approach for dealing with non-homogenous variance is to apply a variance stabilizing transformation and then to run the analysis of variance on the transformed data. In this approach, any conclusions postulated apply only to the transformed populations.

A substantial amount of research has been devoted to the subject of the selection of an appropriate transformation. Since there was no theoretical distribution for the observations, there were two options for selection of the appropriate transformation: empirical selection of a transformation or analytical selection of a transformation. A researcher using the empirical selection method would try the various transformations mentioned in the literature, one at a time, looking for a transformation that modified the data in such a manner so as to remove heterogeneity of variance and non-normality of error residuals (see Kirk, [1982] for example).

The use of the analytical method for selection of a transformation performs a series of predefined transforms on the data, measuring the error sum of squares for each. The intent is to generate a set of points from which the minimum error sum of squares can be estimated. The selection of the transform is based on the minimum error sum of squares. For this research, the analytical approach for selection of a transformation was chosen.

8.3.1. Explanation of Analytical Determination of Transformation

Box and Cox [1964] have demonstrated how the transformation parameter λ , in $y^* = y^\lambda$ may be estimated using the method of maximum likelihood, where y represents the data to be transformed, y^* is the transformed data, and λ is the exponent of the

transform. The procedure involves of performing a standard analysis of variance, for a range of values for λ , on the equation(s):

$$y^\lambda = (y^{\lambda-1})/(\lambda \bar{y}^{\lambda-1}) \text{ for } \lambda \neq 0, \text{ and}$$

$$\bar{y} \ln y \text{ for } \lambda = 0,$$

Equations 8-1

where $\bar{y} = \ln^{-1}[(1/n) \sum \ln y]$ is the geometric mean of the (n) observations y.

The maximum likelihood estimate of λ is the value for which the error sum of squares is a minimum. This is designated as SSe(L) in table 8-2. We cannot select the value of λ by directly comparing the error sum of squares from analyses of variance on y^λ , because for each value of λ the error sum of squares is measured on a different scale. For a more complete discussion of this procedure, see Montgomery [1984].

8.3.2. *Results of the Analytical Determination of Transformation*

For this inquiry, nine values of λ were chosen as shown in table 8-3. The SSe(λ) values are shown next to the λ values under the appropriate column heading for dependent variable. Each SSe(λ) value represents the error sum or squares from performing an analysis of variance on the dependent variable as transformed by equation 8-1. In table 8-3, the row SSe(L) represents the minimum value found. If the minimum SSe(λ) is near the value $\lambda = 1$, this implies that the data do not support the need for transformation. It can be seen from table 8-3 that none of the minimum values of SSe(λ) are close to the value of $\lambda = 1$. This implies that transformation is desirable for each dependent variable and that the value of λ is close to the λ associated with the minimum SSe(λ). At the bottom of the table the λ associated with the minimum SSe(λ) is listed

along with the transformation determined by using the Box and Cox procedure. The remainder of the data analysis will use the dependent variables transformed as indicated.

Table 8-3, Results of Procedure for Analytic Determination of Transformation

λ	TSD	TOS	OBJ	COST	TIME
1.00	2038104	1799965	4255743	6.01E+08	2.02E+09
0.75	1255173	1500566	3657613	5.98E+08	6.48E+08
0.50	876023	1345510	3277028	6.05E+08	2.24E+08
0.25	738865	1298441	3076010	6.24E+08	8.87E+07
0.00	820865	1346459	3039912	6.57E+08	5.14E+07
-0.25	1289105	1495409	3176398	7.07E+08	7.34E+07
-0.50	2.88E+06	1770814	3519015	7.79E+08	2.22E+08
-0.75	8.62E+06	2224573	4136779	8.80E+08	8.51E+08
-1.00	3.19E+07	2949697	5153033	1.02E+09	3.48E+09
SSE(L)	738865	1298441	3039912	5.98E+08	5.14E+07
λ at SSE(L)	0.25	0.25	0.00	0.75	0.00
Transformation	$y^{0.25}$	$y^{0.25}$	$\ln(y)$	$y^{0.75}$	$\ln(y)$

8.3.3. Normal probability plots of residuals

After transformation, the variables are expected to adhere more closely to the assumptions of the analysis of variance, those are, normality of error residuals, independently distributed error residuals and homogeneity of variance of the residuals. The analysis that follows tests the transformed variables against the assumptions of the analysis of variance model. The tests follow the pattern set earlier, tests for normality of residuals, results of autocorrelation analysis, and tests for homogeneity of variance.

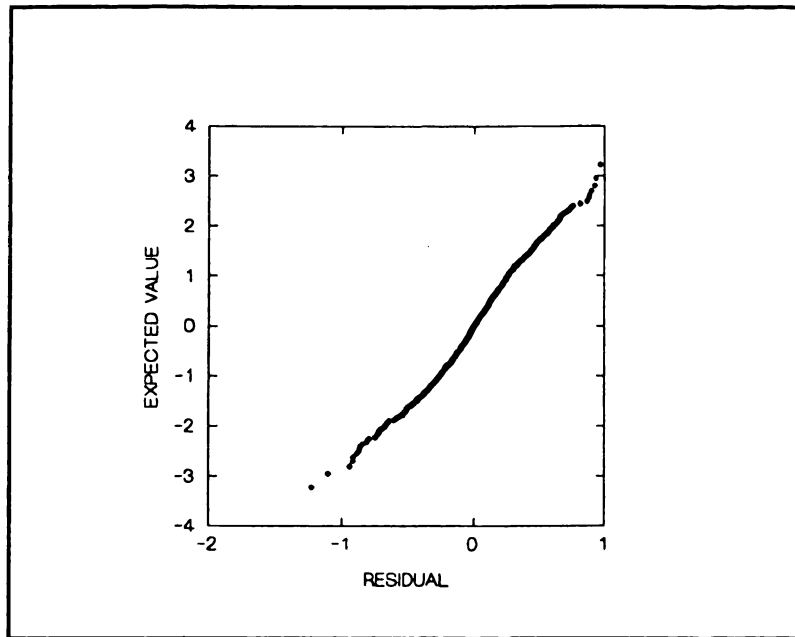


Figure 8-11, Normal probability plot of TOS residuals After Transformation

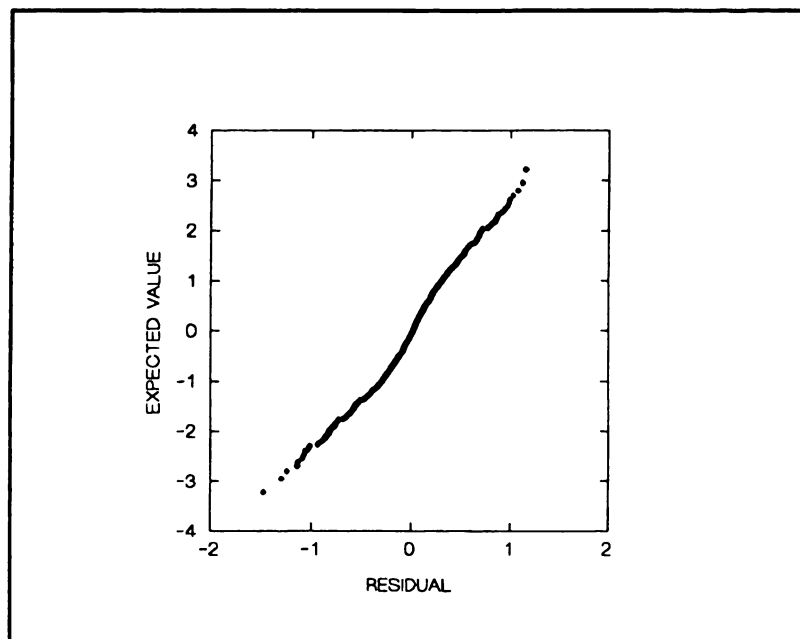


Figure 8-12, Normal probability plot of TSD residuals After Transformation

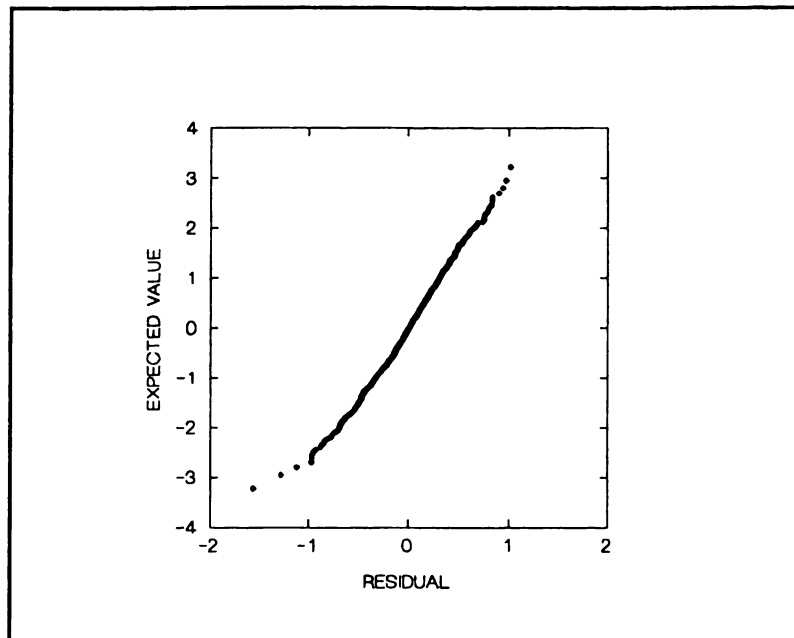


Figure 8-13, Normal probability plot of OBJ residuals After Transformation

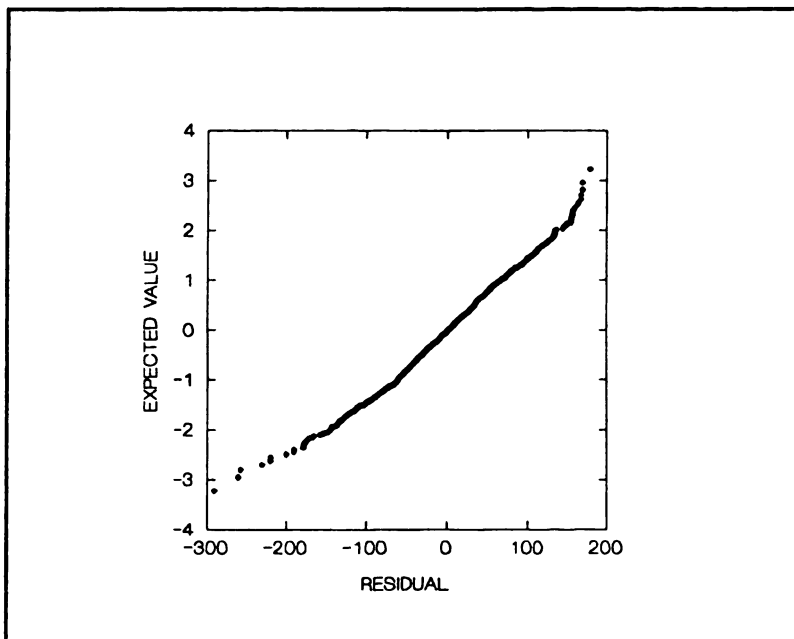


Figure 8-14, Normal probability plot of COST residuals After Transformation

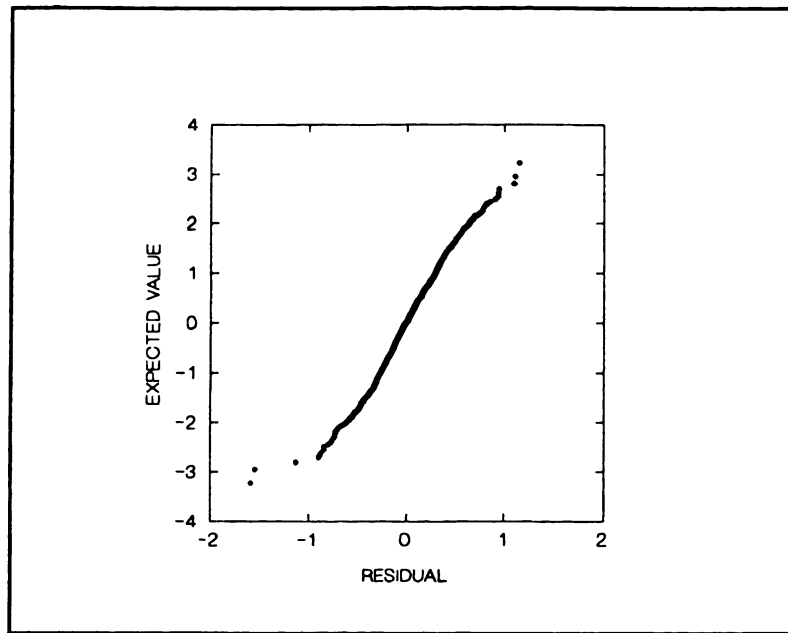


Figure 8-15, Normal probability plot of TIME residuals After Transformation

All normal probability plots now indicate that the residuals of the transformed dependent variables are approximately normally distributed (i. e. no major departures from normality are indicated by the plots).

8.3.4. *Independent distribution of error residuals*

The independent distribution of error residuals can be assessed by measuring the autocorrelation of the data. The values of autocorrelation, after transformation, are listed in the following table. There seems to be no autocorrelation, and it can be concluded that the error residuals are independently distributed.

Table 8-4, Autocorrelation of transformed residuals

	TOS	TSD	OBJ	COST	TIME
Autocorrelation	-0.007	0.047	0.003	-0.052	-0.047

8.3.5. Homogeneity of Variance

After transformation of the data, verification of homogeneity of variance is necessary to establish credibility of results. The tests for homogeneity of variance are the same as used before; plots of residuals versus estimates and Bartlett's test for homogeneity of variance. Figures 8-16 through 8-20 present the plots of residual versus estimate for the dependent variables.

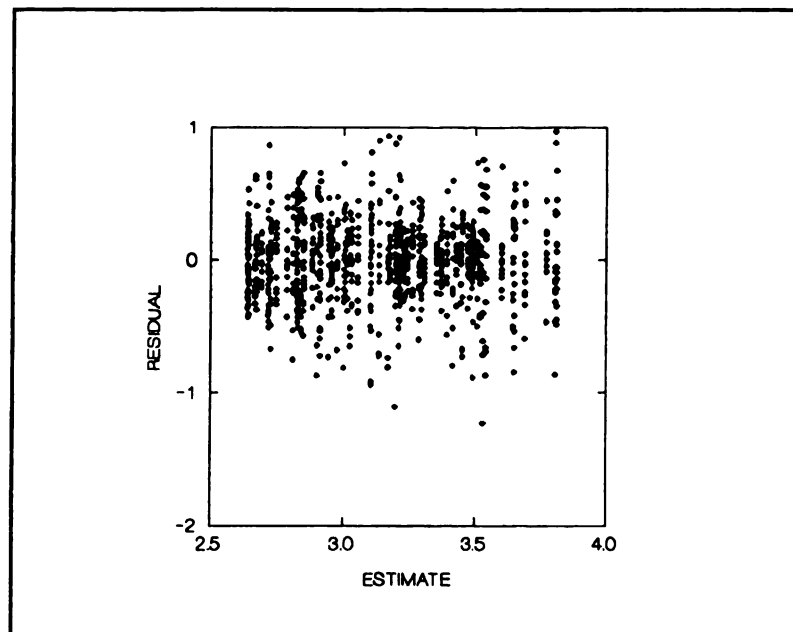


Figure 8-16, Plot of Residual vs. Estimate - TOS After Transformation

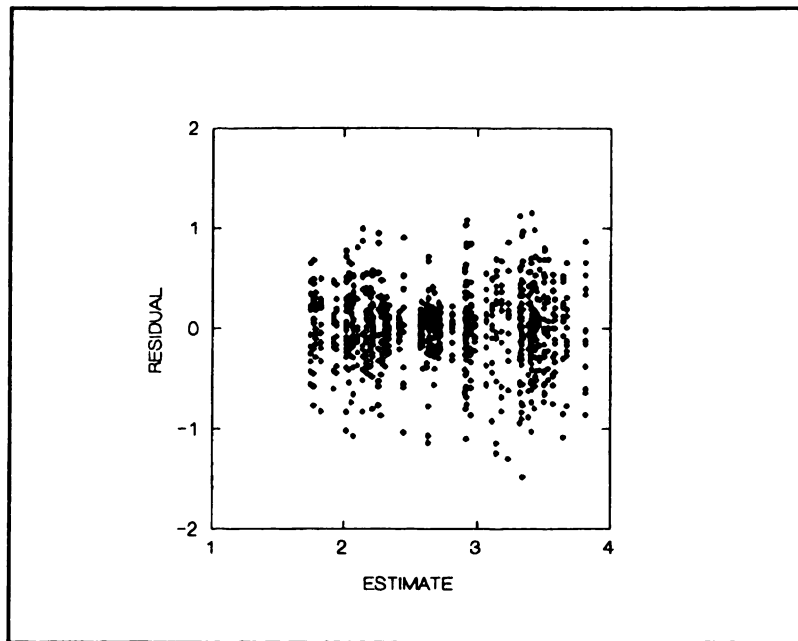


Figure 8-17, Plot of Residual vs. Estimate - TSD After Transformation

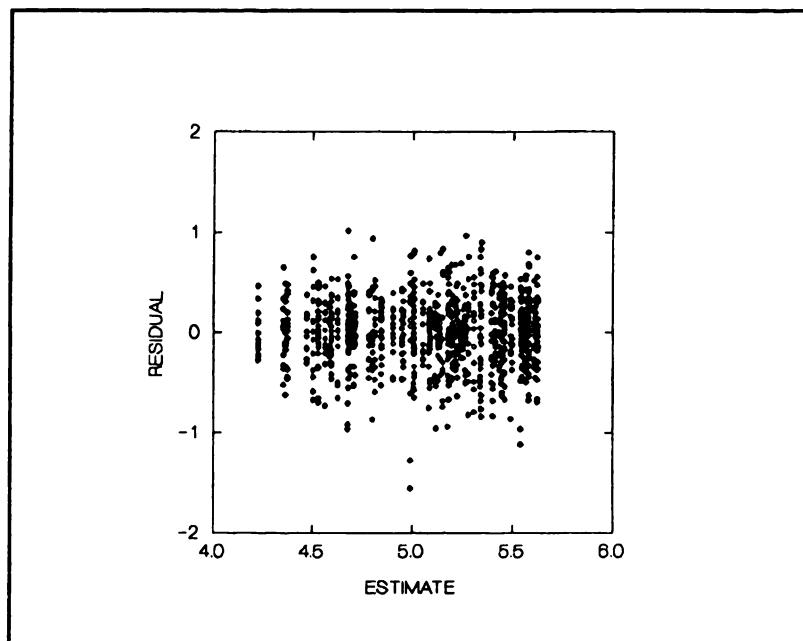


Figure 8-18, Plot of Residual vs. Estimate - OBJ After Transformation

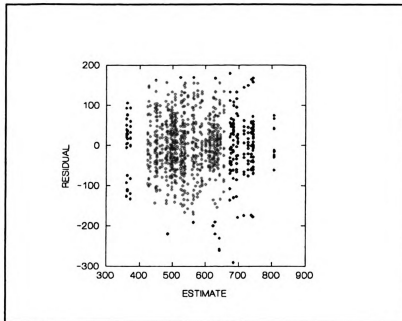


Figure 8-19, Plot of Residual vs. Estimate - COST After Transformation

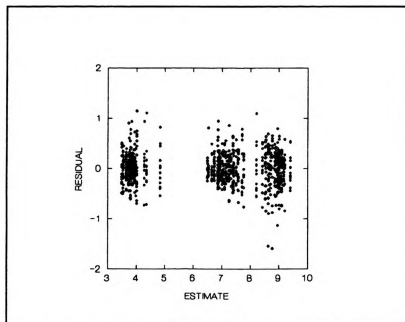


Figure 8-20, Plot of Residual vs. Estimate - TIME After Transformation

The plots of estimate vs. residual for the transformed dependent variables exhibit no obvious patterns and hence no obvious heterogeneity of variance.

8.3.6. *Bartlett's test after transformations*

As stated earlier, Bartlett's test for homogeneity of variance tests the hypothesis that the variances of all the treatments are equal for each performance measure. We reject this hypothesis if the test statistic exceeds the appropriate chi-square value.

The statistics for each of the transformed dependent variables is shown in Table 8-5.

Table 8-5, Bartlett's Test For Homogeneity of Variance

	TOS	TSD	OBJ	COST	TIME
Chi-Square, 80df	60.39	60.39	60.39	60.39	60.39
Test Statistic, 80df	348.413	481.981	201.125	236.860	288.969

As can be seen from the table each of the test statistic values exceeds the chi-square value, so we reject the hypothesis that all variances are equal for all transformed dependent variables.

8.4. JUSTIFICATION FOR USE OF ANALYSIS OF VARIANCE

At this juncture, it has been shown that the transformed dependent variables have normally and independently distributed error residuals, and that the variance among cells is heterogeneous. Therefore, the data still violate the assumptions of the analysis of variance model. Both Montgomery [1984] and Kirk [1982] state that for a balanced, fixed effects design, the F test is only slightly affected by heterogeneous variances if the error residuals are normally distributed. Since the design is balanced, at a sample size of 13 (after random deletion of data points in treatment cells to balance the design), and since the error residuals are normally and independently distributed, this research will rely on the robustness of the F test to heterogeneity of variance as the justification for using analysis of variance.

8.5. RESULTS OF ANOVA

The data, collected as described in chapter 7, were analyzed using the multi-factor ANOVA routines in SYSTAT for Windows. A summary of significant effects is given in Table 8-6.

Table 8-6 Analysis Of Variance Results
indicates significance at $p < .01$, "y" indicates significance at $p < .05$.

Source	TOS	TSD	OBJ	COST	TIME
AWU	Y	Y	Y	Y	Y
PCTNTN	Y	Y	Y	Y	N
MMR	Y	Y	y	Y	Y
ALGORITHM	Y	Y	Y	Y	Y
AWU*PCTNTN	y	N	y	Y	Y
AWU*MMR	y	Y	Y	Y	Y
AWU*ALGORITHM	Y	Y	Y	Y	Y
PCTNTN*MMR	Y	Y	N	Y	y
PCTNTN*ALGORITHM	Y	Y	Y	N	Y
MMR*ALGORITHM	Y	Y	Y	N	Y
AWU*PCTNTN*MMR	N	Y	Y	Y	Y
AWU*PCTNTN*ALGORITHM	Y	N	Y	N	Y
AWU*MMR*ALGORITHM	Y	Y	Y	N	Y
PCTNTN*MMR*ALGORITHM	N	Y	Y	N	Y
AWU*PCTNTN*MMR*ALGORITHM	Y	Y	Y	N	Y

As can be seen from table 8-6, the all main effects and the majority of interaction effects are significant at the $p < .01$ level. All interaction effects were significant with one or another dependent variable(s).

8.5.1. Detailed Results

In this section, each of the hypotheses proposed will be considered. As was noted before, a hypothesis was constructed for each of the dependent variables. In all cases direct statements cannot be offered to reject or accept any hypothesis because of the presence of significant interactions, Kirk [1982]. The data do offer supporting evidence for hypotheses, as discussed below.

In order to compare the performance of the solution methods by problem type, a Tukey HSD multiple comparison test was performed on the data representing each of the five performance measures. This test compares each of the 81 treatments against all the other treatments, testing for significant differences among the treatment means. There is very little agreement among statisticians about the relative merits of the various multiple comparison procedures. Tukey's HSD multiple comparison test was chosen for several reasons. First, the data meet the assumptions of the test. Second, the Tukey test is more powerful than other tests for large number of comparisons. Third, the Tukey HSD multiple comparison test controls the error at α for the entire collection of tests.

Following the presentation of relevant statistics for each dependent variable, interactions of interest to this research will be examined and discussed. Since this research is primarily interested in a comparison of algorithms, emphasis will be placed on discerning the main effect of ALGORITHM in the presence of interactions.

The reader is reminded that when considering a hypothesis, the subscript attached to the dependent variable represents an algorithm. For example, $TIME_1$ refers to the TIME for Loucks heuristic, $TIME_2$ refers to the TIME for simulated annealing and $TIME_3$ refers to the TIME for tabu search.

In the charts and discussion that follows an abbreviation for problem types will be used consisting of a three letter code. The first letter in the code stands for the level of Average Worker Utilization, H for high, M for medium and L for low. The second letter in the abbreviation stands for the level of average percentage of tasks known per worker, using the same letters as the abbreviation for AWU. The last letter stands for the level of max-to-min ratio, again using the same letters, H, M, and L. As an example, the problem with the label, HLM, would be a problem type having a high level (H) of average worker utilization, a low level (L) of average percentage of tasks known per worker, and a medium (M) level of max-to-min ratio.

8.5.1.1. TIME - Speed of Algorithm

The hypotheses relating to this aspect of algorithm performance is:

$$H_0 : \text{TIME}_1 = \text{TIME}_2 = \text{TIME}_3$$

$$H_1 : \text{TIME}_1 \neq \text{TIME}_2 \neq \text{TIME}_3$$

This hypothesis tests whether there are significant speed differences among the algorithms. Since there are significant interactions among the factors the hypothesis cannot be tested directly.

Results of ANOVA on TIME, Transformed. The analysis of variance for the dependent variable TIME as presented in table 8-7 indicates that all main effects and interaction effects are significant at $p < .001$, with two exceptions. First, the main effect PCTNTN, or average percentage of tasks known per worker, was not significant, and second, the interaction of PCTNTN*MMR, or the interaction of average percentage of tasks known per worker with the max-min ratio, was significant at the $p < .05$ level. The multiple r correlation coefficient was 0.990 (squared value = 0.980), indicating that the factors and interactions accounted for most of the variation in the dependent variable TIME.

Examination of the mean-square for each of the effects indicated that the mean-square for the main effect ALGORITHM is at least 2 orders of magnitude larger than any other effect and about 3 orders of magnitude larger than the other main effects. Mean-squares for all effects are an order of magnitude larger than the mean-square for error for the variable TIME.

Table 8-7. ANOVA on TIME, Transformed

ANOVA on TIME, Transformed					
LEVELS ENCOUNTERED DURING PROCESSING ARE:					
AWU\$:	High	Low	Med		
PCTNTN\$:	High	Low	Med		
MMR\$:	High	Low	Med		
ALGORITHM\$:	Loucks	Sim_Ann	Tabu_Srch		
DEP VAR: TIME	N: 1053	MULTIPLE R: 0.990	SQUARED MULTIPLE R: 0.980		
ANALYSIS OF VARIANCE					
SOURCE	SUM-OF-SQUARES	DF	MEAN-SQUARE	F-RATIO	P
AWU\$	4.199	2	2.100	21.515	0.000
PCTNTN\$	0.305	2	0.153	1.564	0.210
MMR\$	4.607	2	2.303	23.603	0.000
ALGORITHM\$	4456.638	2	2228.319	22833.320	0.000
AWU\$*PCTNTN\$	4.645	4	1.161	11.898	0.000
AWU\$*MMR\$	2.178	4	0.545	5.580	0.000
AWU\$					
*ALGORITHM\$	17.080	4	4.270	43.753	0.000
PCTNTN\$*MMR\$	1.012	4	0.253	2.592	0.035
PCTNTN\$					
*ALGORITHM\$	19.868	4	4.967	50.897	0.000
MMR\$					
*ALGORITHM\$	12.223	4	3.056	31.312	0.000
AWU\$*PCTNTN\$					
*MMR\$	4.800	8	0.600	6.148	0.000
AWU\$*PCTNTN\$					
*ALGORITHM\$	16.451	8	2.056	21.071	0.000
AWU\$*MMR\$					
*ALGORITHM\$	5.094	8	0.637	6.524	0.000
PCTNTN\$*MMR\$					
*ALGORITHM\$	5.183	8	0.648	6.639	0.000
AWU\$*PCTNTN\$					
*MMR\$					
*ALGORITHM\$	12.414	16	0.776	7.950	0.000
ERROR	94.858	972	0.098		
DURBIN-WATSON D STATISTIC 2.090					
FIRST ORDER AUTOCORRELATION -.047					

TIME by Algorithm vs. Problem Type. As can be seen from figure 8-21, the time to completion for Loucks' algorithm is always less than simulated annealing which is always less than tabu search (indicating the presence of an ordinal interaction). The Tukey HSD multiple comparison test ($3 \times 3 \times 3$, all factors, all factor levels) verifies that these visual results are, with two exceptions, statistically significant (for problems LLH and MLL, the differences are not significant). The comparison was carried out by looking for significant differences between solution methods on a pairwise basis for each of the problem types. Such pairwise differences were found to be significant for all pairs except between tabu search and simulated annealing on problem types LLH and MLL.

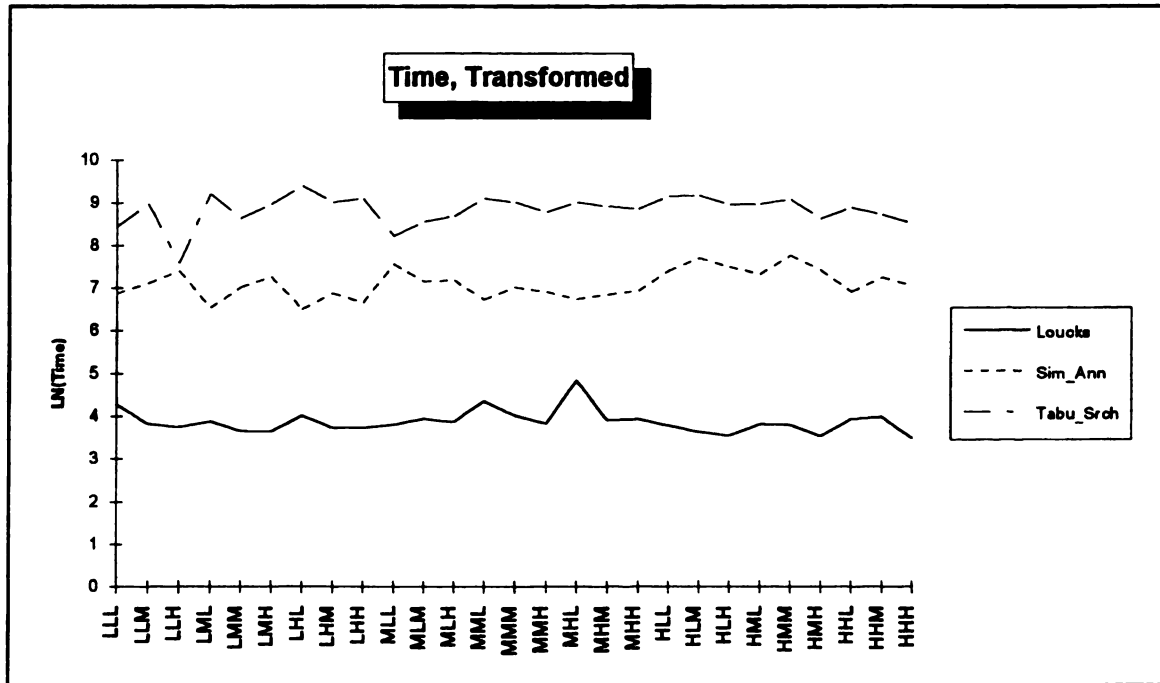


Figure 8-21, Plot of TIME by Algorithm vs. Problem Type

While nothing statistical can properly be said about other differences in the performance of the algorithms on TIME-to-completion, examination of the charts leads to several speculations. The performance of tabu search seems to exhibit a pattern of

better performance when the problem type has a high level of MMR. Similarly, simulated annealing seems to perform better (than others in the problem type group, i.e. HML vs. HML and HMM) with problems with a low level of MMR, whereas Loucks' heuristic performs worse with a low level of MMR.

Due to the ordinal interaction effects (an ordinal interaction is the case when the rank order of the treatment effects remains constant, Pedhazur [1982]), we cannot directly test the hypothesis. The data does provide evidence in support of the alternate hypothesis H_1 , that is, that the TIME-to-completion for each of the algorithms is not equal.

Table 8-8, Table of the best TIME performance on problem type by algorithm .

TIME - Best									
MMR	AWU			AWU			AWU		
	Low	Low	Low	Med	Med	Med	High	High	High
	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
Low	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks
Med	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks
High	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks	Loucks

Table 8-9, Table of the worst TIME performance on problem type by algorithm.

TIME - Worst									
MMR	AWU			AWU			AWU		
	Low	Low	Low	Med	Med	Med	High	High	High
	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
Low	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S
Med	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S
High	Tie	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S	Tabu_S

As can be seen from tables 8-8 and 8-9, Loucks' heuristic always had the best TIME-to-completion and tabu search always had the worst TIME-to-completion.

8.5.1.2. COST - Total Manpower Cost of the Schedule.

To test for differences in COST, the following hypotheses were proposed:

$$H_0 : \text{COST}_1 = \text{COST}_2 = \text{COST}_3$$

$$H_1 : \text{COST}_1 \neq \text{COST}_2 \neq \text{COST}_3$$

Results of ANOVA on COST, Transformed. The results of analysis of variance for the dependent variable COST show that all main effects were significant at $p < .001$. The interaction effects AWU*PCTNTN, AWU*MMR, PCTNTN*MMR, and AWU*PCTNTN*MMR are significant at $p < .001$ and AWU*ALGORITHM is significant at $p < .01$. Most of the interaction effects were not significant at $p < .05$. An examination of the mean-square for the each of the effects show that the mean-square for PCTNTN is twice as large as for AWU and more than an order of magnitude larger than any other effect. The multiple r correlation is 0.813 (squared value = 0.660), indicating that the factors and their interactions accounted for most of the variation in the dependent variable COST.

No evidence can be found for rejection of the null hypothesis,

$H_{20} : \text{COST}_1 = \text{COST}_2 = \text{COST}_3$. Statistically, we cannot support the idea that one or more algorithms is different in COST performance than any other.

Table 8-10. ANOVA on COST, Transformed

ANOVA on COST, Transformed					
LEVELS ENCOUNTERED DURING PROCESSING ARE:					
AWU\$:	High	Low	Med		
PCTNTN\$:	High	Low	Med		
MMR\$:	High	Low	Med		
ALGORITHM\$:	Loucks	Sim_Ann	Tabu_Srch		
DEP VAR: COST	N: 1053	MULTIPLE R: 0.813	SQUARED MULTIPLE R: 0.660		
ANALYSIS OF VARIANCE					
SOURCE	SUM-OF-SQUARES	DF	MEAN-SQUARE	F-RATIO	P
AWU\$	2428046.223	2	1214023.111	235.282	0.000
PCTNTN\$	5234622.308	2	2617311.154	507.245	0.000
MMR\$	246073.570	2	123036.785	23.845	0.000
ALGORITHM\$	182134.677	2	91067.339	17.649	0.000
AWU\$*PCTNTN\$	146193.949	4	36548.487	7.083	0.000
AWU\$*MMR\$	445239.567	4	111309.892	21.572	0.000
AWU\$					
*ALGORITHM\$	68764.822	4	17191.205	3.332	0.010
PCTNTN\$*MMR\$	494109.619	4	123527.405	23.940	0.000
PCTNTN\$					
*ALGORITHM\$	41125.660	4	10281.415	1.993	0.094
MMR\$					
*ALGORITHM\$	43090.215	4	10772.554	2.088	0.080
AWU\$*PCTNTN\$					
*MMR\$	252429.220	8	31553.653	6.115	0.000
AWU\$*PCTNTN\$					
*ALGORITHM\$	18298.392	8	2287.299	0.443	0.895
AWU\$*MMR\$					
*ALGORITHM\$	67488.884	8	8436.111	1.635	0.111
PCTNTN\$*MMR\$					
*ALGORITHM\$	32790.962	8	4098.870	0.794	0.608
AWU\$*PCTNTN\$					
*MMR\$					
*ALGORITHM\$	44171.530	16	2760.721	0.535	0.930
ERROR	5015379.444	972	5159.855		
DURBIN-WATSON D STATISTIC	2.102				
FIRST ORDER AUTOCORRELATION	-.052				

COST by Algorithm vs. Problem Type .Figure 8-22 plots the COST of the schedules produced by the algorithms, by problem type.

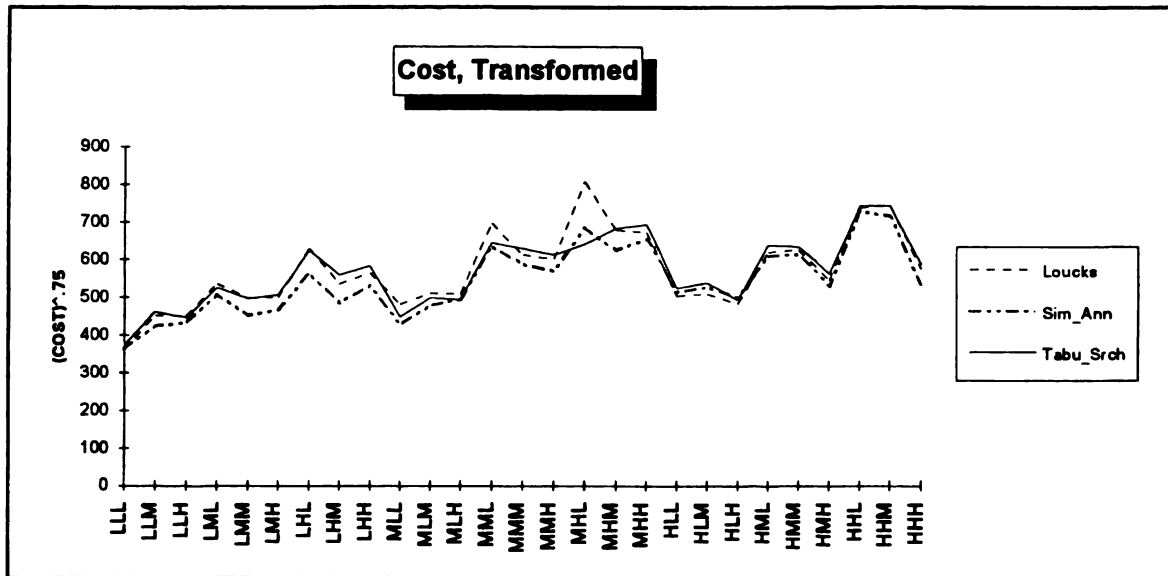


Figure 8-22, Plot of COST by Algorithm vs. Problem Type

Upon examination of the chart, the first pattern that might be noticed is that the COST performance of the three algorithms seems to be highly correlated across problem type. This is, indeed, the case. The correlation coefficients are shown in table 8-11.

Table 8-11, Correlation coefficients on COST performance between Algorithms

Correlation Coefficients			
	<i>Loucks - COST</i>	<i>Sim_Ann - COST</i>	<i>Tabu_Srch - COST</i>
Loucks - COST	1.0		
Sim_Ann - COST	0.96	1.0	
Tabu_Srch - COST	0.93	0.97	1.0

As can be seen from table 8-10, all correlations are very high, over .93. This indicates that COST performance is more influenced by problem type than by algorithm type.

It is also apparent, from the chart, that the COST performance varies by problem type. If we group the problem types, first by AWU (the first letter) and then by PCTNTN (the second letter), we note that when PCTNTN is low, the COST performance is lowest, in the AWU group. This speculation is borne out statistically. A Tukey HSD multiple comparison test (3x3x3x3, all factors, all factor levels) was performed on the COST data. The Tukey test indicates that the COST performance for all the algorithms on problems LLL, LLM, and LLH is lower than problem types LML, LMM, LMH, LHL, LHM, and LHH at $p < .001$.

Another pattern in the data is that it seems that the COSTs are highest when problem level PCTNTN is highest, and medium when the problem type has a level of PCTNTN of medium. This speculation has only partial statistical support across problem types.

Table 8-12, Best COST performance across problem types by algorithm.

COST - Best									
AWU			AWU			AWU			
Low			Med	Med	Med	High			
MMR	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie
	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie
High	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie

Inspection of table 8-12 reveals that across all problem types there were ties for best performance.

Table 8-13, Worst COST performance across problem type by algorithm

COST - Worst									
AWU			AWU			AWU			
Low	Low	Low	Med	Med	Med	High	High	High	
MMR	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Tie	Tie	Tie	Tie	Tabu_S	Tie	Tie	Tie
	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie
Low	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie
Med	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie
High	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tie

As can be seen from table 8-13, tabu search has the only worst performances at problem type MHL. All others are tied for both best and worst performance.

8.5.1.3. TOS - Total man-hours of overstaffing.

The hypotheses to test total man-hours of overstaffing were formulated as follows:

$$H_0 : TOS_1 = TOS_2 = TOS_3$$

$$H_1 : TOS_1 \neq TOS_2 \neq TOS_3$$

As with all other hypotheses proposed in this research, no direct test can be made nor can general, statistically valid statements be made about main or interaction effects because of the presence of significant disordinal interaction effects (a disordinal interaction is present when an interaction changes the rank order of the treatment effects, Pedhazur [1982]). As will be shown below, evidence can be offered in support of the alternate hypothesis.

Results of ANOVA on TOS, Transformed. The analysis of variance for the dependent variable TOS indicates that all main effects are significant at $p < .001$. The

interaction effects, AWU*ALGORITHM, PCTNTN*ALGORITHM, PCTNTN*MMR, MMR*ALGORITHM, AWU*PCTNTN*ALGORITHM and AWU*MMR*ALGORITHM are significant at $p < .001$. The interaction effects, AWU*PCTNTN, AWU*MMR, and AWU*PCTNTN*MMR*ALGORITHM are significant at $p < .05$. Examination of the mean-square of all effects indicates that the factor ALGORITHM has a mean square nearly an order of magnitude larger than any other main effect. Only the interaction effect PCTNTN*MMR*ALGORITHM has a lower mean-square than error. The multiple r correlation of 0.734 (squared value = 0.539) indicated that the factors and interactions accounted for most of the variation in the variable TOS. Complete ANOVA results for TOS are given in table 8-14.

Table 8-14. ANOVA on TOS, Transformed.

ANOVA on TOS, Transformed.					
LEVELS ENCOUNTERED DURING PROCESSING ARE:					
AWU\$:	High	Low	Med		
PCTNTN\$:	High	Low	Med		
MMR\$:	High	Low	Med		
ALGORITHM\$:	Loucks	Sim_Ann	Tabu_Srch		
DEP VAR: TOS	N: 1053	MULTIPLE R: 0.734	SQUARED MULTIPLE R: 0.539		
ANALYSIS OF VARIANCE					
SOURCE	SUM-OF-SQUARES	DF	MEAN-SQUARE	F-RATIO	P
AWU\$	3.024	2	1.512	16.237	0.000
PCTNTN\$	4.368	2	2.184	23.456	0.000
MMR\$	2.269	2	1.135	12.186	0.000
ALGORITHM\$	26.243	2	13.121	140.918	0.000
AWU\$*PCTNTN\$	1.170	4	0.292	3.141	0.014
AWU\$*MMR\$	1.082	4	0.271	2.905	0.021
AWU\$					
*ALGORITHM\$	13.547	4	3.387	36.373	0.000
PCTNTN\$*MMR\$	1.805	4	0.451	4.846	0.001
PCTNTN\$					
*ALGORITHM\$	36.714	4	9.179	98.575	0.000
MMR\$					
*ALGORITHM\$	4.218	4	1.054	11.324	0.000
AWU\$*PCTNTN\$					
*MMR\$	1.129	8	0.141	1.516	0.147
AWU\$*PCTNTN\$					
*ALGORITHM\$	2.574	8	0.322	3.455	0.001
AWU\$*MMR\$					
*ALGORITHM\$	3.490	8	0.436	4.685	0.000
PCTNTN\$*MMR\$					
*ALGORITHM\$	0.625	8	0.078	0.838	0.569
AWU\$*PCTNTN\$					
*MMR\$					
*ALGORITHM\$	3.370	16	0.211	2.262	0.003
ERROR	90.506	972	0.093		
DURBIN-WATSON D STATISTIC 2.013					
FIRST ORDER AUTOCORRELATION -.007					

TOS by Algorithm vs. Problem Type. Figure 8-22 plots the total number of hours overstaffed that each algorithm generated in the schedules produced, by problem type.

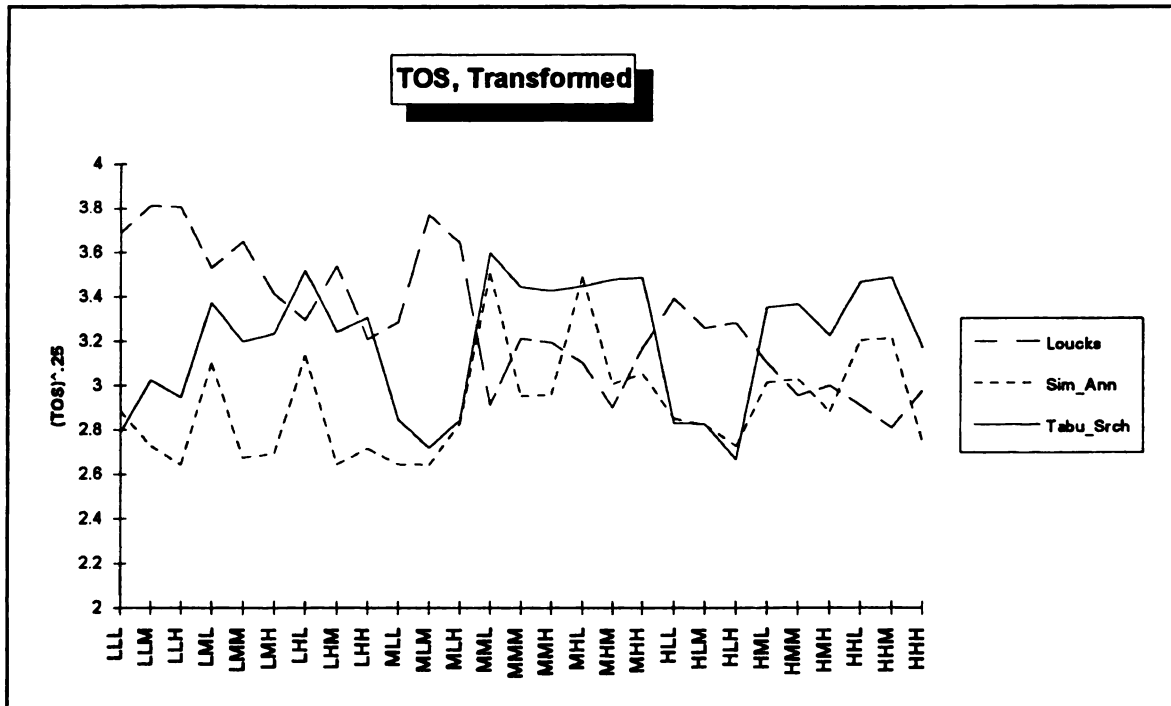


Figure 8-22, Plot of TOS by Algorithm vs. Problem Type

Inspection of the chart indicated that there may be differences between Loucks' algorithm and the other two algorithms when problem type has a low level of the factor PCTNTN. A 3x3x3x3 (all factor, all factor level) Tukey HSD multiple comparison test was performed on the TOS data. The results indicate that differences in total man-hours of overstaffing for problem types LLL, LLM, LLH, MLL, MLM, MLH, HLL, and HLH are significant between Loucks' heuristic and the pair, simulated annealing and tabu search. The exceptions to the "rule" that tabu search and simulated annealing outperform Loucks' heuristic on problems with a low level of the factor PCTNTN are that for problem type HLM there are no significant differences among the algorithms and for problem type MLL the difference between Loucks' heuristic and tabu search is not significant. There are no significant differences between simulated annealing and tabu

search on these problems. These results provide evidence for the alternate hypothesis,

$$H_1 : TOS_1 \neq TOS_2 \neq TOS_3.$$

Table 8-15, Best TOS performance across problem type by algorithm

TOS - Best									
MMR	AWU			AWU			AWU		
	Low	Low	Low	Med	Med	Med	High	High	High
	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Tie	Tie	Tie	Loucks	Tie	Tie	Tie	Loucks
Low	Tie	Sim_Ann	Sim_A	Tie	Tie	Tie	Tie	Tie	Loucks
Med	Tie	Sim_Ann	Sim_A	Tie	Tie	Tie	Tie	Tie	Sim_A
High	Tie	Sim_Ann	Sim_A	Tie	Tie	Tie	Tie	Tie	Sim_A

The best performances across problem types by algorithm is presented in table 8-15. These results may be summarized as follows, simulated annealing performed best on 5 of 27 problem types, Loucks' heuristic on 3 of 27 and ties for 19 of 27. There seems to be no pattern to the best performances.

Table 8-16, Worst TOS performance across problem type by algorithm

TOS - Worst									
MMR	AWU			AWU			AWU		
	Low	Low	Low	Med	Med	Med	High	High	High
	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Loucks	Tie	Tie	Loucks	Tie	Tie	Loucks	Tie	Tie
Low	Loucks	Tie	Tie	Loucks	Tie	Tabu_S	Loucks	Tie	Tie
Med	Loucks	Tie	Tie	Loucks	Tie	Tie	Loucks	Tie	Tie
High	Loucks	Tie	Tie	Loucks	Tie	Tie	Loucks	Tie	Tie

As shown in table 8-16, worst performances on TOS are dominated by Louck's heuristic, with 9 "worsts" of 27 problem types, while tabu search had 1 "worst" of 27 problem types. There were 17 statistical ties.

8.5.1.4. TSD - Sum of squared deviations between workers scheduled and targeted work hours.

The hypothesis proposed to test for differences among the algorithms for the sum of squared deviations between workers scheduled and targeted work hours is:

$$H_0 : TSD_1 = TSD_2 = TSD_3$$

$$H_1 : TSD_1 \neq TSD_2 \neq TSD_3$$

As noted before, no direct tests on main effects are statistically permissible when significant disordinal interactions are present. Nevertheless, indirect evidence will be offered in support of the alternate hypothesis, $H_1 : TSD_1 \neq TSD_2 \neq TSD_3$.

Results of ANOVA on TSD, Transformed. The results of the analysis of variance for the dependent variable TSD show that all main and interaction effects are significant at $p < .001$ except the two interaction effects, AWU*PCTNTN and AWU*PCTNTN*ALGORITHM. Examination of the mean-square for each of the effects show that ALGORITHM and PCTNTN and the interaction ALGORITHM*PCTNTN are the dominant sources of variation. All mean-squares for main and interaction effects are larger than the mean-square for error. The multiple r correlation coefficient for this analysis of variance is 0.846 (squared value = 0.715), indicating that the main and interaction effects account for most of the variation of the dependent variable TSD.

Table 8-17. ANOVA on TSD, Transformed

ANOVA on TSD, Transformed					
LEVELS ENCOUNTERED DURING PROCESSING ARE:					
AWU\$:	High	Low	Med		
PCTNTN\$:	High	Low	Med		
MMR\$:	High	Low	Med		
ALGORITHM\$:	Loucks	Sim_Ann	Tabu_Srch		
DEP VAR: TSD	N: 1053	MULTIPLE R: 0.846	SQUARED MULTIPLE R: 0.715		
ANALYSIS OF VARIANCE					
SOURCE	SUM-OF-SQUARES	DF	MEAN-SQUARE	F-RATIO	P
AWU\$	3.353	2	1.676	12.497	0.000
PCTNTN\$	70.965	2	35.482	264.518	0.000
MMR\$	2.526	2	1.263	9.416	0.000
ALGORITHM\$	105.198	2	52.599	392.123	0.000
AWU\$*PCTNTN\$	0.612	4	0.153	1.141	0.336
AWU\$*MMR\$	4.113	4	1.028	7.666	0.000
AWU\$					
*ALGORITHM\$	20.303	4	5.076	37.840	0.000
PCTNTN\$*MMR\$	2.485	4	0.621	4.632	0.001
PCTNTN\$					
*ALGORITHM\$	64.567	4	16.142	120.336	0.000
MMR\$					
*ALGORITHM\$	20.264	4	5.066	37.766	0.000
AWU\$*PCTNTN\$					
*MMR\$	4.550	8	0.569	4.240	0.000
AWU\$*PCTNTN\$					
*ALGORITHM\$	1.199	8	0.150	1.117	0.349
AWU\$*MMR\$					
*ALGORITHM\$	10.064	8	1.258	9.379	0.000
PCTNTN\$*MMR\$					
*ALGORITHM\$	8.206	8	1.026	7.647	0.000
AWU\$*PCTNTN\$					
*MMR\$					
*ALGORITHM\$	8.843	16	0.553	4.120	0.000
ERROR	130.384	972	0.134		
DURBIN-WATSON D STATISTIC	1.905				
FIRST ORDER AUTOCORRELATION	.047				

TSD by Algorithm vs. Problem Type. Figure 8-22 plots that total squared deviations between all worker's desired number of target hours and the number of hours scheduled for the workers by the algorithms, by problem type.

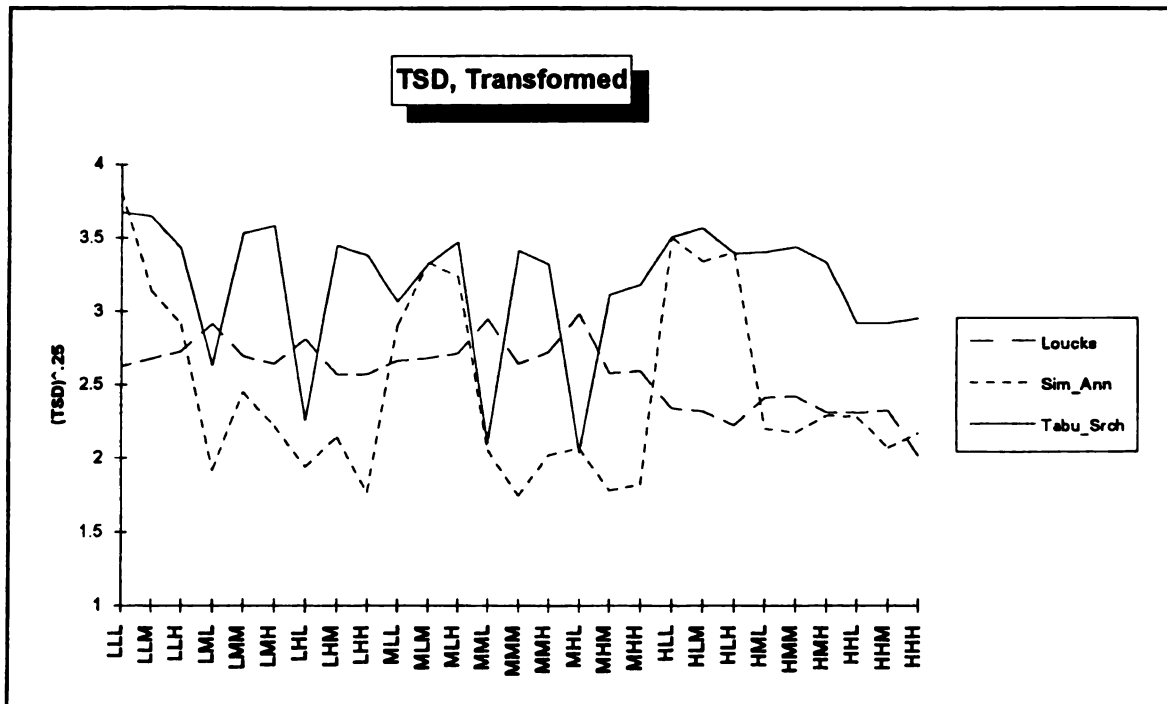


Figure 8-22, Plot of TSD by Algorithm vs. Problem Type

Review of figure 8-22 indicates that there may be a significant difference between the performance of the pair, simulated annealing and tabu search, and Loucks' heuristic, when the level of PCTNTN is low. A Tukey HSD test was performed on the TSD data (a 3x3x3x3, all factor, all factor level test). Results of the test show that differences are significant for problem types LLL, LLM, MLL, MLM, MLH, HLL, HLM, and HLH. For these problems Loucks' heuristic outperformed both simulated annealing and tabu search.

There are other significant differences between algorithms, among them being significant differences between simulated annealing and Loucks on problem types LHH,

LHL, LHM, LML, MHH, and MMH, where simulated annealing outperforms Loucks' heuristic.

Further examination of the plots show that Loucks' heuristic "spikes up" and tabu search "spikes down" when the level of MMR is low. In addition, simulated annealing is highest when the level of PCTNTN is low and lowest when PCTNTN is high.

All these significant differences provide evidence for the alternate hypothesis,

$H_1 : TSD_1 \neq TSD_2 \neq TSD_3$. That is, there are significant differences between the performance of the algorithms on the squared differences between a worker's scheduled and targeted hours.

Table 8-18 Best TSD performance across problem type by algorithm

TSD - Best									
AWU			AWU			AWU			
Low			Med			High			
MMR	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Loucks	Sim_Ann	Tie	Tie	Tie	Tie	Loucks	Tie	Tie
	Loucks	Tie	Tie	Loucks	Sim_A	Sim_A	Loucks	Tie	Tie
Low									
Med									
High									
	Tie	Tie	Sim_A	Loucks	Sim_A	Sim_A	Loucks	Tie	Tie

For TSD, an examination of table 8-18 shows that Loucks' heuristic performed best on 7 of 27 problem types. Simulated annealing had the best performance on TSD for 6 of 27 problem types and tabu search for none of 27 problem types.

Table 8-19, Worst TSD performance across problem type by algorithm

TSD - Worst									
AWU			AWU			AWU			
Low	Low	Low	Med	Med	Med	High	High	High	
MMR	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Tie	Loucks	Tie	Loucks	Loucks	Tie	Tabu_S	Tabu_S
	Tie	Tabu_S	Tabu_S	Tie	Tabu_S	Tie	Tie	Tabu_S	Tabu_S
High	Tie	Tabu_S	Tabu_S	Tie	Tabu_S	Tie	Tie	Tabu_S	Tabu_S

Table 8-19 shows which algorithm had the worst performance by problem type. Tabu search had the largest number of worst performances for 12 of 27 problem types, Loucks' heuristic was the worst for 3 of 27, and there were ties for the worst for 12 of 27 problem types.

8.5.1.5. OBJ - Objective Function Value

The hypothesis as proposed to test for differences among the algorithms with respect to the objective function value is:

$$H_0 : OBJ_1 = OBJ_2 = OBJ_3$$

$$H_1 : OBJ_1 \neq OBJ_2 \neq OBJ_3$$

As before no direct confirmation or rejection of hypotheses can be made in the presence of significant interactions. The data do provide evidence in support of the alternate hypothesis, $H_1 : OBJ_1 \neq OBJ_2 \neq OBJ_3$.

Results of ANOVA on OBJ, Transformed. The analysis of variance results for the dependent variable OBJ show that the main effects AWU, PCTNTN and ALGORITHM are significant at $p < .001$, and MMR is significant at $p < .002$. All interaction effects are

significant at $p < .001$ except for $AWU*PCTNTN$, $AWU*PCTNTN*ALGORITHM$ and $PCTNTN*MMR*ALGORITHM$ which are significant at $p < .05$ and $PCTNTN*MMR$ which is not significant. An examination of the mean-squares for all effects indicates that $ALGORITHM$ is 4 times larger than any other effect, and that all main and interaction effects are larger than the error mean-square. The multiple r correlation for this analysis of variance is 0.755 (squared value = .570) indicating, along with the small error mean-square that the main and interaction effects account for most of the variation in OBJ .

OBJ by Algorithm vs. Problem Type. This figure, 8-23, shows the objective function performance of the algorithms by problem type.

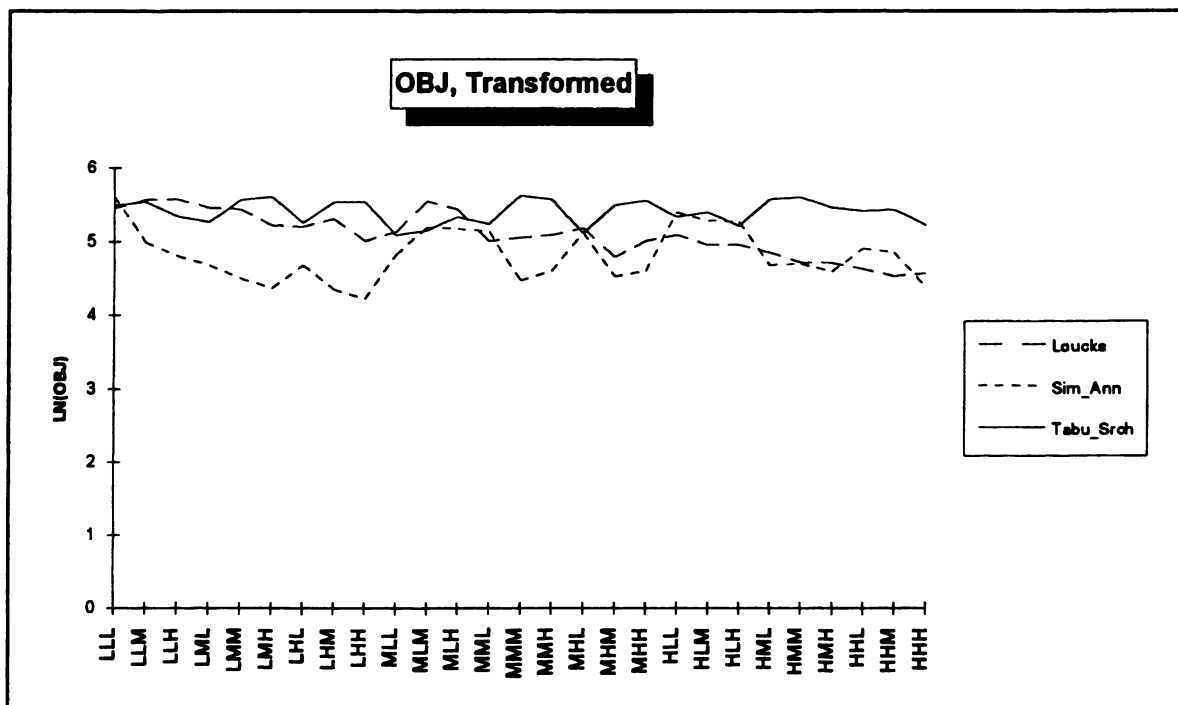


Figure 8-23, Plot of OBJ by Algorithm vs. Problem Type

An examination of figure 8-23 reveals that the only discernible pattern is that the performance of all three algorithms performance is most similar at LLL, MLL, and HLL. Simulated annealing seems to perform worse than other problems when the level of

MMR is low, this occurs 7 of 9 times. Another notable feature is that there appear to be significant differences between the performance of algorithms at several points. A Tukey HSD multiple comparison test (3x3x3x3, all factor, all factor level) was performed on the OBJ data. Simulated annealing outperformed Loucks' heuristic and tabu search on problem types LHH, LHM, LMH, and LMM. Simulated annealing and Loucks' heuristic outperformed tabu search on problem types HMH, HML, and HMM.

These significant differences provide evidence in support of the alternate hypothesis, $H_1 : OBJ1 \neq OBJ2 \neq OBJ3$. That is, there is significant difference among the algorithms for the dependent performance measure OBJ on some problem types.

Table 8-20, Best OBJ performance across problem type by algorithm

OBJ - Best									
AWU			AWU			AWU			
Low			Med	Med	Med	High			
MMR	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Sim_A	Sim_A	Tie	Tie	Tie	Loucks	Tie	Tie
	Sim_A	Sim_A	Sim_A	Tie	Sim_A	Tie	Loucks	Tie	Tie
High	Sim_A	Sim_A	Sim_A	Tie	Sim_A	Tie	Loucks	Tie	Tie

Table 8-20 shows that simulated annealing has the best performance of all three algorithms on OBJ for 10 of 27 problem types, Loucks' heuristic for 3 of 27, and ties for 14 of 27.

Table 8-21, Worst OBJ performance across problem type by algorithm

OBJ - Worst									
MMR	AWU			AWU			AWU		
	Low	Low	Low	Med	Med	Med	High	High	High
	PCTNTN			PCTNTN			PCTNTN		
	Low	Med	High	Low	Med	High	Low	Med	High
	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Tabu_S	Tabu_S
	Tie	Tie	Tie	Tie	Tabu_S	Tabu_S	Tie	Tabu_S	Tabu_S
	Tie	Tie	Tabu_S	Tie	Tabu_S	Tabu_S	Tie	Tabu_S	Tabu_S

In table 8-21 it can be seen that tabu search has the worst performance for 11 of 27 problem types, the rest were ties. No patterns are apparent in the worst performances.

8.6. DIFFERENTIAL DIFFICULTY OF PROBLEM TYPES

The presence of significant interactions in the data presented, restricts what might be said, on a statistically sound basis, about the differential difficulty of the problems, and problem types used in this research. Further, these interactions have obscured any indications that one or another problem type might be more difficult to solve than another. Evidence for this statement is that the measures of problem difficulty, AWU, PCTNTN, and MMR, all have significant disordinal interactions with ALGORITHM. The nature of these disordinal interactions have been explored above and will not be repeated here. This research provides no evidence in support of general conclusions about problem difficulty, with the problems and algorithms used here. Other research, see Jacobs and Bechtol [1993] for example, have found that some problems are more difficult to solve than others, for a single algorithm. This research would support the conclusion that some problems are more difficult to solve for one algorithm or another, but not when all three algorithms are considered jointly.

CHAPTER 9 - SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

9.1. SUMMARY

Determining the work tours for workers with limited interchangeability is a task faced regularly, usually weekly, by managers in thousands of service businesses. Prior research of this problem is sparse. Only Loucks has investigated the problem considering both dissimilarity of workers' available hours and task qualifications. This heterogeneity of workforce characteristics is the key aspect of Loucks' research, and of the research presented here.

The objectives of this study were to apply and implement two new algorithms, simulated annealing and tabu search, to the work tour scheduling problem with limited worker interchangeability, and to compare these two new algorithms to Loucks' heuristic. Any of these methods might be used, on a computer, by a manager in the execution of the tour of duty scheduling task, on a regular basis. All would be an improvement on the manual, trial and error methods used in most service firms.

The data presented provide evidence in support of alternate hypotheses for all performance measures. Stated another way, evidence was presented to support the statement that there are significant differences among all three algorithms on all five performance measures. The data further indicate that the algorithms perform very

similarly with regard to cost, and very differently with regard to time-to-completion. For the measures of solution quality, TOS, TDS and OBJ, the differences weren't clear cut, over all problem types each algorithm was best at least once, and worst at least once.

On time-to-completion, the performance of the algorithms suggest that Loucks' heuristic is better than simulated annealing and simulated annealing is better than tabu search. On the other measures, simulated annealing might be preferred, having 21 of 34 "bests" and no "worsts". On the same measures, Loucks' heuristic had 13 of 34 "bests" and 12 of 36 "worsts" and tabu search had no "bests" and 24 of 36 "worsts".

From a practical standpoint, a manager might prefer simulated annealing over the other solution methods, based on the performance on the quality of solution performance measures. Although simulated annealing was not the fastest solution method in time-to-solution, its solution time was acceptably short, about 22 minutes on average.

Prior research concluded that tabu search outperformed simulated annealing, for example see Knox [1990]. This research does not support that conclusion. Not only was simulated annealing faster in time-to-completion, but, as shown above, outperformed tabu search in measures of solution quality and cost.

9.2. CONTRIBUTIONS

There are several contributions that this research offers the field of employee scheduling.

- Comparative results of three algorithms on five performance measures for the work tour scheduling problem, as detailed above.

- Results contrary to previous research for the relative performance of simulated annealing vs. tabu search, i.e. that simulated annealing generally outperformed tabu search both on time-to-completion, cost, and quality of solution.
 - Development of a method for generating initial, feasible solutions for the work tour scheduling problem.
 - Conception and development of transition generators for simulated annealing and tabu search.
 - Origination and development of the idea of differentiating between "good" and "random" transition generators for simulated annealing.
 - Evidence to show that the combination of "good" and "random" transition generators for simulated annealing is more effective than either "good" or "random" transition generators alone.
 - The first use of the construction - improvement paradigm for simulated annealing and tabu search. Published research, to date, has applied tabu search and simulated annealing in an environment where an initial solution can easily be generated with the implemented algorithm or where the generation of an initial can be done randomly.
 - The first use of simulated annealing and tabu search on a highly constrained problem. Published research, to date, has applied tabu search and simulated annealing in an environment where feasibility is easily maintained or guaranteed.
- Designing transition generators that don't create constraint violations for the work tour scheduling problem is a non-trivial process.

9.3. OPPORTUNITIES FOR FURTHER RESEARCH

Recently, a number of comparative studies of existing algorithms for employee scheduling have been performed, see Bechtold, Brusco and Showalter [1991] and Li, Robinson and Mabert [1991]. Each study used test problems generated with unique problem generators, including this study. While these methods for generation of test problems have merit, use of a standard technique for generation of test problems, or a standardized set of test problems would make results comparable across studies, and save researchers time and expense, while guaranteeing that the algorithm under investigation gets a fair and vigorous (as the case may be) test. The use of a standardized problem set, or a standard method for generating such problems, would also give the consumers of such research more confidence in the results of research into new methods of problem solution. Several other areas, such as the traveling salesman problem, have standard test problems, with varying degrees of difficulty, that researchers use to test their algorithms.

The performance of both simulated annealing and Tabu search rely in large part on the quality of the transition generators. In this research, effort was expended to ensure as much as possible that neither algorithm, simulated annealing and tabu search, had an advantage based on the set of transition generators used in the research. This was done by using the same transition generators for each algorithm, with algorithm specific modifications made during implementation. Further investigation into transition generators probably would yield useful improvements in the performance of both algorithms.

During each iteration of tabu search, every possible transition is generated using the transition generators applied in the particular implementation. The time-to-completion performance of tabu search could be improved by developing a method of reusing previously generated solutions that would not have been affected by use of a

transition to improve the current solution. If such a method were found and used, a substantial proportion of the previously generated potential transitions could be reused, saving computation time.

Prior research into simulated annealing has not established a set of guidelines on when a particular type of cooling schedule might be useful. Most research has used what van Laarhoven and Aarts [1987] call "conceptually simple" cooling schedules. This research compared such conceptually simple cooling schedules with more complex cooling schedules, and found that at least one of the more complex schedules yielded better results, and, therefore, was used in the remainder of the research. While several cooling schedules were investigated during the course of this research, there are many types and permutations of cooling schedules that have not been investigated. Further research into cooling schedules for simulated annealing probably would find a cooling schedule that would converge to a better solution more quickly.

Practical testing of these algorithms in service firms is a reasonable next step in continuing this research. Since the objective function in tabu search and simulated annealing is easily changed to accommodate the needs of different firms, they would seem to be prime candidates for such real-world testing.

APPENDICES

APPENDIX 1.**CALCULATION OF THEORETICAL NUMBER OF WORK TOURS ADAPTED FROM LOUCKS
[1987].***NOTATION*

D = Number of Operation Days per Week.

D_{min} = Allowed Minimum Number of Workdays per Week

D_{max} = Allowed Maximum Number of Workdays per Week

H = Number of Operating Hours per Day

H_{min} = Allowed Minimum Number of Work Hours per Day

H_{max} = Allowed Maximum number of Work Hours per Day

ASSUMPTIONS

A worker's work hours within a day must be contiguous, a worker's work days within a week need not be contiguous, and the worker's number of hours per day need not be constant.

EQUATION

$$\sum_{N=D_{\min}}^{D_{\max}} \left[\left\{ \sum_{M=H_{\min}}^{H_{\max}} (H-M+1) \right\}^N \frac{D!}{(N! (D-N)!)} \right]$$

EXAMPLE

D = 7 Days H = 18 Hours
 D_{min} = 1 Day H_{min} = 3 Hours
 D_{max} = 5 Days H_{max} = 8 Hours

Number Of Workdays (N)	Number Of Tours In Subset
1	567
2	137,781
3	18,600,435
4	1,506,635,235
5	73,222,472,421
Total Tours	74,747,846,439

APPENDIX 2

TRANSFORMATION OF PROBLEM FORMULATION INTO SINGLE FUNCTION FORM

Transformation of the integer program formulation into single function form is necessary for the application of the tabu search and simulated annealing. In this appendix, that transformation will be accomplished.

In a tabu search and simulated annealing there is no way to accommodate constraints directly. Since there are constraints in the problem formulation, they need to be incorporated into the objective function. This is done by using the penalty method. In a penalty method, a constrained problem in optimization is transformed to an unconstrained problem by associating a cost or penalty with each violation of a constraint.

Suppose for example we have a constrained problem such as :

minimize $g(x)$
 subject to $h_i(x) \geq 0, i = 1, 2, \dots, n$
 where x is an m vector.

We transform this to the unconstrained form:

minimize $g(x) + \rho \sum_i \Phi[h_i(x)]$

where Φ = penalty function

ρ = penalty coefficient.

Several alternatives exist for the penalty function Φ . A common penalty function is to square the violation of the constraint, this is the function that we will use. ρ values are generally proportioned so that moderate violations of the constraint yield a penalty that is a significant percentage of the nominal operating cost. The ρ values will be assigned after field experience.

When we apply the method outlined above, the following objective function is generated:

$$\begin{aligned}
 F_f = & W_1 * \sum_{jkl} e_{jkl} + W_2 * (\sum_{ijkl} x_{ijkl} - h_i) \\
 & + \rho_1 * (r_{jkl} + e_{jkl} - \sum_{ijkl} x_{ijkl})^2 \\
 & + \rho_2 * (1 - \sum_{ijkl} x_{ijkl})^2 \\
 & + \rho_3 * (s_{\min} - \sum_{jk} x_{ijkl})^2 \\
 & + \rho_4 * (|U_{il}| * y_{il} - \sum_{jk} x_{ijkl})^2 \\
 & + \rho_5 * (\sum_{jk} x_{ijkl} - s_{\max})^2 \\
 & + \rho_6 * (\sum_l y_{il} - w_{\max})^2 \\
 & + \rho_7 * (\sum_j x_{ijkl} - x_{ij(k+1)l} + x_{ij(k+2)l})^2
 \end{aligned}$$

It will be noted that the ρ_1, ρ_2 , etc. subscripts correspond to and indicate the constraints as listed in the formulation in section 3. Since the most of the constraints are inequality constraints, the value of the term to be squared determines whether it gets included as a penalty, i.e. if the summation takes on any positive value then the constraint is violated, so the term gets squared and included as a penalty, otherwise (the term's value less than or equal to 0) it is ignored.

APPENDIX 3

SYNTHETIC PROBLEM GENERATOR

A computer program, due in large part to Loucks [1987], will be used to generate test problems in which experimental factors can be controlled and for which optimal solutions are known.

The first step in the program is to read in several general problem parameters; Number of Workers (I), Number of Tasks (J), Number of Operating Hours (K), Number of Days (L), Allowed Minimum ($s_{\min}$) and Maximum ($s_{\max}$) Shift Length, and the Allowed Maximum Number of WorkDays ($w_{\max}$).

The next step in the procedure is to decide for each worker i , how many tasks $|T_i|$ and which specific tasks T_i the worker is qualified to perform. $|T_i|$ is chosen by sampling from a discrete probability distribution and T_i is chosen by sampling from a uniform distribution.

Then for each worker i , $|W_i|$, the number of work days, is chosen from a discrete probability distribution. The workers specific workdays, W_i , are chosen by sampling from a uniform distribution.

The worker's first assigned hour b_{i1} is chosen by sampling from a discrete probability distribution. Then the length of worker i 's assigned shift for workday 1, s_{i1} , is decided by sampling from another discrete probability distribution.

Using the same worker i and workday l task j is chosen from T_j for each hour k from b_{ijl} to c_{ijl} by sampling from a uniform distribution. The steps outlined above are then repeated for each day and each worker to constitute a schedule.

At this point, decisions regarding a specific worker's availability must be addressed. These three decisions are what days, how many hours on those days, and the first hour the worker is available. These decisions are made by looking at the schedule already developed and setting availabilities equivalent to the schedule extant. Additional availability is determined by sampling from a uniform distribution for days not already scheduled, hours not already scheduled and hours earlier than those scheduled.

Finally, calculation of the staffing requirements, r_{ijk} , for each task j in each hour k of each day l , and of the workers targeted hour h_i is made. The staffing requirements are set equal to the number of workers scheduled to the task-hour. The number of targeted tour hours for each worker is set to the number of hours scheduled.

The three solution methods would receive as input parameters, the r_{ijk} , h_i , and the several parameters input to the problem generator.

Table A4-1. Raw Data, Continued.

APPENDIX 4**RAW DATA**

The raw data from runs of the algorithms is as follows in table A4-1.

Table A4-1. Raw Data

AWUS\$	PCTNTN\$	MMR\$	ALGORITHMS\$	TOS	TSD	OBJ	COST	TIME
High	High	High	Loucks	91	25	116	6237	52.66992
High	High	High	Loucks	58	12	70	4264.25	27.73828
High	High	High	Loucks	88	29	117	6575.25	54.15039
High	High	High	Loucks	52	26	78	6179	42.07031
High	High	High	Loucks	75	23	98	4736	33.55859
High	High	High	Loucks	57	24	81	5781.25	36.62891
High	High	High	Loucks	77	13	90	3506.25	20.04883
High	High	High	Loucks	142	10	152	3422.5	24.92969
High	High	High	Loucks	73	12	85	3996	24.21875
High	High	High	Loucks	91	17	108	4495.5	27.58008
High	High	High	Loucks	115	13	128	4930.25	35.15039
High	High	High	Loucks	82	12	94	3646.5	27.52148
High	High	High	Loucks	111	18	129	6354.75	49.88086
High	High	High	Loucks	72	10	82	3795	26.4707
High	High	High	Loucks	28	18	46	4141.5	29.21875
High	High	High	Sim_Ann	80	41	121	5808	870.9895
High	High	High	Sim_Ann	67	30	97	4042	1136.027
High	High	High	Sim_Ann	78	37	115	6130	1194.244
High	High	High	Sim_Ann	76	18	94	5809	952.1747
High	High	High	Sim_Ann	39	11	50	4422	1080.146
High	High	High	Sim_Ann	67	17	84	5421	942.2617
High	High	High	Sim_Ann	41	10	51	3399	1074.136
High	High	High	Sim_Ann	34	21	55	3376	1271.589

Table A4-1. Raw Data, Continued.

High	High	High	Sim_Ann	47	40	87	3867	1141.091
High	High	High	Sim_Ann	36	14	50	4246	1657.115
High	High	High	Sim_Ann	62	13	75	4690	1222.279
High	High	High	Sim_Ann	72	18	90	3531	1497.032
High	High	High	Sim_Ann	75	42	117	6022	1080.312
High	High	High	Sim_Ann	84	44	128	3762	918.1513
High	High	High	Sim_Ann	64	17	81	4051	1491.598
High	High	High	Tabu_Srch	126	31	157	6204	6585.908
High	High	High	Tabu_Srch	113	134	247	4541.75	4460.722
High	High	High	Tabu_Srch	116	19	135	6484.5	8563.681
High	High	High	Tabu_Srch	115	35	150	6225.25	6221.88
High	High	High	Tabu_Srch	88	74	162	4948.75	5711.454
High	High	High	Tabu_Srch	103	33	136	5818.25	8010.408
High	High	High	Tabu_Srch	85	104	189	3852.75	5596.472
High	High	High	Tabu_Srch	85	208	293	3894.25	2776.129
High	High	High	Tabu_Srch	86	101	187	4347.5	3048.144
High	High	High	Tabu_Srch	90	94	184	4800.75	3567.977
High	High	High	Tabu_Srch	103	116	219	5050.5	7707.466
High	High	High	Tabu_Srch	122	136	258	3960	2735.551
High	High	High	Tabu_Srch	114	47	161	6428.75	6650.193
High	High	High	Tabu_Srch	107	71	178	3984.75	5354.23
High	High	High	Tabu_Srch	93	88	181	4331.25	5511.541
High	High	Low	Loucks	128	17	145	6327	54.54004
High	High	Low	Loucks	23	39	62	7353.75	51.95996
High	High	Low	Loucks	33	20	53	6662.5	45.86035
High	High	Low	Loucks	84	32	116	7187.25	61.18945
High	High	Low	Loucks	87	37	124	5948.25	45.37012
High	High	Low	Loucks	90	22	112	7390.75	50.42969
High	High	Low	Loucks	62	35	97	7045.5	58.22949
High	High	Low	Loucks	32	28	60	4669.5	32.02051
High	High	Low	Loucks	102	25	127	8722.75	75.52051
High	High	Low	Loucks	70	34	104	7078.5	51.58008
High	High	Low	Loucks	151	23	174	5975.5	63.26953
High	High	Low	Loucks	59	30	89	6410.25	41.46973
High	High	Low	Loucks	107	36	143	6195.75	54.93066
High	High	Low	Loucks	77	20	97	7210.5	54.27051
High	High	Low	Loucks	108	29	137	6839.25	52.73047
High	High	Low	Sim_Ann	131	37	168	6013	851.1406
High	High	Low	Sim_Ann	90	38	128	6762	1051.93
High	High	Low	Sim_Ann	80	4	84	6468	1297
High	High	Low	Sim_Ann	89	58	147	6651	1437.078
High	High	Low	Sim_Ann	103	25	128	5602	952.4062
High	High	Low	Sim_Ann	138	33	171	7104	811.4726
High	High	Low	Sim_Ann	95	46	141	6575	1043.68
High	High	Low	Sim_Ann	59	10	69	4389	1089.508
High	High	Low	Sim_Ann	160	40	200	8408	832.7734

Table A4-1. Raw Data, Continued.

High	High	Low	Sim_Ann	98	35	133	6551	1085.328
High	High	Low	Sim_Ann	93	18	111	5661	917.461
High	High	Low	Sim_Ann	75	11	86	6105	684.4594
High	High	Low	Sim_Ann	106	15	121	5841	1060.281
High	High	Low	Sim_Ann	137	21	158	6938	1268.766
High	High	Low	Sim_Ann	110	45	155	6311	1038.438
High	High	Low	Tabu_Srch	163	59	222	6280.75	5279.299
High	High	Low	Tabu_Srch	137	111	248	7215	9323.22
High	High	Low	Tabu_Srch	140	186	326	7154.5	4141.211
High	High	Low	Tabu_Srch	146	53	199	7224.25	5043.595
High	High	Low	Tabu_Srch	141	83	224	5907	8455.643
High	High	Low	Tabu_Srch	178	51	229	7474	5767.549
High	High	Low	Tabu_Srch	139	116	255	6963	9455.264
High	High	Low	Tabu_Srch	93	76	169	4686	7685.332
High	High	Low	Tabu_Srch	194	94	288	8704.25	7929.292
High	High	Low	Tabu_Srch	156	257	413	7012.5	5741.82
High	High	Low	Tabu_Srch	129	76	205	6012.5	8303.015
High	High	Low	Tabu_Srch	127	89	216	6660	4149.88
High	High	Low	Tabu_Srch	137	38	175	6105	8883.984
High	High	Low	Tabu_Srch	165	29	194	7177.5	6987.894
High	High	Low	Tabu_Srch	161	22	183	6748.5	9041.403
High	High	Med	Loucks	105	22	127	6391.75	48.56055
High	High	Med	Loucks	43	40	83	7390.75	61.50977
High	High	Med	Loucks	27	21	48	6703.5	41.29883
High	High	Med	Loucks	48	32	80	7298.25	79.41992
High	High	Med	Loucks	118	26	144	5923.5	51.52148
High	High	Med	Loucks	62	26	88	7501.75	57.60938
High	High	Med	Loucks	78	31	109	7062	58.05078
High	High	Med	Loucks	18	28	46	4686	31.20117
High	High	Med	Loucks	66	28	94	8833.75	79.9707
High	High	Med	Loucks	60	33	93	7078.5	55.46875
High	High	Med	Loucks	119	28	147	6031	51.95898
High	High	Med	Loucks	63	32	95	6484.25	44.43945
High	High	Med	Loucks	45	36	81	6245.25	49.44141
High	High	Med	Loucks	67	32	99	7359	55.31055
High	High	Med	Loucks	96	27	123	6839.25	55.08984
High	High	Med	Sim_Ann	139	39	178	6050	1368.738
High	High	Med	Sim_Ann	101	21	122	6845	1666.99
High	High	Med	Sim_Ann	74	1	75	6478	1094.17
High	High	Med	Sim_Ann	95	16	111	6854	1190.841
High	High	Med	Sim_Ann	89	22	111	5561	1237.52
High	High	Med	Sim_Ann	141	15	156	7160	1463.881
High	High	Med	Sim_Ann	86	37	123	6584	1461.34
High	High	Med	Sim_Ann	99	30	129	4422	1783.32
High	High	Med	Sim_Ann	124	46	170	8186	1350.07
High	High	Med	Sim_Ann	124	19	143	6749	1518.141

Table A4-1. Raw Data, Continued.

High	High	Med	Sim_Ann	88	21	109	5633	1540.77
High	High	Med	Sim_Ann	80	4	84	6151	1152.434
High	High	Med	Sim_Ann	93	16	109	5816	1433.445
High	High	Med	Sim_Ann	133	33	166	6971	1767.952
High	High	Med	Sim_Ann	128	22	150	6485	1471.392
High	High	Med	Tabu_Srch	163	59	222	6271.5	8252.854
High	High	Med	Tabu_Srch	146	94	240	7316.75	5862.253
High	High	Med	Tabu_Srch	139	196	335	7246.75	5933.308
High	High	Med	Tabu_Srch	127	30	157	7205.75	7637.562
High	High	Med	Tabu_Srch	140	85	225	6014.25	8065.934
High	High	Med	Tabu_Srch	190	132	322	7612.75	8357.039
High	High	Med	Tabu_Srch	129	58	187	6921.75	6007.556
High	High	Med	Tabu_Srch	126	73	199	4669.5	4475.663
High	High	Med	Tabu_Srch	184	70	254	8713.5	7101.482
High	High	Med	Tabu_Srch	165	202	367	7086.75	6491.224
High	High	Med	Tabu_Srch	131	96	227	6068	3626.68
High	High	Med	Tabu_Srch	122	64	186	6576.75	5468.371
High	High	Med	Tabu_Srch	142	89	231	6270	4127.216
High	High	Med	Tabu_Srch	153	11	164	7169.25	6120.851
High	High	Med	Tabu_Srch	157	25	182	6765	6385.818
High	Low	High	Loucks	184	29	213	4142.25	41.13965
High	Low	High	Loucks	104	25	129	3874.5	36.85059
High	Low	High	Loucks	91	23	114	3433.5	29.77051
High	Low	High	Loucks	103	41	144	4812.5	38.1709
High	Low	High	Loucks	65	24	89	4281.25	35.04004
High	Low	High	Loucks	154	24	178	3769.5	29.21973
High	Low	High	Loucks	110	23	133	3979.5	38.87988
High	Low	High	Loucks	104	21	125	3549	39.65039
High	Low	High	Loucks	109	25	134	3916.5	41.46973
High	Low	High	Loucks	145	30	175	4163.25	43.71973
High	Low	High	Loucks	197	22	219	3412.5	54.26953
High	Low	High	Loucks	52	36	88	3869.25	21.86035
High	Low	High	Loucks	126	21	147	3501.75	28.71973
High	Low	High	Loucks	142	18	160	3402	25.0498
High	Low	High	Sim_Ann	56	167	223	4279	1679.924
High	Low	High	Sim_Ann	45	159	204	3948	1852.921
High	Low	High	Sim_Ann	49	173	222	3570	1876.672
High	Low	High	Sim_Ann	84	32	116	4594	1353.788
High	Low	High	Sim_Ann	67	51	118	4275	1604.355
High	Low	High	Sim_Ann	42	130	172	3612	1724.24
High	Low	High	Sim_Ann	49	168	217	3864	1728.107
High	Low	High	Sim_Ann	45	147	192	4095	2591.735
High	Low	High	Sim_Ann	63	144	207	3659	1605.018
High	Low	High	Sim_Ann	78	203	281	4053	2301.054
High	Low	High	Sim_Ann	77	433	510	4389	3225.4
High	Low	High	Sim_Ann	57	242	299	3539	1850.497

Table A4-1. Raw Data, Continued.

High	Low	High	Sim_Ann	35	150	185	3848	1605.744
High	Low	High	Sim_Ann	44	87	131	3575	1303.976
High	Low	High	Sim_Ann	45	81	126	3439	1704.242
High	Low	High	Tabu_Srch	48	139	187	4247.25	7049.514
High	Low	High	Tabu_Srch	45	159	204	3942.75	7550.317
High	Low	High	Tabu_Srch	35	115	150	3512.25	9792.619
High	Low	High	Tabu_Srch	119	119	238	4850	7469.008
High	Low	High	Tabu_Srch	115	247	362	4625	6355.903
High	Low	High	Tabu_Srch	37	89	126	3591	4862.726
High	Low	High	Tabu_Srch	33	142	175	3780	7528.504
High	Low	High	Tabu_Srch	25	65	90	4032	9206.476
High	Low	High	Tabu_Srch	49	130	179	3601.5	9594.32
High	Low	High	Tabu_Srch	50	97	147	3906	7845.326
High	Low	High	Tabu_Srch	46	196	242	4231.5	8563.857
High	Low	High	Tabu_Srch	60	179	239	3554.25	5538.024
High	Low	High	Tabu_Srch	30	109	139	3827.25	10025.25
High	Low	High	Tabu_Srch	49	136	185	3601.5	10067.6
High	Low	High	Tabu_Srch	40	88	128	3412.5	9181.166
High	Low	Low	Loucks	132	19	151	3979.5	76.00977
High	Low	Low	Loucks	162	44	206	3787.5	63.49023
High	Low	Low	Loucks	107	36	143	4150	46.12988
High	Low	Low	Loucks	143	27	170	3585.75	31.95996
High	Low	Low	Loucks	234	31	265	4882.5	54.87012
High	Low	Low	Loucks	64	29	93	3701.25	37.57031
High	Low	Low	Loucks	152	35	187	4341.75	58.92969
High	Low	Low	Loucks	137	45	182	4475	47.73047
High	Low	Low	Loucks	78	20	98	3360	24.5
High	Low	Low	Loucks	113	25	138	3942.75	37.24023
High	Low	Low	Loucks	165	37	202	4550	42.73047
High	Low	Low	Loucks	167	30	197	3638.25	41.63086
High	Low	Low	Loucks	128	43	171	4368	44.80957
High	Low	Low	Loucks	140	22	162	3906	36.41992
High	Low	Low	Sim_Ann	76	308	384	4195	2359.82
High	Low	Low	Sim_Ann	96	75	171	3744	1543.281
High	Low	Low	Sim_Ann	74	59	133	4088	1149.164
High	Low	Low	Sim_Ann	55	90	145	3686	1677.211
High	Low	Low	Sim_Ann	78	247	325	5009	1636.944
High	Low	Low	Sim_Ann	49	167	216	3796	1447.719
High	Low	Low	Sim_Ann	51	151	202	4426	1613.501
High	Low	Low	Sim_Ann	80	88	168	4413	1777.16
High	Low	Low	Sim_Ann	53	214	267	3491	1503.414
High	Low	Low	Sim_Ann	44	87	131	4043	2500.98
High	Low	Low	Sim_Ann	100	147	247	4481	1488.922
High	Low	Low	Sim_Ann	55	189	244	3770	2268.296
High	Low	Low	Sim_Ann	65	268	333	4484	1579.536
High	Low	Low	Sim_Ann	48	125	173	4037	1764.32

Table A4-1. Raw Data, Continued.

High	Low	Low	Sim_Ann	83	11	94	5763	2011.916
High	Low	Low	Tabu_Srch	61	199	260	4105.5	10421.29
High	Low	Low	Tabu_Srch	124	213	337	3931.25	7760.36
High	Low	Low	Tabu_Srch	99	146	245	4262.5	10343.34
High	Low	Low	Tabu_Srch	39	66	105	3617.25	6817.057
High	Low	Low	Tabu_Srch	63	200	263	4940.25	8790.045
High	Low	Low	Tabu_Srch	39	113	152	3780	7635.036
High	Low	Low	Tabu_Srch	43	115	158	4399.5	11071.67
High	Low	Low	Tabu_Srch	114	262	376	4618.75	9586.769
High	Low	Low	Tabu_Srch	29	68	97	3375.75	8850.957
High	Low	Low	Tabu_Srch	37	86	123	4000.5	10687.5
High	Low	Low	Tabu_Srch	139	338	477	4743.75	11169.62
High	Low	Low	Tabu_Srch	36	76	112	3706.5	10281.32
High	Low	Low	Tabu_Srch	45	144	189	4389	11087.76
High	Low	Low	Tabu_Srch	37	108	145	3979.5	10342.25
High	Low	Low	Tabu_Srch	135	345	480	6131.25	10499.45
High	Low	Med	Loucks	85	22	107	4047.75	46.79004
High	Low	Med	Loucks	83	33	116	3881.25	37.18945
High	Low	Med	Loucks	134	36	170	4193.75	41.25
High	Low	Med	Loucks	147	19	166	3596.25	35.96973
High	Low	Med	Loucks	186	31	217	4924.5	52.56055
High	Low	Med	Loucks	70	31	101	3732.75	31.25977
High	Low	Med	Loucks	170	27	197	4352.25	59.10059
High	Low	Med	Loucks	127	48	175	4550	46.95996
High	Low	Med	Loucks	70	19	89	3370.5	24.66016
High	Low	Med	Loucks	117	22	139	3953.25	34.7207
High	Low	Med	Loucks	140	37	177	4581.25	41.74023
High	Low	Med	Loucks	130	25	155	3654	34
High	Low	Med	Loucks	154	40	194	4378.5	50.37012
High	Low	Med	Loucks	80	26	106	3948	30.20996
High	Low	Med	Sim_Ann	73	345	418	4221	4899.347
High	Low	Med	Sim_Ann	94	110	204	3888	4409.528
High	Low	Med	Sim_Ann	60	60	120	4081	2993.818
High	Low	Med	Sim_Ann	49	127	176	3712	4306.466
High	Low	Med	Sim_Ann	76	259	335	5061	3958.614
High	Low	Med	Sim_Ann	48	182	230	3812	1645.015
High	Low	Med	Sim_Ann	49	181	230	4468	1695.454
High	Low	Med	Sim_Ann	86	44	130	4363	1556.766
High	Low	Med	Sim_Ann	51	209	260	3497	1687.77
High	Low	Med	Sim_Ann	46	112	158	4069	1541.541
High	Low	Med	Sim_Ann	96	156	252	4513	1765.36
High	Low	Med	Sim_Ann	51	193	244	3791	1403.4
High	Low	Med	Sim_Ann	57	156	213	4463	1446.859
High	Low	Med	Sim_Ann	46	115	161	4048	1388.879
High	Low	Med	Sim_Ann	74	12	86	5756	1322.522
High	Low	Med	Tabu_Srch	63	285	348	4163.25	11078.23

Table A4-1. Raw Data, Continued.

High	Low	Med	Tabu_Srch	116	266	382	4012.5	6866.375
High	Low	Med	Tabu_Srch	105	191	296	4412.5	7480.438
High	Low	Med	Tabu_Srch	31	81	112	3627.75	8283.397
High	Low	Med	Tabu_Srch	56	241	297	4956	11325.34
High	Low	Med	Tabu_Srch	39	93	132	3801	10134.56
High	Low	Med	Tabu_Srch	40	176	216	4410	10501.86
High	Low	Med	Tabu_Srch	117	285	402	4587.5	10724.9
High	Low	Med	Tabu_Srch	24	72	96	3365.25	9092.756
High	Low	Med	Tabu_Srch	35	91	126	4042.5	10700.86
High	Low	Med	Tabu_Srch	119	243	362	4656.25	8503.333
High	Low	Med	Tabu_Srch	33	63	96	3738	10195.12
High	Low	Med	Tabu_Srch	43	116	159	4394.25	9061.921
High	Low	Med	Tabu_Srch	35	90	125	4005.75	10315.41
High	Low	Med	Tabu_Srch	127	163	290	6125	11819.57
High	Med	High	Loucks	119	26	145	5038.75	44.33008
High	Med	High	Loucks	74	22	96	3045	22.95996
High	Med	High	Loucks	82	23	105	3443.75	30.65039
High	Med	High	Loucks	127	35	162	5611.5	53.87988
High	Med	High	Loucks	171	40	211	5169.25	48.71973
High	Med	High	Loucks	69	39	108	5821.75	48.06055
High	Med	High	Loucks	131	35	166	4516.75	36.58008
High	Med	High	Loucks	85	29	114	4828.5	35.25977
High	Med	High	Loucks	55	41	96	4799.5	32.79004
High	Med	High	Loucks	23	42	65	5858	39.33008
High	Med	High	Loucks	82	20	102	4183.25	30.20996
High	Med	High	Loucks	60	19	79	2834.75	20.65039
High	Med	High	Loucks	52	14	66	3211.75	21.37012
High	Med	High	Loucks	194	42	236	5183.75	47.07031
High	Med	High	Sim_Ann	69	40	109	4865	2276.568
High	Med	High	Sim_Ann	60	25	85	2980	1370.117
High	Med	High	Sim_Ann	64	43	107	3386	1818.139
High	Med	High	Sim_Ann	82	25	107	5365	1433.609
High	Med	High	Sim_Ann	62	15	77	4814	1259.098
High	Med	High	Sim_Ann	98	14	112	5525	1519.795
High	Med	High	Sim_Ann	67	59	126	4343	1869.921
High	Med	High	Sim_Ann	64	22	86	4575	1686.413
High	Med	High	Sim_Ann	68	17	85	4524	1303.401
High	Med	High	Sim_Ann	94	9	103	5575	1446.48
High	Med	High	Sim_Ann	65	44	109	4053	2151.914
High	Med	High	Sim_Ann	51	21	72	2748	2397.154
High	Med	High	Sim_Ann	73	38	111	3125	1846.211
High	Med	High	Sim_Ann	66	15	81	3857	1620.617
High	Med	High	Sim_Ann	76	34	110	4916	1081.051
High	Med	High	Tabu_Srch	125	246	371	5256.25	8622.104
High	Med	High	Tabu_Srch	92	141	233	3342.25	2306.329
High	Med	High	Tabu_Srch	108	197	305	3741	3987.819

Table A4-1. Raw Data, Continued.

High	Med	High	Tabu_Srch	119	78	197	5647.75	8915.841
High	Med	High	Tabu_Srch	104	95	199	5154.75	6051.842
High	Med	High	Tabu_Srch	121	35	156	5713	8225.55
High	Med	High	Tabu_Srch	114	212	326	4734.25	7728.248
High	Med	High	Tabu_Srch	98	84	182	4857.5	8344.492
High	Med	High	Tabu_Srch	108	135	243	4850.25	7839.282
High	Med	High	Tabu_Srch	144	115	259	5981.25	4402.274
High	Med	High	Tabu_Srch	103	180	283	4371.75	6198.898
High	Med	High	Tabu_Srch	73	85	158	2921.75	3127.562
High	Med	High	Tabu_Srch	100	101	201	3342.25	4538.172
High	Med	High	Tabu_Srch	96	119	215	4103.5	7776.352
High	Med	High	Tabu_Srch	128	186	314	5292.5	3691.746
High	Med	Low	Loucks	81	31	112	5459.25	44.37988
High	Med	Low	Loucks	124	39	163	5604.25	54.65039
High	Med	Low	Loucks	140	35	175	5879.75	48.00977
High	Med	Low	Loucks	137	34	171	4908.25	48.71973
High	Med	Low	Loucks	102	28	130	5278	47.24023
High	Med	Low	Loucks	120	27	147	4676.25	45.37012
High	Med	Low	Loucks	58	41	99	5539	43.73047
High	Med	Low	Loucks	89	31	120	4792.25	40.04004
High	Med	Low	Loucks	87	41	128	5655	42.2998
High	Med	Low	Loucks	69	31	100	5082.25	49.60059
High	Med	Low	Loucks	55	37	92	4886.5	38.40039
High	Med	Low	Loucks	111	34	145	5118.5	43.12012
High	Med	Low	Loucks	74	35	109	5814.5	44.92969
High	Med	Low	Loucks	65	47	112	6169.75	44.31934
High	Med	Low	Sim_Ann	96	12	108	5271	1379.929
High	Med	Low	Sim_Ann	102	28	130	5278	2564.102
High	Med	Low	Sim_Ann	109	33	142	5612	1066.021
High	Med	Low	Sim_Ann	55	41	96	4568	2029.533
High	Med	Low	Sim_Ann	80	17	97	4995	1442.489
High	Med	Low	Sim_Ann	81	31	112	4568	1887.909
High	Med	Low	Sim_Ann	83	35	118	5307	1561.881
High	Med	Low	Sim_Ann	74	29	103	4553	1430.566
High	Med	Low	Sim_Ann	82	46	128	5271	1763.677
High	Med	Low	Sim_Ann	79	11	90	4821	1243.98
High	Med	Low	Sim_Ann	97	25	122	6010	1600.959
High	Med	Low	Sim_Ann	48	20	68	4604	1404.496
High	Med	Low	Sim_Ann	72	16	88	4887	1832.316
High	Med	Low	Sim_Ann	110	40	150	5496	1465.438
High	Med	Low	Sim_Ann	103	9	112	5800	1410.733
High	Med	Low	Tabu_Srch	123	95	218	5488.25	9620.726
High	Med	Low	Tabu_Srch	147	167	314	5589.75	9669.143
High	Med	Low	Tabu_Srch	145	93	238	5901.5	9111.43
High	Med	Low	Tabu_Srch	125	283	408	5089.5	6948.048
High	Med	Low	Tabu_Srch	128	149	277	5350.5	9769.343

Table A4-1. Raw Data, Continued.

High	Med	Low	Tabu_Srch	105	119	224	4799.5	9298.901
High	Med	Low	Tabu_Srch	129	271	400	5655	9337.635
High	Med	Low	Tabu_Srch	112	219	331	4843	5824.058
High	Med	Low	Tabu_Srch	140	214	354	5676.75	8790.442
High	Med	Low	Tabu_Srch	133	125	258	5285.25	6662.191
High	Med	Low	Tabu_Srch	146	66	212	6351	6346.297
High	Med	Low	Tabu_Srch	93	97	190	4951.75	7956.849
High	Med	Low	Tabu_Srch	103	67	170	5125.75	9151.667
High	Med	Low	Tabu_Srch	137	69	206	5698.5	10176.05
High	Med	Low	Tabu_Srch	141	167	308	6075.5	5976.499
High	Med	Med	Loucks	41	32	73	5502.75	43
High	Med	Med	Loucks	131	34	165	5611.5	48.05957
High	Med	Med	Loucks	84	40	124	5952.25	45.4209
High	Med	Med	Loucks	82	39	121	4988	43.94043
High	Med	Med	Loucks	102	28	130	5278	47.23047
High	Med	Med	Loucks	94	24	118	4676.25	38.73047
High	Med	Med	Loucks	71	38	109	5560.75	60.68945
High	Med	Med	Loucks	69	30	99	4806.75	37.23926
High	Med	Med	Loucks	85	35	120	5640.5	45.30957
High	Med	Med	Loucks	92	32	124	5118.5	49.70996
High	Med	Med	Loucks	97	54	151	6568.5	52.33984
High	Med	Med	Loucks	60	40	100	4922.75	36.4707
High	Med	Med	Loucks	105	30	135	5147.5	41.7998
High	Med	Med	Loucks	68	38	106	5858	46.09082
High	Med	Med	Loucks	62	45	107	6177	53.94043
High	Med	Med	Sim_Ann	100	9	109	5293	2783.278
High	Med	Med	Sim_Ann	85	55	140	5213	3167.208
High	Med	Med	Sim_Ann	114	31	145	5612	2058.607
High	Med	Med	Sim_Ann	52	26	78	4589.25	2089.365
High	Med	Med	Sim_Ann	68	27	95	4908	2984.844
High	Med	Med	Sim_Ann	81	27	108	4524	3102.753
High	Med	Med	Sim_Ann	83	29	112	5263.5	2814.46
High	Med	Med	Sim_Ann	65	38	103	4487.75	1054.118
High	Med	Med	Sim_Ann	83	44	127	5357.75	1613.114
High	Med	Med	Sim_Ann	94	2	96	4901	2587.141
High	Med	Med	Sim_Ann	107	12	119	6104.5	2830.647
High	Med	Med	Sim_Ann	53	19	72	4654.5	2327.558
High	Med	Med	Sim_Ann	73	23	96	4908.25	1996.978
High	Med	Med	Sim_Ann	123	11	134	5604.25	1101.294
High	Med	Med	Sim_Ann	102	10	112	5821.75	3132.777
High	Med	Med	Tabu_Srch	129	108	237	5502.75	8048.395
High	Med	Med	Tabu_Srch	158	298	456	5756.5	10211.44
High	Med	Med	Tabu_Srch	155	136	291	5974	6743.694
High	Med	Med	Tabu_Srch	116	220	336	5075	7556.386
High	Med	Med	Tabu_Srch	128	149	277	5350.5	9769.5
High	Med	Med	Tabu_Srch	126	170	296	4843	7198.274

Table A4-1. Raw Data, Continued.

High	Med	Med	Tabu_Srch	115	117	232	5553.5	9259.342
High	Med	Med	Tabu_Srch	125	248	373	4995.25	9666.027
High	Med	Med	Tabu_Srch	132	215	347	5763.75	10024.34
High	Med	Med	Tabu_Srch	129	77	206	5183.75	9218.443
High	Med	Med	Tabu_Srch	136	67	203	6314.75	10302.79
High	Med	Med	Tabu_Srch	94	98	192	5009.75	7389.456
High	Med	Med	Tabu_Srch	107	85	192	5147.5	9775.879
High	Med	Med	Tabu_Srch	141	67	208	5756.5	10023.01
High	Med	Med	Tabu_Srch	136	116	252	6068.25	7241.36
Low	High	High	Loucks	219	36	255	3778.5	49.1001
Low	High	High	Loucks	73	34	107	5296.5	43.01025
Low	High	High	Loucks	53	47	100	5984.75	46.56982
Low	High	High	Loucks	141	47	188	3316.5	34.5498
Low	High	High	Loucks	72	32	104	4125	25.42969
Low	High	High	Loucks	58	38	96	5280	102.6104
Low	High	High	Loucks	137	51	188	5032.5	42.24023
Low	High	High	Loucks	85	47	132	5124.5	36.41992
Low	High	High	Loucks	86	39	125	3481.5	30.37012
Low	High	High	Loucks	292	47	339	5057.25	39.93018
Low	High	High	Loucks	43	35	78	4174.5	33.00977
Low	High	High	Loucks	163	48	211	4207.5	31.25977
Low	High	High	Loucks	103	54	157	4083.75	29.77002
Low	High	High	Loucks	107	38	145	3456.75	29.72021
Low	High	High	Loucks	91	51	142	6080.25	54.47998
Low	High	High	Sim_Ann	28	24	52	3514.5	1262.175
Low	High	High	Sim_Ann	62	6	68	4983	618.2425
Low	High	High	Sim_Ann	61	1	62	5540.75	739.0205
Low	High	High	Sim_Ann	38	17	55	3019.5	1025.09
Low	High	High	Sim_Ann	42	3	45	3836.25	895.12
Low	High	High	Sim_Ann	84	12	96	4884	761.5492
Low	High	High	Sim_Ann	59	24	83	4578.75	690.6294
Low	High	High	Sim_Ann	58	2	60	4671.25	604.34
Low	High	High	Sim_Ann	36	23	59	3267	588.0055
Low	High	High	Sim_Ann	63	1	64	4661.25	805.6214
Low	High	High	Sim_Ann	51	3	54	3877.5	743.4317
Low	High	High	Sim_Ann	49	26	75	3861	632.9564
Low	High	High	Sim_Ann	47	11	58	3696	716.1358
Low	High	High	Sim_Ann	41	36	77	3242.25	1045.64
Low	High	High	Sim_Ann	91	18	109	5527.5	676.9068
Low	High	High	Tabu_Srch	102	190	292	4191	6806.579
Low	High	High	Tabu_Srch	106	58	164	5403.75	11225.35
Low	High	High	Tabu_Srch	120	102	222	6086.5	12114.73
Low	High	High	Tabu_Srch	111	210	321	3638.25	4286.863
Low	High	High	Tabu_Srch	131	194	325	4603.5	9085.792
Low	High	High	Tabu_Srch	141	73	214	5403.75	9445.887
Low	High	High	Tabu_Srch	137	174	311	5247	9061.55

Table A4-1. Raw Data, Continued.

Low	High	High	Tabu_Srch	116	122	238	5244.75	10897.02
Low	High	High	Tabu_Srch	107	206	313	3918.75	6260.292
Low	High	High	Tabu_Srch	148	174	322	5387.25	12266.45
Low	High	High	Tabu_Srch	106	138	244	4356	5831.373
Low	High	High	Tabu_Srch	140	243	383	4644.75	7158.472
Low	High	High	Tabu_Srch	101	135	236	4207.5	10363.71
Low	High	High	Tabu_Srch	97	188	285	3770.25	9280.053
Low	High	High	Tabu_Srch	140	39	179	5956.5	12449.53
Low	High	Low	Loucks	73	60	133	4941.75	42.57007
Low	High	Low	Loucks	195	66	261	5254	48.61011
Low	High	Low	Loucks	185	67	252	7380	117.27
Low	High	Low	Loucks	174	75	249	6270	56.07007
Low	High	Low	Loucks	133	51	184	3802.75	40.20996
Low	High	Low	Loucks	85	63	148	6252.5	56.67993
Low	High	Low	Loucks	85	42	127	3330	29.27002
Low	High	Low	Loucks	121	77	198	6456.5	86.5
Low	High	Low	Loucks	93	39	132	3034	27.67993
Low	High	Low	Loucks	251	65	316	5254	49.55005
Low	High	Low	Loucks	81	85	166	6971.25	96.16992
Low	High	Low	Loucks	108	68	176	5123.25	52.72998
Low	High	Low	Loucks	177	64	241	4892.25	41.03003
Low	High	Low	Loucks	207	94	301	7647.75	107.4299
Low	High	Low	Loucks	91	70	161	6954.75	105.73
Low	High	Low	Sim_Ann	101	10	111	4364.25	776.708
Low	High	Low	Sim_Ann	92	12	104	4551	568.9242
Low	High	Low	Sim_Ann	164	25	189	6437	649.8895
Low	High	Low	Sim_Ann	136	20	156	5502.75	645.7544
Low	High	Low	Sim_Ann	35	6	41	3239	518.4334
Low	High	Low	Sim_Ann	111	25	136	5412	551.6926
Low	High	Low	Sim_Ann	34	9	43	2876.75	732.98
Low	High	Low	Sim_Ann	148	19	167	5605.5	549.7141
Low	High	Low	Sim_Ann	44	9	53	2682.5	818.6563
Low	High	Low	Sim_Ann	83	16	99	4541.75	565.993
Low	High	Low	Sim_Ann	161	15	176	6146.25	768.1727
Low	High	Low	Sim_Ann	111	5	116	4537.5	644.2188
Low	High	Low	Sim_Ann	78	21	99	4191	821.1563
Low	High	Low	Sim_Ann	265	32	297	6608.25	669.5452
Low	High	Low	Sim_Ann	200	26	226	6179.25	628.918
Low	High	Low	Tabu_Srch	148	63	211	4818	9574.344
Low	High	Low	Tabu_Srch	134	44	178	4976.5	14196.18
Low	High	Low	Tabu_Srch	194	5	199	6734.25	10025.82
Low	High	Low	Tabu_Srch	160	18	178	5717.25	10530.81
Low	High	Low	Tabu_Srch	91	106	197	3843.75	11088.67
Low	High	Low	Tabu_Srch	143	23	166	5729.75	10381.87
Low	High	Low	Tabu_Srch	98	111	209	3468.75	10269.04
Low	High	Low	Tabu_Srch	167	14	181	5836.75	15019.46

Table A4-1. Raw Data, Continued.

Low	High	Low	Tabu_Srch	93	94	187	3182	7797.804
Low	High	Low	Tabu_Srch	108	17	125	4837.75	11825.61
Low	High	Low	Tabu_Srch	172	8	180	6303	15021.73
Low	High	Low	Tabu_Srch	130	18	148	4752	12400.55
Low	High	Low	Tabu_Srch	125	36	161	4603.5	8240.825
Low	High	Low	Tabu_Srch	279	18	297	6707.25	20470.95
Low	High	Low	Tabu_Srch	231	9	240	6435	20707.08
Low	High	Med	Loucks	187	58	245	5222.25	47.01001
Low	High	Med	Loucks	233	52	285	4719	44.92993
Low	High	Med	Loucks	194	44	238	5618.25	49.20996
Low	High	Med	Loucks	163	46	209	5428.5	52.83984
Low	High	Med	Loucks	66	48	114	5255.25	40.42993
Low	High	Med	Loucks	193	33	226	2945.25	36.58008
Low	High	Med	Loucks	139	22	161	1716	25.76001
Low	High	Med	Loucks	260	41	301	3638.25	41.03003
Low	High	Med	Loucks	69	54	123	4661.25	33.38989
Low	High	Med	Loucks	134	41	175	2813.25	37.90015
Low	High	Med	Loucks	318	32	350	4290	54.70996
Low	High	Med	Loucks	63	37	100	3902.25	31.68994
Low	High	Med	Loucks	281	45	326	3968.25	47.18018
Low	High	Med	Loucks	51	40	91	3588.75	26.74976
Low	High	Med	Loucks	108	41	149	4504.5	40.47998
Low	High	Med	Sim_Ann	67	14	81	4760.25	720.7294
Low	High	Med	Sim_Ann	44	3	47	4281.75	953.06
Low	High	Med	Sim_Ann	89	9	98	5197.5	1332.813
Low	High	Med	Sim_Ann	72	15	87	4925.25	942.5352
Low	High	Med	Sim_Ann	50	12	62	4793.25	812.043
Low	High	Med	Sim_Ann	39	41	80	2813.25	1145.645
Low	High	Med	Sim_Ann	25	83	108	1707.75	728.7656
Low	High	Med	Sim_Ann	37	19	56	3357.75	945.8126
Low	High	Med	Sim_Ann	51	3	54	4191	742.7032
Low	High	Med	Sim_Ann	33	50	83	2590.5	686.5468
Low	High	Med	Sim_Ann	58	51	109	4116.75	1027.223
Low	High	Med	Sim_Ann	39	7	46	3638.25	734.4688
Low	High	Med	Sim_Ann	52	97	149	3770.25	2481.734
Low	High	Med	Sim_Ann	30	5	35	3300	1407.609
Low	High	Med	Sim_Ann	51	3	54	4191	1180.266
Low	High	Med	Tabu_Srch	118	161	279	5197.5	9737.797
Low	High	Med	Tabu_Srch	134	213	347	5057.25	7058.071
Low	High	Med	Tabu_Srch	144	84	228	5676	9737.406
Low	High	Med	Tabu_Srch	129	74	203	5445	9508.459
Low	High	Med	Tabu_Srch	129	147	276	5461.5	10870.24
Low	High	Med	Tabu_Srch	89	159	248	3242.25	9921.771
Low	High	Med	Tabu_Srch	45	107	152	1947	7619.292
Low	High	Med	Tabu_Srch	105	197	302	3960	3957.215
Low	High	Med	Tabu_Srch	110	114	224	4702.5	11338.54

Table A4-1. Raw Data, Continued.

Low	High	Med	Tabu_Srch	82	155	237	3102	6129.08
Low	High	Med	Tabu_Srch	102	147	249	4529.25	9745.893
Low	High	Med	Tabu_Srch	111	169	280	4273.5	6335.663
Low	High	Med	Tabu_Srch	93	164	257	4207.5	8612.497
Low	High	Med	Tabu_Srch	89	142	231	3828	10255.3
Low	High	Med	Tabu_Srch	106	124	230	4686	7409.092
Low	Low	High	Loucks	165	41	206	3001.5	48.5
Low	Low	High	Loucks	148	42	190	4581.5	33.73
Low	Low	High	Loucks	125	50	175	3687.5	43.78
Low	Low	High	Loucks	75	59	134	3512.5	33.17
Low	Low	High	Loucks	295	52	347	2468.75	33.61
Low	Low	High	Loucks	198	58	256	3175	37.01999
Low	Low	High	Loucks	161	57	218	3968.75	44.16
Low	Low	High	Loucks	483	40	523	2793	77.28
Low	Low	High	Loucks	175	58	233	3056.25	37.46002
Low	Low	High	Loucks	191	75	266	4425	40.63998
Low	Low	High	Loucks	519	69	588	4131.25	57.99997
Low	Low	High	Loucks	303	70	373	3687.5	46.35004
Low	Low	High	Loucks	179	59	238	2831.25	32.24005
Low	Low	High	Sim_Ann	27	62	89	2849.25	1189.801
Low	Low	High	Sim_Ann	31	20	51	4307	1668.34
Low	Low	High	Sim_Ann	55	29	84	3493.75	2542.29
Low	Low	High	Sim_Ann	52	36	88	3268.75	2662.776
Low	Low	High	Sim_Ann	33	80	113	2331.25	1172.422
Low	Low	High	Sim_Ann	46	54	100	2950	2347.958
Low	Low	High	Sim_Ann	59	64	123	3775	1234.07
Low	Low	High	Sim_Ann	22	94	116	2000.25	1557.25
Low	Low	High	Sim_Ann	70	242	312	3318	994.3298
Low	Low	High	Sim_Ann	33	159	192	2756.25	1082.375
Low	Low	High	Sim_Ann	65	125	190	3012.5	3080.866
Low	Low	High	Sim_Ann	79	107	186	4062.5	3002.002
Low	Low	High	Sim_Ann	62	60	122	3887.5	1145.521
Low	Low	High	Sim_Ann	64	84	148	3450	3424.082
Low	Low	High	Sim_Ann	45	63	108	2631.25	1009.422
Low	Low	High	Tabu_Srch	40	71	111	3008.75	2412.863
Low	Low	High	Tabu_Srch	49	48	97	4502	2762.699
Low	Low	High	Tabu_Srch	93	125	218	3756.25	2355.824
Low	Low	High	Tabu_Srch	87	111	198	3506.25	2630.117
Low	Low	High	Tabu_Srch	71	172	243	2643.75	1612.938
Low	Low	High	Tabu_Srch	83	135	218	3231.25	2737.156
Low	Low	High	Tabu_Srch	121	258	379	4212.5	2564.738
Low	Low	High	Tabu_Srch	24	66	90	2037	1267.516
Low	Low	High	Tabu_Srch	58	128	186	3265.5	2863.637
Low	Low	High	Tabu_Srch	27	91	118	2735.25	1954.31
Low	Low	High	Tabu_Srch	103	229	332	3306.25	1667.656
Low	Low	High	Tabu_Srch	109	141	250	4312.5	1636.696

Table A4-1. Raw Data, Continued.

Low	Low	High	Tabu_Srch	135	379	514	4400	1531.27
Low	Low	High	Tabu_Srch	96	164	260	3668.75	2345.96
Low	Low	High	Tabu_Srch	92	224	316	2962.5	999.8594
Low	Low	Low	Loucks	194	39	233	2604	89.14844
Low	Low	Low	Loucks	154	25	179	1438.5	59.03906
Low	Low	Low	Loucks	178	52	230	2766.75	77.32813
Low	Low	Low	Loucks	141	58	199	2950	34.71094
Low	Low	Low	Loucks	92	65	157	3486	69.26563
Low	Low	Low	Loucks	331	51	382	3055.5	183.5625
Low	Low	Low	Loucks	283	56	339	2518.75	51.57031
Low	Low	Low	Loucks	181	32	213	1727.25	83.32031
Low	Low	Low	Loucks	164	63	227	2803.5	74.58594
Low	Low	Low	Loucks	137	49	186	3218.25	83.16406
Low	Low	Low	Loucks	250	67	317	2850	51.46094
Low	Low	Low	Loucks	209	40	249	2609.25	102.6016
Low	Low	Low	Loucks	126	27	153	1449	58.99219
Low	Low	Low	Loucks	290	53	343	2772	95.52344
Low	Low	Low	Sim_Ann	67	207	274	2751	728
Low	Low	Low	Sim_Ann	51	179	230	1575	638.2579
Low	Low	Low	Sim_Ann	83	399	482	2929.5	1197.246
Low	Low	Low	Sim_Ann	66	106	172	2862.5	909.4943
Low	Low	Low	Sim_Ann	41	101	142	1601.25	814.3293
Low	Low	Low	Sim_Ann	95	297	392	3643.5	2076.354
Low	Low	Low	Sim_Ann	105	480	585	3333.75	1368.514
Low	Low	Low	Sim_Ann	81	161	242	2537.5	1467.724
Low	Low	Low	Sim_Ann	66	186	252	1905.75	658.9811
Low	Low	Low	Sim_Ann	76	206	282	2871.75	757.8456
Low	Low	Low	Sim_Ann	73	313	386	3344.25	777.4648
Low	Low	Low	Sim_Ann	62	76	138	2643.75	1795.322
Low	Low	Low	Sim_Ann	70	156	226	2766.75	952.5607
Low	Low	Low	Sim_Ann	51	139	190	1575	839.6924
Low	Low	Low	Sim_Ann	86	357	443	2945.25	891.9806
Low	Low	Low	Tabu_Srch	43	157	200	2661.75	5751.403
Low	Low	Low	Tabu_Srch	31	351	382	1485.75	2129.258
Low	Low	Low	Tabu_Srch	60	224	284	2845.5	4205.222
Low	Low	Low	Tabu_Srch	102	208	310	3137.5	5414.227
Low	Low	Low	Tabu_Srch	31	71	102	1596	2846.293
Low	Low	Low	Tabu_Srch	82	224	306	3612	7750.327
Low	Low	Low	Tabu_Srch	68	251	319	3144.75	4476.091
Low	Low	Low	Tabu_Srch	102	220	322	2706.25	6259.887
Low	Low	Low	Tabu_Srch	57	159	216	1916.25	3349.801
Low	Low	Low	Tabu_Srch	67	151	218	2856	4411.915
Low	Low	Low	Tabu_Srch	55	141	196	3270.75	6864.893
Low	Low	Low	Tabu_Srch	113	223	336	2993.75	4822.892
Low	Low	Low	Tabu_Srch	50	184	234	2682.75	7514.857
Low	Low	Low	Tabu_Srch	31	297	328	1512	2860.348

Table A4-1. Raw Data, Continued.

Low	Low	Low	Tabu_Srch	67	134	201	2856	4226.961
Low	Low	Med	Loucks	199	56	255	3031.25	38.66406
Low	Low	Med	Loucks	299	57	356	3531.25	52.00781
Low	Low	Med	Loucks	128	49	177	2800	35.14844
Low	Low	Med	Loucks	404	57	461	3087.5	51.19531
Low	Low	Med	Loucks	237	56	293	3575	43.39063
Low	Low	Med	Loucks	179	53	232	3037.5	44.42969
Low	Low	Med	Loucks	122	37	159	3024	36.96094
Low	Low	Med	Loucks	330	68	398	3400	68.5
Low	Low	Med	Loucks	239	85	324	4318.75	40.10156
Low	Low	Med	Loucks	181	20	201	2344.25	41.02344
Low	Low	Med	Loucks	144	34	178	3385.75	41.91406
Low	Low	Med	Loucks	225	64	289	4500	57.78125
Low	Low	Med	Loucks	167	50	217	3837.5	49.60156
Low	Low	Med	Loucks	191	63	254	4306.25	42.57
Low	Low	Med	Sim_Ann	33	47	80	2825	803.16
Low	Low	Med	Sim_Ann	78	162	240	3512.5	1939.897
Low	Low	Med	Sim_Ann	48	88	136	2718.75	934.2608
Low	Low	Med	Sim_Ann	66	137	203	3050	891.964
Low	Low	Med	Sim_Ann	65	133	198	3481.25	1641.486
Low	Low	Med	Sim_Ann	60	151	211	2975	912.8812
Low	Low	Med	Sim_Ann	59	182	241	2968.75	1071.983
Low	Low	Med	Sim_Ann	75	192	267	3325	1566.278
Low	Low	Med	Sim_Ann	80	86	166	3987.5	1314.506
Low	Low	Med	Sim_Ann	18	13	31	2276	1071.214
Low	Low	Med	Sim_Ann	25	16	41	3226.25	834.933
Low	Low	Med	Sim_Ann	65	83	148	4293.75	919.4609
Low	Low	Med	Sim_Ann	64	93	157	3087.5	1629.605
Low	Low	Med	Sim_Ann	51	56	107	3675	1411.74
Low	Low	Med	Sim_Ann	100	216	316	4200	2004.764
Low	Low	Med	Tabu_Srch	88	190	278	3187.5	17213.29
Low	Low	Med	Tabu_Srch	107	231	338	3743.75	20874.25
Low	Low	Med	Tabu_Srch	81	221	302	3006.25	11595.97
Low	Low	Med	Tabu_Srch	105	286	391	3356.25	13071.19
Low	Low	Med	Tabu_Srch	103	193	296	3762.5	7646.517
Low	Low	Med	Tabu_Srch	104	269	373	3300	5459.535
Low	Low	Med	Tabu_Srch	32	125	157	3013.5	7682.857
Low	Low	Med	Tabu_Srch	105	294	399	3550	7842.982
Low	Low	Med	Tabu_Srch	129	231	360	4337.5	7160.318
Low	Low	Med	Tabu_Srch	36	61	97	2469	2511.988
Low	Low	Med	Tabu_Srch	40	43	83	3385.75	7146.139
Low	Low	Med	Tabu_Srch	111	201	312	4625	9174.211
Low	Low	Med	Tabu_Srch	102	191	293	3375	4743.554
Low	Low	Med	Tabu_Srch	100	163	263	4037.5	8462.167
Low	Low	Med	Tabu_Srch	131	303	434	4437.5	6777.957
Low	Med	High	Loucks	170	40	210	3414.75	32.34998

Table A4-1. Raw Data, Continued.

Low	Med	High	Loucks	171	59	230	4451.5	38.78003
Low	Med	High	Loucks	71	47	118	3378.5	31.08997
Low	Med	High	Loucks	294	45	339	2363.5	41.58008
Low	Med	High	Loucks	138	59	197	4567.5	46.03003
Low	Med	High	Loucks	166	59	225	3886	34.77002
Low	Med	High	Loucks	105	47	152	4016.5	34
Low	Med	High	Loucks	99	51	150	3211.75	33.22998
Low	Med	High	Loucks	94	41	135	3849.75	37.17993
Low	Med	High	Loucks	167	64	231	4712.5	41.46997
Low	Med	High	Loucks	207	48	255	4226.75	46.91016
Low	Med	High	Loucks	259	43	302	3356.75	43.93994
Low	Med	High	Loucks	183	41	224	4328.25	48.43994
Low	Med	High	Loucks	102	32	134	3530.75	34.98999
Low	Med	High	Loucks	47	51	98	4502.25	35.37012
Low	Med	High	Sim_Ann	50	37	87	3262.5	1610.704
Low	Med	High	Sim_Ann	58	20	78	4067.25	1372.542
Low	Med	High	Sim_Ann	39	27	66	3175.5	734.5739
Low	Med	High	Sim_Ann	30	64	94	2211.25	3063.197
Low	Med	High	Sim_Ann	70	9	79	4205	1857.685
Low	Med	High	Sim_Ann	46	36	82	3545.25	1180.669
Low	Med	High	Sim_Ann	71	57	128	3842.5	1259.727
Low	Med	High	Sim_Ann	38	4	42	2842	3335.497
Low	Med	High	Sim_Ann	50	19	69	3646.75	1349.329
Low	Med	High	Sim_Ann	69	16	85	4190.5	959.6131
Low	Med	High	Sim_Ann	59	26	85	3922.25	752.0529
Low	Med	High	Sim_Ann	41	15	56	3110.25	1795.531
Low	Med	High	Sim_Ann	63	44	107	4161.5	1524.585
Low	Med	High	Sim_Ann	51	61	112	3451	1126.704
Low	Med	High	Sim_Ann	57	14	71	4176	2012.282
Low	Med	High	Tabu_Srch	92	133	225	3588.75	9622.247
Low	Med	High	Tabu_Srch	120	146	266	4560.25	6893.072
Low	Med	High	Tabu_Srch	110	224	334	3726.5	7197.274
Low	Med	High	Tabu_Srch	78	190	268	2617.25	5367.087
Low	Med	High	Tabu_Srch	141	190	331	4727	10583.81
Low	Med	High	Tabu_Srch	113	189	302	4089	5653.604
Low	Med	High	Tabu_Srch	123	181	304	4263	4796.558
Low	Med	High	Tabu_Srch	89	111	200	3277	7127.227
Low	Med	High	Tabu_Srch	112	165	277	4147	10711.26
Low	Med	High	Tabu_Srch	122	139	261	4574.75	10116.65
Low	Med	High	Tabu_Srch	126	171	297	4458.75	7375.626
Low	Med	High	Tabu_Srch	76	98	174	3422	8735.873
Low	Med	High	Tabu_Srch	127	226	353	4698	10293.9
Low	Med	High	Tabu_Srch	110	206	316	3944	7339.302
Low	Med	High	Tabu_Srch	114	123	237	4618.25	10709.55
Low	Med	Low	Loucks	279	66	345	3719.25	55.48004
Low	Med	Low	Loucks	262	68	330	3132	42.45001

Table A4-1. Raw Data, Continued.

Low	Med	Low	Loucks	116	55	171	3596	52.45996
Low	Med	Low	Loucks	198	46	244	2660.75	34.54999
Low	Med	Low	Loucks	234	83	317	4748.75	58.01001
Low	Med	Low	Loucks	140	74	214	4879.25	48.98999
Low	Med	Low	Loucks	234	69	303	3820.75	43.17999
Low	Med	Low	Loucks	164	77	241	4132.5	42.29004
Low	Med	Low	Loucks	192	58	250	3182.75	39.38
Low	Med	Low	Loucks	73	64	137	4698	44.05005
Low	Med	Low	Loucks	338	79	417	4219.5	55.15002
Low	Med	Low	Loucks	96	83	179	5995.75	65.25
Low	Med	Low	Loucks	117	81	198	5763.75	70.41003
Low	Med	Low	Loucks	63	93	156	6111.75	56.95996
Low	Med	Low	Loucks	28	92	120	5843.5	49.42993
Low	Med	Low	Sim_Ann	89	32	121	3400.25	646.5704
Low	Med	Low	Sim_Ann	58	33	91	2776.75	1543.07
Low	Med	Low	Sim_Ann	57	17	74	3146.5	636.9218
Low	Med	Low	Sim_Ann	33	12	45	2356.25	550.8906
Low	Med	Low	Sim_Ann	87	15	102	4154.25	728.3282
Low	Med	Low	Sim_Ann	95	3	98	4335.5	633.5938
Low	Med	Low	Sim_Ann	87	10	97	3364	610.1094
Low	Med	Low	Sim_Ann	93	15	108	3567	635.7968
Low	Med	Low	Sim_Ann	46	16	62	2733.25	670.0938
Low	Med	Low	Sim_Ann	95	16	111	4118	644.8126
Low	Med	Low	Sim_Ann	66	9	75	3668.5	686.4688
Low	Med	Low	Sim_Ann	149	11	160	5372.25	592.125
Low	Med	Low	Sim_Ann	134	5	139	5154.75	586.3046
Low	Med	Low	Sim_Ann	153	18	171	5459.25	591.9766
Low	Med	Low	Sim_Ann	155	15	170	5299.75	702.711
Low	Med	Low	Tabu_Srch	113	74	187	3581.5	10758.3
Low	Med	Low	Tabu_Srch	101	126	227	3103	10010.37
Low	Med	Low	Tabu_Srch	107	75	182	3523.5	10530.11
Low	Med	Low	Tabu_Srch	80	119	199	2755	4277.104
Low	Med	Low	Tabu_Srch	133	83	216	4495	7287.255
Low	Med	Low	Tabu_Srch	128	44	172	4589.25	11978.08
Low	Med	Low	Tabu_Srch	133	120	253	3741	7298.421
Low	Med	Low	Tabu_Srch	124	50	174	3828	10637
Low	Med	Low	Tabu_Srch	96	86	182	3146.5	5194.014
Low	Med	Low	Tabu_Srch	121	12	133	4357.25	10088.52
Low	Med	Low	Tabu_Srch	105	74	179	3958.5	10457.64
Low	Med	Low	Tabu_Srch	184	36	220	5633.25	11821.77
Low	Med	Low	Tabu_Srch	166	35	201	5415.75	13043.45
Low	Med	Low	Tabu_Srch	175	6	181	5640.5	11681.94
Low	Med	Low	Tabu_Srch	175	5	180	5459.25	17760.98
Low	Med	Med	Loucks	215	40	255	3349.5	36.91003
Low	Med	Med	Loucks	306	48	354	4255.75	56.95996
Low	Med	Med	Loucks	94	56	150	3581.5	30.31995

Table A4-1. Raw Data, Continued.

Low	Med	Med	Loucks	318	51	369	2892.75	43.98999
Low	Med	Med	Loucks	211	47	258	4589.25	49.54004
Low	Med	Med	Loucks	227	50	277	4850.25	38.60999
Low	Med	Med	Loucks	109	64	173	4618.25	42.13
Low	Med	Med	Loucks	231	57	288	3849.75	41.46997
Low	Med	Med	Loucks	80	47	127	2994.25	25.42993
Low	Med	Med	Loucks	62	61	123	4980.75	39.92993
Low	Med	Med	Loucks	98	44	142	3298.75	34.15991
Low	Med	Med	Loucks	254	45	299	2552	42.72998
Low	Med	Med	Loucks	145	53	198	4089	31.91003
Low	Med	Med	Loucks	238	63	301	3936.75	42.12
Low	Med	Med	Loucks	277	54	331	3458.25	31.91992
Low	Med	Med	Sim_Ann	58	66	124	3204.5	985.2095
Low	Med	Med	Sim_Ann	63	79	142	4118	1066.227
Low	Med	Med	Sim_Ann	39	34	73	3262.5	973.1096
Low	Med	Med	Sim_Ann	65	127	192	2878.25	806.3563
Low	Med	Med	Sim_Ann	90	78	168	4408	1816.872
Low	Med	Med	Sim_Ann	59	1	60	4495	971.3322
Low	Med	Med	Sim_Ann	66	18	84	4226.75	931.1801
Low	Med	Med	Sim_Ann	54	26	80	3523.5	1018.833
Low	Med	Med	Sim_Ann	28	18	46	2711.5	1517.404
Low	Med	Med	Sim_Ann	54	4	58	4524	728.7225
Low	Med	Med	Sim_Ann	37	13	50	3016	1019.898
Low	Med	Med	Sim_Ann	35	65	100	2392.5	1496.205
Low	Med	Med	Sim_Ann	50	12	62	3733.75	1231.175
Low	Med	Med	Sim_Ann	63	38	101	3610.5	885.9002
Low	Med	Med	Sim_Ann	41	50	91	3226.25	1478.394
Low	Med	Med	Tabu_Srch	99	169	268	3523.5	8479.467
Low	Med	Med	Tabu_Srch	118	208	326	4574.75	9061.945
Low	Med	Med	Tabu_Srch	94	153	247	3719.25	7776.426
Low	Med	Med	Tabu_Srch	125	317	442	3371.25	7401.425
Low	Med	Med	Tabu_Srch	135	155	290	4770.5	7829.894
Low	Med	Med	Tabu_Srch	101	83	184	4843	11087.57
Low	Med	Med	Tabu_Srch	122	128	250	4640	11711.88
Low	Med	Med	Tabu_Srch	105	157	262	3951.25	5642.389
Low	Med	Med	Tabu_Srch	76	120	196	3103	4886.335
Low	Med	Med	Tabu_Srch	131	143	274	5118.5	3117.3
Low	Med	Med	Tabu_Srch	92	144	236	3458.25	8230.509
Low	Med	Med	Tabu_Srch	92	216	308	2820.25	1187.308
Low	Med	Med	Tabu_Srch	107	145	252	4197.75	5736.227
Low	Med	Med	Tabu_Srch	95	96	191	3915	2773.574
Low	Med	Med	Tabu_Srch	117	280	397	3842.5	2989.664
Med	High	High	Loucks	104	34	138	5387.25	43.66992
Med	High	High	Loucks	79	51	130	7260	73.81934
Med	High	High	Loucks	81	58	139	6154.5	44.31934
Med	High	High	Loucks	90	60	150	6245.25	50.75

Table A4-1. Raw Data, Continued.

Med	High	High	Loucks	100	50	150	5544	50.31055
Med	High	High	Loucks	31	59	90	6575.25	48.61035
Med	High	High	Loucks	93	41	134	5799.75	45.80957
Med	High	High	Loucks	140	55	195	7242.75	59.26953
Med	High	High	Loucks	100	23	123	5445	83
Med	High	High	Loucks	97	66	163	8084.5	63.92969
Med	High	High	Loucks	99	45	144	5197.5	40.08984
Med	High	High	Loucks	35	51	86	7501.75	59.25977
Med	High	High	Loucks	185	38	223	4677.75	41.2002
Med	High	High	Loucks	112	36	148	4372.5	42.73047
Med	High	High	Loucks	283	46	329	5098.5	56.68945
Med	High	High	Sim_Ann	87	29	116	5115	1195.172
Med	High	High	Sim_Ann	132	12	144	6740.25	913.1876
Med	High	High	Sim_Ann	90	10	100	5593.5	904.6094
Med	High	High	Sim_Ann	118	17	135	5610	915.4804
Med	High	High	Sim_Ann	69	10	79	5082	1212.875
Med	High	High	Sim_Ann	91	1	92	6096.75	1072.234
Med	High	High	Sim_Ann	73	7	80	5403.75	1303.594
Med	High	High	Sim_Ann	109	8	117	6660	1084.355
Med	High	High	Sim_Ann	83	1	84	5247	754.1338
Med	High	High	Sim_Ann	150	19	169	7298.25	891.9688
Med	High	High	Sim_Ann	50	12	62	4727.25	1279.75
Med	High	High	Sim_Ann	102	18	120	6863.5	866.1856
Med	High	High	Sim_Ann	51	20	71	4397.25	1123.555
Med	High	High	Sim_Ann	49	19	68	4116.75	869.4766
Med	High	High	Sim_Ann	57	5	62	4677.75	917.8984
Med	High	High	Tabu_Srch	146	144	290	5610	7438.066
Med	High	High	Tabu_Srch	196	86	282	7284.75	5000.086
Med	High	High	Tabu_Srch	150	86	236	6096.75	6859.852
Med	High	High	Tabu_Srch	166	45	211	6055.5	9909.55
Med	High	High	Tabu_Srch	138	129	267	5742	4147.257
Med	High	High	Tabu_Srch	123	39	162	6402	10546.68
Med	High	High	Tabu_Srch	129	69	198	5890.5	7472.119
Med	High	High	Tabu_Srch	153	46	199	7094.75	11427.97
Med	High	High	Tabu_Srch	152	128	280	5832.75	8102.617
Med	High	High	Tabu_Srch	194	31	225	7751.5	9681.277
Med	High	High	Tabu_Srch	129	139	268	5436.75	6968.51
Med	High	High	Tabu_Srch	171	99	270	7548	5792.108
Med	High	High	Tabu_Srch	138	221	359	5148	8911.778
Med	High	High	Tabu_Srch	115	159	274	4686	5416.699
Med	High	High	Tabu_Srch	135	171	306	5370.75	5305.424
Med	High	Low	Loucks	100	58	158	8029	115.4502
Med	High	Low	Loucks	236	90	326	7210.5	84.25
Med	High	Low	Loucks	22	69	91	8047.5	285.3398
Med	High	Low	Loucks	44	66	110	6756.75	94.2002
Med	High	Low	Loucks	154	81	235	8352.75	147.5303

Table A4-1. Raw Data, Continued.

Med	High	Low	Loucks	141	84	225	7334.25	103.9697
Med	High	Low	Loucks	129	84	213	7359	126.9307
Med	High	Low	Loucks	107	71	178	7326	131.8701
Med	High	Low	Loucks	184	83	267	7152.75	72.83008
Med	High	Low	Sim_Ann	187	35	222	7224.25	818.6016
Med	High	Low	Sim_Ann	178	45	223	4560.25	1197.063
Med	High	Low	Sim_Ann	32	8	40	2557.5	961.9688
Med	High	Low	Sim_Ann	178	16	194	6352.5	734.8046
Med	High	Low	Sim_Ann	178	8	186	7335.25	1064.133
Med	High	Low	Sim_Ann	84	14	98	4551	667.4532
Med	High	Low	Sim_Ann	182	35	217	6789.75	944.375
Med	High	Low	Sim_Ann	46	18	64	2886	731.9532
Med	High	Low	Sim_Ann	144	29	173	6435	983.836
Med	High	Low	Sim_Ann	171	13	184	6558.75	744.5782
Med	High	Low	Sim_Ann	152	8	160	6493.5	936.9394
Med	High	Low	Sim_Ann	171	10	181	6660	763.145
Med	High	Low	Sim_Ann	139	22	161	6591.75	892.2002
Med	High	Low	Sim_Ann	204	14	218	6459.75	755.7578
Med	High	Low	Sim_Ann	37	12	49	2598.75	1058.305
Med	High	Low	Tabu_Srch	210	12	222	7409.25	8462.034
Med	High	Low	Tabu_Srch	199	24	223	4726.75	5155.003
Med	High	Low	Tabu_Srch	55	53	108	2763.75	3679.271
Med	High	Low	Tabu_Srch	190	10	200	6484.5	8295.267
Med	High	Low	Tabu_Srch	180	10	190	7390.75	11326.3
Med	High	Low	Tabu_Srch	101	5	106	4699	6079.922
Med	High	Low	Tabu_Srch	186	31	217	6864	5497.299
Med	High	Low	Tabu_Srch	59	45	104	3061.75	9085.802
Med	High	Low	Tabu_Srch	164	11	175	6624.75	11838.1
Med	High	Low	Tabu_Srch	191	11	202	6756.75	10701.26
Med	High	Low	Tabu_Srch	199	43	242	6900.5	14113.67
Med	High	Low	Tabu_Srch	186	7	193	6798.75	15656.73
Med	High	Low	Tabu_Srch	162	11	173	6748.5	11756.63
Med	High	Low	Tabu_Srch	205	13	218	6484.5	14710.02
Med	High	Low	Tabu_Srch	60	41	101	2796.75	5923.634
Med	High	Med	Loucks	141	33	174	5832.75	71.9502
Med	High	Med	Loucks	81	35	116	5387.25	41.95996
Med	High	Med	Loucks	80	32	112	4851	44.97949
Med	High	Med	Loucks	53	59	112	6501	52.39941
Med	High	Med	Loucks	26	50	76	5883	53.2793
Med	High	Med	Loucks	61	44	105	6550.5	66.24023
Med	High	Med	Loucks	135	42	177	5395.5	45.75977
Med	High	Med	Loucks	17	66	83	6740.25	50.2002
Med	High	Med	Loucks	46	45	91	6575.25	52.78027
Med	High	Med	Loucks	131	49	180	5824.5	46.84961
Med	High	Med	Loucks	77	39	116	5469.75	41.41992
Med	High	Med	Loucks	105	46	151	5214	37.12988

Table A4-1. Raw Data, Continued.

Med	High	Med	Loucks	53	39	92	5717.25	40.47949
Med	High	Med	Loucks	109	39	148	6278.25	56.08008
Med	High	Med	Loucks	76	53	129	5758.5	42.18066
Med	High	Med	Sim_Ann	95	10	105	5494.5	959.5468
Med	High	Med	Sim_Ann	59	7	66	5040.75	1096.398
Med	High	Med	Sim_Ann	59	16	75	4653	1664.34
Med	High	Med	Sim_Ann	91	9	100	5940	1078.641
Med	High	Med	Sim_Ann	56	9	65	5337.25	1488.914
Med	High	Med	Sim_Ann	137	15	152	6162.75	1019.625
Med	High	Med	Sim_Ann	90	5	95	5007.75	712.0626
Med	High	Med	Sim_Ann	112	16	128	6063.75	820.3594
Med	High	Med	Sim_Ann	97	17	114	6063.75	856.2968
Med	High	Med	Sim_Ann	81	7	88	5412	823.8798
Med	High	Med	Sim_Ann	69	21	90	4974.75	847.1718
Med	High	Med	Sim_Ann	65	19	84	4776.75	819.9238
Med	High	Med	Sim_Ann	80	8	88	5346	897.0468
Med	High	Med	Sim_Ann	121	5	126	5915.25	877.6894
Med	High	Med	Sim_Ann	83	5	88	5346	702.8282
Med	High	Med	Tabu_Srch	130	51	181	5865.75	9813.695
Med	High	Med	Tabu_Srch	134	132	266	5700.75	6041.461
Med	High	Med	Tabu_Srch	104	111	215	5073.75	6935.43
Med	High	Med	Tabu_Srch	164	118	282	6550.5	10330.87
Med	High	Med	Tabu_Srch	141	168	309	6169.75	9889.275
Med	High	Med	Tabu_Srch	171	59	230	6451.5	10876.08
Med	High	Med	Tabu_Srch	156	121	277	5618.25	8148.905
Med	High	Med	Tabu_Srch	145	23	168	6402	11330.81
Med	High	Med	Tabu_Srch	171	93	264	6690.75	5700.898
Med	High	Med	Tabu_Srch	131	83	214	5857.5	9263.625
Med	High	Med	Tabu_Srch	139	75	214	5593.5	6712.223
Med	High	Med	Tabu_Srch	125	93	218	5304.75	5198.697
Med	High	Med	Tabu_Srch	130	98	228	5832.75	3562.57
Med	High	Med	Tabu_Srch	190	126	316	6492.75	6949.164
Med	High	Med	Tabu_Srch	161	171	332	6030.75	4281.176
Med	Low	High	Loucks	229	57	286	4331.25	43.39014
Med	Low	High	Loucks	180	39	219	3365.25	48.93994
Med	Low	High	Loucks	125	65	190	4868.75	41.46973
Med	Low	High	Loucks	234	71	305	4487.5	42.24023
Med	Low	High	Loucks	158	69	227	4831.25	52.67969
Med	Low	High	Loucks	211	72	283	4650	43.83008
Med	Low	High	Loucks	269	78	347	4437.5	56.52002
Med	Low	High	Loucks	96	43	139	3937.5	47.07031
Med	Low	High	Loucks	243	37	280	3097.5	45.64014
Med	Low	High	Loucks	237	34	271	3265.5	60.68994
Med	Low	High	Loucks	132	38	170	3108	41.8501
Med	Low	High	Sim_Ann	96	211	307	3906	1360.652
Med	Low	High	Sim_Ann	57	281	338	3654	1030.231

Table A4-1. Raw Data, Continued.

Med	Low	High	Sim_Ann	79	147	226	4181.25	1197.615
Med	Low	High	Sim_Ann	42	103	145	3375.75	1394.682
Med	Low	High	Sim_Ann	64	72	136	3825	1533.711
Med	Low	High	Sim_Ann	55	14	69	4475	1670.368
Med	Low	High	Sim_Ann	91	130	221	4331.25	1714.152
Med	Low	High	Sim_Ann	112	148	260	4675	1436.969
Med	Low	High	Sim_Ann	68	47	115	4293.75	1434.713
Med	Low	High	Sim_Ann	76	82	158	4100	1393.759
Med	Low	High	Sim_Ann	71	222	293	3953.25	1264.553
Med	Low	High	Sim_Ann	45	127	172	3696	1076.865
Med	Low	High	Sim_Ann	41	194	235	3113.25	1029.591
Med	Low	High	Sim_Ann	36	136	172	3276	1072.928
Med	Low	High	Sim_Ann	42	125	167	3123.75	1104.441
Med	Low	High	Tabu_Srch	78	199	277	3801	4229.696
Med	Low	High	Tabu_Srch	36	98	134	3570	4198
Med	Low	High	Tabu_Srch	108	164	272	4431.25	8448.304
Med	Low	High	Tabu_Srch	42	119	161	3417.75	4226.261
Med	Low	High	Tabu_Srch	120	272	392	4206.25	8657.672
Med	Low	High	Tabu_Srch	104	99	203	4843.75	6208.404
Med	Low	High	Tabu_Srch	145	298	443	4700	9253.6
Med	Low	High	Tabu_Srch	148	278	426	4931.25	6265.438
Med	Low	High	Tabu_Srch	124	151	275	4712.5	8576.398
Med	Low	High	Tabu_Srch	126	286	412	4456.25	5879.25
Med	Low	High	Tabu_Srch	63	154	217	3911.25	4883.992
Med	Low	High	Tabu_Srch	44	154	198	3701.25	6220.18
Med	Low	High	Tabu_Srch	29	98	127	3071.25	5299.454
Med	Low	High	Tabu_Srch	28	102	130	3244.5	5769.06
Med	Low	High	Tabu_Srch	34	87	121	3102.75	3732.306
Med	Low	Low	Loucks	65	39	104	2415	28.5
Med	Low	Low	Loucks	142	35	177	3506.25	39.48975
Med	Low	Low	Loucks	146	44	190	2893.75	24.99023
Med	Low	Low	Loucks	114	38	152	2982	43.17969
Med	Low	Low	Loucks	101	33	134	2350	24.10986
Med	Low	Low	Loucks	88	60	148	4331.25	35.81006
Med	Low	Low	Loucks	134	55	189	4425.75	89.08984
Med	Low	Low	Loucks	156	90	246	5500	67.72998
Med	Low	Low	Loucks	89	69	158	4814.25	67.33008
Med	Low	Low	Sim_Ann	43	81	124	2425.5	1261.411
Med	Low	Low	Sim_Ann	70	11	81	3306.25	1747.551
Med	Low	Low	Sim_Ann	51	26	77	2706.25	2784.96
Med	Low	Low	Sim_Ann	41	27	68	2662.5	1721.535
Med	Low	Low	Sim_Ann	44	152	196	3013.5	1666.869
Med	Low	Low	Sim_Ann	45	49	94	2275	2821.674
Med	Low	Low	Sim_Ann	30	89	119	2252.25	2642.253
Med	Low	Low	Sim_Ann	33	54	87	2404.5	2276.967
Med	Low	Low	Sim_Ann	102	39	141	4037.5	1507.436

Table A4-1. Raw Data, Continued.

Med	Low	Low	Sim_Ann	45	107	152	4373.25	1527.623
Med	Low	Low	Sim_Ann	67	66	133	3300	1650.5
Med	Low	Low	Sim_Ann	38	82	120	3302.25	2121.262
Med	Low	Low	Sim_Ann	62	146	208	4788	1719.307
Med	Low	Low	Sim_Ann	115	1	116	4931.25	967.1354
Med	Low	Low	Sim_Ann	60	114	174	4756.5	2174.806
Med	Low	Low	Tabu_Srch	37	107	144	2425.5	2503.156
Med	Low	Low	Tabu_Srch	99	80	179	3537.5	3329.594
Med	Low	Low	Tabu_Srch	61	40	101	2800	5886.153
Med	Low	Low	Tabu_Srch	73	69	142	2893.75	3573.966
Med	Low	Low	Tabu_Srch	40	98	138	3050.25	2735.606
Med	Low	Low	Tabu_Srch	60	96	156	2387.5	2122.75
Med	Low	Low	Tabu_Srch	30	93	123	2294.25	2442.207
Med	Low	Low	Tabu_Srch	27	126	153	2399.25	2154.414
Med	Low	Low	Tabu_Srch	120	53	173	4187.5	6332.785
Med	Low	Low	Tabu_Srch	51	99	150	4425.75	4019.943
Med	Low	Low	Tabu_Srch	102	143	245	3568.75	3961.482
Med	Low	Low	Tabu_Srch	41	105	146	3328.5	2407.207
Med	Low	Low	Tabu_Srch	73	171	244	4872	4415.373
Med	Low	Low	Tabu_Srch	151	39	190	5187.5	11124.87
Med	Low	Low	Tabu_Srch	64	112	176	4798.5	7548.639
Med	Low	Med	Loucks	203	67	270	4375	47.61963
Med	Low	Med	Loucks	248	27	275	3291.75	58.71973
Med	Low	Med	Loucks	126	69	195	4456.25	45.58984
Med	Low	Med	Loucks	193	33	226	3118.5	39.33008
Med	Low	Med	Loucks	184	38	222	3454.5	40.75
Med	Low	Med	Loucks	212	51	263	4212.5	58
Med	Low	Med	Loucks	253	57	310	4493.75	52.02002
Med	Low	Med	Loucks	185	85	270	4531.25	45.75977
Med	Low	Med	Loucks	233	59	292	4500	56.07959
Med	Low	Med	Loucks	213	52	265	4743.75	74.15039
Med	Low	Med	Loucks	317	45	362	3685.5	106.1201
Med	Low	Med	Loucks	120	59	179	4162.5	38.56006
Med	Low	Med	Loucks	119	66	185	4825	38.78027
Med	Low	Med	Loucks	214	55	269	3868.75	41.35986
Med	Low	Med	Sim_Ann	28	182	210	3302.25	1176.266
Med	Low	Med	Sim_Ann	50	165	215	3407.25	1289.344
Med	Low	Med	Sim_Ann	34	112	146	3155.25	1044.578
Med	Low	Med	Sim_Ann	38	118	156	3491.25	1128.602
Med	Low	Med	Sim_Ann	24	330	354	3071.25	1202.633
Med	Low	Med	Sim_Ann	38	182	220	3433.5	1576.258
Med	Low	Med	Sim_Ann	39	105	144	3895.5	1101.039
Med	Low	Med	Sim_Ann	70	45	115	4375	1148.465
Med	Low	Med	Sim_Ann	75	83	158	4306.25	1384.24
Med	Low	Med	Sim_Ann	101	89	190	4937.5	2208.998
Med	Low	Med	Sim_Ann	94	196	290	4731.25	1138.279

Table A4-1. Raw Data, Continued.

Med	Low	Med	Sim_Ann	42	127	169	3643.5	1143.998
Med	Low	Med	Sim_Ann	38	140	178	3664.5	1047.859
Med	Low	Med	Sim_Ann	64	183	247	3958.5	1768.383
Med	Low	Med	Sim_Ann	73	39	112	3668.75	935.7226
Med	Low	Med	Tabu_Srch	30	52	82	3354.75	3638.032
Med	Low	Med	Tabu_Srch	32	77	109	3323.25	5996.126
Med	Low	Med	Tabu_Srch	24	104	128	3150	5800.89
Med	Low	Med	Tabu_Srch	32	86	118	3501.75	2217.226
Med	Low	Med	Tabu_Srch	26	90	116	3123.75	5171.868
Med	Low	Med	Tabu_Srch	39	123	162	3459.75	3668.602
Med	Low	Med	Tabu_Srch	37	83	120	3942.75	9106.852
Med	Low	Med	Tabu_Srch	125	192	317	4737.5	6393.968
Med	Low	Med	Tabu_Srch	129	179	308	4681.25	8052.21
Med	Low	Med	Tabu_Srch	165	233	398	5412.5	9754.562
Med	Low	Med	Tabu_Srch	115	201	316	4906.25	6961.242
Med	Low	Med	Tabu_Srch	29	58	87	3638.25	3230.71
Med	Low	Med	Tabu_Srch	44	156	200	3690.75	4212.336
Med	Low	Med	Tabu_Srch	55	146	201	3900.75	4812.554
Med	Low	Med	Tabu_Srch	117	167	284	3975	2940.602
Med	Med	High	Loucks	181	63	244	4872	117.3701
Med	Med	High	Loucks	119	61	180	6409	49.97949
Med	Med	High	Loucks	85	56	141	5183.75	46.2998
Med	Med	High	Loucks	19	57	76	5089.5	33.29004
Med	Med	High	Loucks	66	49	115	5133	39.06055
Med	Med	High	Loucks	73	59	132	4915.5	42.01953
Med	Med	High	Loucks	85	56	141	5829	45.25977
Med	Med	High	Loucks	136	44	180	4640	46.95996
Med	Med	High	Loucks	275	63	338	4002	33.71973
Med	Med	High	Loucks	86	49	135	5227.25	43.5
Med	Med	High	Loucks	110	55	165	4930	43.83008
Med	Med	High	Loucks	119	53	172	5046	43.33984
Med	Med	High	Loucks	144	49	193	5829	58.33008
Med	Med	High	Loucks	133	62	195	5234.5	41.4707
Med	Med	High	Sim_Ann	69	14	83	4415.25	978.2188
Med	Med	High	Sim_Ann	104	16	120	5850.75	1111.57
Med	Med	High	Sim_Ann	95	0	95	4777.75	860.6718
Med	Med	High	Sim_Ann	63	2	65	4690.75	1108.484
Med	Med	High	Sim_Ann	66	7	73	4756	997.9922
Med	Med	High	Sim_Ann	64	19	83	4364.5	1121.133
Med	Med	High	Sim_Ann	105	9	114	5372.25	1007.43
Med	Med	High	Sim_Ann	74	38	112	4277.5	1062.391
Med	Med	High	Sim_Ann	44	61	105	3639.5	1356.375
Med	Med	High	Sim_Ann	81	2	83	4857.5	1123.564
Med	Med	High	Sim_Ann	73	33	106	4625.5	1503.859
Med	Med	High	Sim_Ann	80	11	91	4712.5	783.127
Med	Med	High	Sim_Ann	87	28	115	5328.75	1042.172

Table A4-1. Raw Data, Continued.

Med	Med	High	Sim_Ann	86	56	142	4886.5	818.7188
Med	Med	High	Sim_Ann	74	41	115	4451.5	965.4766
Med	Med	High	Tabu_Srch	90	51	141	4603.75	10170.55
Med	Med	High	Tabu_Srch	138	32	170	6126.25	12442.79
Med	Med	High	Tabu_Srch	137	82	219	5133	11192.97
Med	Med	High	Tabu_Srch	157	238	395	5408.5	5442.532
Med	Med	High	Tabu_Srch	114	71	185	5162	11338.63
Med	Med	High	Tabu_Srch	126	85	211	4879.25	8157.196
Med	Med	High	Tabu_Srch	182	126	308	5952.25	11762.29
Med	Med	High	Tabu_Srch	138	136	274	4785	8059.454
Med	Med	High	Tabu_Srch	121	390	511	4263	1316.773
Med	Med	High	Tabu_Srch	154	139	293	5444.75	9688.102
Med	Med	High	Tabu_Srch	127	149	276	5053.25	3079.884
Med	Med	High	Tabu_Srch	141	156	297	5220	6573.16
Med	Med	High	Tabu_Srch	149	60	209	5814.5	2695.346
Med	Med	High	Tabu_Srch	176	300	476	5568	2780.887
Med	Med	High	Tabu_Srch	148	177	325	5038.75	8395.376
Med	Med	Low	Loucks	64	57	121	4161.5	38.06006
Med	Med	Low	Loucks	32	52	84	6496	92.06006
Med	Med	Low	Loucks	84	89	173	7068.75	109.6299
Med	Med	Low	Loucks	129	77	206	5887	59.38037
Med	Med	Low	Loucks	75	73	148	6510.5	69.08984
Med	Med	Low	Loucks	162	74	236	6387.25	235.1899
Med	Med	Low	Loucks	101	74	175	6235	68.99023
Med	Med	Low	Loucks	111	74	185	6285.75	72.06006
Med	Med	Low	Loucks	94	89	183	6256.75	60.7002
Med	Med	Low	Loucks	47	81	128	5923.25	52.11963
Med	Med	Low	Loucks	29	94	123	6293	70.41016
Med	Med	Low	Loucks	47	80	127	6278.5	81.02002
Med	Med	Low	Loucks	50	80	130	6278.5	81.33984
Med	Med	Low	Loucks	46	76	122	6300.25	74.85986
Med	Med	Low	Sim_Ann	82	29	111	3770	859.4949
Med	Med	Low	Sim_Ann	159	3	162	5720.25	1002.132
Med	Med	Low	Sim_Ann	177	10	187	6061	963.9311
Med	Med	Low	Sim_Ann	133	19	152	5321.5	794.2955
Med	Med	Low	Sim_Ann	107	24	131	5183.75	682.6358
Med	Med	Low	Sim_Ann	324	53	377	5626	1122.554
Med	Med	Low	Sim_Ann	166	10	176	5807.25	732.8471
Med	Med	Low	Sim_Ann	151	15	166	5604.25	801.4178
Med	Med	Low	Sim_Ann	157	12	169	5676.75	745.9403
Med	Med	Low	Sim_Ann	131	23	154	5473.75	963.7977
Med	Med	Low	Sim_Ann	135	13	148	5444.75	745.7781
Med	Med	Low	Sim_Ann	149	30	179	5597	789.9418
Med	Med	Low	Sim_Ann	172	26	198	5756.5	798.5589
Med	Med	Low	Sim_Ann	165	34	199	5705.75	774.1353
Med	Med	Low	Sim_Ann	157	31	188	5676.75	777.9511

Table A4-1. Raw Data, Continued.

Med	Med	Low	Tabu_Srch	121	72	193	4089	4214.455
Med	Med	Low	Tabu_Srch	155	17	172	5749.25	9894.763
Med	Med	Low	Tabu_Srch	178	9	187	6068.25	9910.408
Med	Med	Low	Tabu_Srch	156	20	176	5495.5	10657.65
Med	Med	Low	Tabu_Srch	130	17	147	5336	11946.71
Med	Med	Low	Tabu_Srch	344	33	377	5785.5	5604.286
Med	Med	Low	Tabu_Srch	190	22	212	6024.75	11574.52
Med	Med	Low	Tabu_Srch	163	9	172	5684	10457.44
Med	Med	Low	Tabu_Srch	155	16	171	5655	12828.04
Med	Med	Low	Tabu_Srch	145	15	160	5611.5	12542.54
Med	Med	Low	Tabu_Srch	150	12	162	5589.75	12582.35
Med	Med	Low	Tabu_Srch	164	19	183	5676.75	6026.188
Med	Med	Low	Tabu_Srch	190	16	206	5879.75	6502.94
Med	Med	Low	Tabu_Srch	186	19	205	5821.75	10400.03
Med	Med	Low	Tabu_Srch	168	20	188	5756.5	9122.642
Med	Med	Med	Loucks	81	50	131	4886.5	40.59033
Med	Med	Med	Loucks	122	68	190	5604.25	174.4399
Med	Med	Med	Loucks	77	43	120	4277.5	35.41992
Med	Med	Med	Loucks	32	58	90	6691.75	44.49023
Med	Med	Med	Loucks	39	56	95	5959.5	42.79004
Med	Med	Med	Loucks	169	42	211	5633.25	49.98975
Med	Med	Med	Loucks	68	53	121	5046	109.46
Med	Med	Med	Loucks	128	42	170	4930	55.81055
Med	Med	Med	Loucks	58	55	113	5270.75	41.62988
Med	Med	Med	Loucks	212	43	255	5974	56.90918
Med	Med	Med	Loucks	102	37	139	5270.75	55.96973
Med	Med	Med	Loucks	144	54	198	4748.75	40.59082
Med	Med	Med	Loucks	69	50	119	5017	40.91992
Med	Med	Med	Loucks	87	41	128	5140.25	38.4502
Med	Med	Med	Loucks	150	63	213	5959.5	72.4502
Med	Med	Med	Sim_Ann	80	4	84	4538.5	1471.757
Med	Med	Med	Sim_Ann	69	19	88	4988	1488.24
Med	Med	Med	Sim_Ann	51	14	65	4038.25	869.3614
Med	Med	Med	Sim_Ann	119	8	127	6213.25	970.9021
Med	Med	Med	Sim_Ann	99	6	105	5539	1285.601
Med	Med	Med	Sim_Ann	85	15	100	5234.5	1037.638
Med	Med	Med	Sim_Ann	73	2	75	4676.25	1533.612
Med	Med	Med	Sim_Ann	72	33	105	4632.75	968.059
Med	Med	Med	Sim_Ann	59	5	64	4864.75	949.3903
Med	Med	Med	Sim_Ann	69	21	90	4937.25	841.5786
Med	Med	Med	Sim_Ann	84	26	110	4959	1292.436
Med	Med	Med	Sim_Ann	60	14	74	4415.25	1038.638
Med	Med	Med	Sim_Ann	79	24	103	4669	1228.901
Med	Med	Med	Sim_Ann	62	5	67	4821.25	1029.248
Med	Med	Med	Sim_Ann	101	3	104	5481	861.9123
Med	Med	Med	Tabu_Srch	113	81	194	4843	10782.52

Table A4-1. Raw Data, Continued.

Med	Med	Med	Tabu_Srch	138	104	242	5524.5	7365.099
Med	Med	Med	Tabu_Srch	95	112	207	4437	10046.44
Med	Med	Med	Tabu_Srch	187	110	297	6749.75	6023.649
Med	Med	Med	Tabu_Srch	145	62	207	5894.25	10660.56
Med	Med	Med	Tabu_Srch	161	133	294	5836.25	11688.02
Med	Med	Med	Tabu_Srch	136	131	267	5169.25	6490.522
Med	Med	Med	Tabu_Srch	148	169	317	5198.25	6636.888
Med	Med	Med	Tabu_Srch	154	222	376	5568	7911.051
Med	Med	Med	Tabu_Srch	171	107	278	5966.75	2069.955
Med	Med	Med	Tabu_Srch	140	156	296	5379.5	6363.073
Med	Med	Med	Tabu_Srch	114	124	238	4835.75	8740.931
Med	Med	Med	Tabu_Srch	162	223	385	5307	5679.787
Med	Med	Med	Tabu_Srch	156	223	379	5560.75	10999.57
Med	Med	Med	Tabu_Srch	139	45	184	5778.25	8055.93

APPENDIX 5

TRANSFORMED TREATMENT AVERAGES

The following table contains the averages for all treatments, transformed.

Table A5-1, Treatment Averages Of Transformed Data.

Loucks Algorithm					
Prob Code	TOS, TR	TSD, TR	OBJ, TR	COST, TR	TIME, TR
LLL	3.68756981	2.623659	5.448964	360.5765	4.276163
LLM	3.80919794	2.678535	5.562366	451.0143	3.812422
LLH	3.80474414	2.727923	5.575478	452.2695	3.741555
LML	3.52946296	2.916315	5.458652	536.7941	3.873745
LMM	3.64961788	2.695987	5.444545	497.6382	3.650642
LMH	3.41354138	2.641605	5.221404	499.157	3.641273
LHL	3.29464376	2.812526	5.198295	629.0174	4.017039
LHM	3.53852246	2.572244	5.305293	534.1921	3.732213
LHH	3.20911262	2.572255	5.008788	564.3802	3.726088
MLL	3.28765375	2.665968	5.127013	481.5239	3.796806
MLM	3.77149454	2.682186	5.545857	511.2446	3.936986
MLH	3.64453096	2.712386	5.441346	507.3724	3.859069
MML	2.91130408	2.946763	5.007029	696.879	4.355322
MMM	3.21383325	2.642246	5.052077	612.9689	4.01571
MMH	3.19504784	2.720401	5.081813	600.5305	3.830925
MHL	3.10515068	2.98098	5.180546	806.8586	4.833689
MHM	2.90124152	2.582176	4.784963	677.1428	3.904514
MHH	3.16849647	2.592148	5.005658	673.647	3.933642
HLL	3.39205746	2.337598	5.086168	503.8827	3.783275
HLM	3.26098437	2.321544	4.954461	508.3235	3.642734
HLH	3.28419721	2.222774	4.94931	482.9435	3.549933
HML	3.1042442	2.413765	4.846642	619.4568	3.811938
HMM	2.95684714	2.420609	4.714285	625.6777	3.804802
HMH	3.00204943	2.312057	4.707661	545.7676	3.529322
HHL	2.9123058	2.309074	4.625746	738.3506	3.946244
HHM	2.80910885	2.324815	4.52878	743.9041	3.98554

Table A5-1, Treatment Averages Of Transformed Data, Continued.

HMH	3.00204943	2.312057	4.707661	545.7676	3.529322
HHL	2.9123058	2.309074	4.625746	738.3506	3.946244
HHM	2.80910885	2.324815	4.52878	743.9041	3.98554
HHH	2.9768455	2.017801	4.561662	575.5501	3.489987
Simulated Annealing					
Prob Code	TOS, TR	TSD, TR	OBJ, TR	COST, TR	TIME, TR
LLL	2.88500219	3.811668	5.620798	363.0179	6.869889
LLM	2.7280887	3.142305	4.990172	424.7753	7.102331
LLH	2.64049174	2.913566	4.802522	432.1944	7.389616
LML	3.10208378	1.9188	4.679741	507.4568	6.533043
LMM	2.67403013	2.450024	4.502941	452.1348	7.010332
LMH	2.69220397	2.217111	4.364174	468.6283	7.257451
LHL	3.13553783	1.941385	4.677791	563.7331	6.498659
LHM	2.64549738	2.143129	4.352516	485.4361	6.875115
LHH	2.71655589	1.76749	4.226439	530.3867	6.65227
MLL	2.6436337	2.907326	4.815162	427.8137	7.565333
MLM	2.64268339	3.330769	5.194478	478.9016	7.149058
MLH	2.82367666	3.232949	5.174868	494.4134	7.186962
MML	3.50801501	2.054536	5.137057	634.963	6.730929
MMM	2.9528344	1.745513	4.469544	587.8412	7.013154
MMH	2.95585606	2.018074	4.593008	569.8974	6.914176
MHL	3.49085141	2.070007	5.118077	684.5016	6.747708
MHM	3.00671567	1.781411	4.526797	625.0043	6.856451
MHH	3.05394218	1.82278	4.592587	655.5758	6.948315
HLL	2.85191535	3.501189	5.392766	514.206	7.40888
HLM	2.82505204	3.338172	5.278227	527.0655	7.711407
HLH	2.72651845	3.404583	5.263612	499.6395	7.506961
HML	3.0147545	2.20262	4.680619	609.257	7.328474
HMM	3.03119367	2.174349	4.692431	613.6116	7.768487
HMH	2.88237409	2.289503	4.585005	530.5328	7.419882
HHL	3.20725993	2.280839	4.903232	728.9325	6.916515
HHM	3.21673299	2.071022	4.844266	716.4646	7.254684
HHH	2.74941379	2.169311	4.373568	534.4027	7.057893
Prob Code	TOS, TR	TSD, TR	OBJ, TR	COST, TR	TIME, TR
LLL	2.78726578	3.672179	5.488974	372.1571	8.436559
LLM	3.02560503	3.6442	5.537719	461.5865	8.95993
LLH	2.94361764	3.427997	5.341951	446.5498	7.582972
LML	3.37160237	2.63482	5.260555	525.6285	9.204443
LMM	3.19658124	3.530166	5.561432	496.9625	8.627637
LMH	3.23234244	3.580901	5.611644	506.5952	8.96597
LHL	3.51946979	2.259808	5.252267	623.7	9.40518
LHM	3.2431408	3.447872	5.539649	559.5205	9.01298
LHH	3.3058334	3.381047	5.54248	583.1518	9.09625
MLL	2.84883604	3.065972	5.086821	448.9304	8.221573
MLM	2.7190393	3.321153	5.150526	497.9061	8.549226
MLH	2.84073237	3.467942	5.336888	492.513	8.674009
MML	3.59982702	2.102858	5.241208	644.9879	9.094553
MMM	3.44420574	3.410434	5.625284	630.0202	9.006775
MMH	3.42640372	3.317348	5.577264	612.6197	8.774985
MHL	3.44933091	2.036342	5.111455	642.1973	9.014694

Table A5-1, Treatment Averages Of Transformed Data, Continued.

MHM	3.47764689	3.11034	5.49593	683.5541	8.910635
MHH	3.48711502	3.181695	5.554162	693.3119	8.858674
HLL	2.82931291	3.505416	5.338849	523.7326	9.166496
HLM	2.82675299	3.566959	5.398908	537.3386	9.17174
HLH	2.66910954	3.394023	5.212062	493.2508	8.962131
HML	3.35392525	3.401689	5.577329	638.2453	8.984758
HMM	3.36865155	3.435886	5.603779	635.4891	9.080902
HMH	3.2287544	3.330518	5.458598	562.5959	8.638979
HHL	3.46888803	2.921539	5.411545	744.2469	8.885592
HHM	3.48838464	2.919401	5.430515	742.8339	8.718145
HHH	3.17432375	2.951413	5.217802	589.1031	8.527804

BIBLIOGRAPHY

BIBLIOGRAPHY

- Andrews, B. H. and Parsons, H. L., 1989, "L. L. Bean Chooses a Telephone Agent Scheduling System," Interfaces, Vol. 19, Nov-Dec 1989, pp. 1-9.
- Aggarwal, S. C., 1982, "A Focused Review of Scheduling in Services," European Journal of Operational Research, Vol. 9, Iss. 2, pp. 114-121.
- Bechtold, S. E. and Showalter, M. J., "Methodology for Labor Scheduling in a Service Operating System," Decision Sciences, Vol. 18, Iss. 1, pp. 89-107.
- Bechtold, S. E., Brusco, M. J., and Showalter, M. J., "A Comparative Evaluation of Labor Tour Scheduling Methods," Decision Sciences Journal, Vol 22, No. 4, pp. 683-699.
- Brady, R. M., "Optimization strategies gleaned from biological evolution," Nature, Vol. 317, 31 October 1985, pp. 804-806.
- Brusco, M. J., and Jacobs, L. W., "A Simulated Annealing Approach to the Cyclic Staff-Scheduling Problem," Naval Research Logistics, Vol 40, pp. 69-84.
- Davis, L. and Ritter, F., "Schedule Optimization with Probabilistic Search," Proceedings of the 3rd Annual IEEE Conference on Artificial Intelligence Applications, 1987, pp. 231-236.
- Davis, L. and Steenstrup, M., "Genetic Algorithms and Simulated Annealing: An Overview," In Davis (Ed.), Genetic algorithms and simulated annealing, London, Pitman, 1987, pp. 1-11.
- Glover, F., McMillan, C. and Glover, R., "A Heuristic Programming Approach to the Employee Scheduling Problem and Some Thoughts on 'Managerial Robots,'" Journal of Operations Management, Vol. 4, No. 2, pp. 113-128.
- Glover, F., and McMillan, C., "The General Employee Scheduling Problem: An Integration of MS and AI," Computers and Operations Research, Vol. 13, No. 5, pp. 563-573.
- Glover, F., "Tabu Search - Part 1," ORSA Journal on Computing, Vol 1, No 3, pp 190-206.

- Glover, F., "Tabu Search - Part 2," *ORSA Journal on Computing*, Vol 2, No 1, pp 4-32.
- Glover, F., Klingman D., and Phillips, N., "Netform Modeling and Applications," *Interfaces*, Vol 20, #4, July-August 1990, pp. 7-27.
- Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning, 1989, Reading, Mass., Addison Wesley.
- Henderson, W. E. and Berry, W. L., "Heuristic Methods for Telephone Operator Shift Scheduling: An Experimental Analysis," *Management Science*, Vol. 22, No. 12, pp. 1372-1380.
- Huang, M. D., Romeo, F., and Sangiovanni-Vincentelli, A. L., "An Efficient General Cooling Schedule for Simulated Annealing," *Proceedings IEEE Int. Conference on Computer Aided Design*, Santa Clara, November 1986, pp. 381-384.
- Kirk, R. E., Experimental Design, 1982, Brooks/Cole, Belmont California.
- Kirkpatrick, S., Gelatt, C. D. Jr., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, Number 4598, 13 May 1983, pp. 671-680.
- Knox, J. E., "The Application of Tabu Search to the Symmetric Traveling Salesman Problem," Unpublished Doctoral Dissertation, University of Colorado, 1989.
- Koelling, C. P. and Bailey, J. E., "A Multiple Criteria Decision Aid for Personnel Scheduling," *IIE Transactions*, Vol. 16, Iss, 4, pp. 299-307.
- Krajewski, L. J. and Ritzman, L. P., "Shift Scheduling in Banking Operations: A Case Application," *Interfaces*, Vol 10, No. 2, pp. 1-8.
- van Laarhoven, P. J. M. and Aarts, E. H. L., Simulated Annealing: Theory and Applications, 1987, Dordrecht, Kluwer.
- Li, C., Robinson, E. P., and Mabert, V. A., "An Evaluation of Tour Scheduling Heuristics with Differences in Employee Productivity and Cost," *Decision Sciences Journal*, Vol 22, No. 4, pp. 700-718.
- Loucks, J. S., and Jacobs, F. R., "Tour Scheduling and Task Assignment of a Heterogeneous Workforce: A Heuristic Approach," *Decision Sciences Journal*, Vol 22, No. 4, pp. 719-738.
- Lundy, M., and Mees, A., "Convergence of an Annealing Algorithm," *Math. Prog.*, Vol. 34, pp. 111-124.
- Mabert, V. A., "A Case Study of Encoder Shift Scheduling Under Uncertainty," *Management Science*, Vol. 25, Iss. 7, pp. 623-631.

- Mabert, V. A. and Raedels, A. R., "The Detail Scheduling of a Part-Time Workforce - A Case Study of Teller Staffing," Decision Sciences, Vol. 8, Iss. 1, pp. 109-120.
- Mabert, V. A. and Watts, C. A., "A Simulation Analysis of Tour Shift Construction Procedures," Management Science, Vol. 28, Iss. 5, pp.520-532.
- Metropolis, N., Rosenbluth, A., Rosenbluth, N., Teller, A., and Teller, E., "Equation of State Calculations by Fast Computing Machines", Journal of Chemical Physics, 21 (1953), pp. 1087-1092.
- McGinnis, L. F., Culver, W. D., and Deane, R. H., "One and Two Phase Heuristics for Workforce Scheduling," Computers and Industrial Engineering, Vol. 2, 1978, pp. 7-15.
- Miller, J. G., and Berry, W. L., "Heuristic Methods for Assigning Men to Machines: An Experimental Analysis," AIIE Transactions, Vol 6, No 2, pp 97-104.
- Montgomery, D. C., Design and Analysis of Experiments, 1984, John Wiley & Sons, New York, New York.
- Moondra, S. L., "An L.P. Model for Workforce Scheduling for Banks," Journal of Bank Research, Vol. 6, Iss. 4, pp. 299-301.
- Morris, J. G. and Showalter, M. J., "Simple Approaches to Shift, Days-Off and Tour Scheduling Problems," Management Science, Vol. 29, Iss. 8, pp. 942-950.
- Pedhazur, E. J., Multiple Regression in Behavioral Research, 1982, Holt, Rinehart, and Winston, Inc., New York, New York.
- Romeo, F., Sangiovanni-Vincentelli, A. L., and Sechen, C., "Research on Simulated Annealing at Berkely," Proc. IEEE Int. Conference on Computer Design, Port Chester, November 1984, pp. 652-657.
- Skorin-Kapov, Jadranka, "Tabu Search Applied to the Quadratic Assignment Problem," ORSA Journal on Computing, Vol 2, No 1, pp 33-45.
- Segal, M., "The Operator-Scheduling Problem: A Network Flow Approach," Operations Research, Vol. 22, No. 4, pp. 808-823.
- Sze, D. Y., "A Queueing Model for Telephone Operator Staffing," Operations Research, Vol. 32, Iss. 2, pp. 229-249.
- Tien, J. M. and Kamiyama, A., "On Manpower Scheduling Algorithms," SIAM Review, Vol 24, No. 3, pp. 275-287.
- Vakharia, A. J., and Chang, Y., "A Simulated Annealing Approach to Scheduling a Manufacturing Cell," Naval Research Logistics, Vol. 37, pp. 559-577.

White, S. R., "Concepts of Scale in Simulated Annealing," Proc. IEEE Int. Conference on Computer Design, Port Chester, November 1984, pp. 646-651.

MICHIGAN STATE UNIV. LIBRARIES



31293009087853