T-4632

A COMPUTER-ASSISTED TUTORIAL FOR IMPLEMENTING THE WATSON-WOOLSEY METHOD FOR SHIFT SCHEDULING

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

by

Frederick Douglas Jackson

ProQuest Number: 10784038

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10784038

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

A thesis submitted to the Faculty and Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematical and Computer Sciences).

Golden, Colorado

Date 11/11/94

Signed:

Frederick Douglas Jackson

Approved:

Dr. R.E.D. Woolsey
Thesis Advisor

Golden, Colorado

Date 11/11/94

Dr. Graeme Fairweather
Professor and Head
Department of Mathematical
and Computer Sciences

Abstract

This thesis provides a computer assisted tutorial for solving shift scheduling problems using the Watson-Woolsey Method for Shift Scheduling. The Watson-Woolsey Method for Shift Scheduling is a heuristic defined by James L. Watson Jr. in his Master's Thesis "Efficient Solution of Shift Scheduling Problems by Determining the Minimal Sets of Constraints and Variables" written in 1991 for the Colorado School of Mines.

A shift scheduling problem is easily formulated as an Integer Programming(IP) problem. The Watson-Woolsey Method (WWM) reduces that IP problem to its smallest equivalent. The Method involves removing redundant and/or dominated constraints. The computer assisted tutorial will instruct the user in the application of the WWM. The software will analyze the problem to determine which of two heuristics to follow, equal or unequal costs; display the rules of the WWM; verify user's application of the rules and provides hints when the rules are incorrectly applied. Several sample problems will be provided, they will contain all possible combinations of the rules.

The software is written in Microsoft Visual Basic and designed to run under Microsoft Windows 3.1 on an IBM compatible personal computer.

Table of Contents

Table of Figures

Table of Tables

Acknowledgments

Several people have helped to make this thesis a reality. Mentioning them all here would, inevitably, lead to leaving someone out. However there are some who I must single out as extraordinary contributors.

First and foremost are my parents, Ruth and Fred Jackson, who have supported my decision to return to college, both financially and emotionally. They have provided a steadying influence in my life for as long as I can remember, something for which I will be forever grateful.

Second are the rest of my family, who have always been there when I was in need. Specifically my sister, Shari, who was attending CSM and reminded me that Dr. Woolsey was here.

The members of my committee, Dr. Barbara Bath and Dr. Ruth Maurer, deserve a great deal of credit for the quality of this document. They required me to be specific and thorough in my explanations. Additionally, they were there to answer questions and lend support over a very tumultuous couple of years. They have contributed considerably to my academic maturity.

I want to acknowledge Professor Bill Astle for his help in overcoming my statistical nightmares. He has made a difficult subject understandable.

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

I want to make it clear that "the guild", and all those who have populated it over the years, have made this task simpler by just being there. Although I cannot say they were instrumental this task, I will say that having them around has added greatly to this particular experience.

Finally, I want to thank Dr. Gene Woolsey for being exactly who he is, "The single greatest outlier in the OR profession." I would not have come to CSM if he were not here. Every conversation with him is a class session and I have never failed to learn something new. I feel honored to have worked with him and whenever anyone says "He's one of Woolsey's students." I feel honored to have been mentioned in the same sentence.

Thank you one and all.

Introduction

General Problem

Shift scheduling problems are common in various types of business; from grocery stores to the phone company. Every type of personnel needs to be scheduled during the day and different time periods may require different amounts of personnel. The problem is twofold:

1. Worker requirements are generated for each period; these vary by time of day.
2. Workers are scheduled in shifts that may vary in length.

A general mathematical formulation for this type of problem is given below:

Minimize: $$TotalCost = \sum_{j=1}^{n} C_j \cdot X_j$$

Subject To: $$\sum_{j=1}^{n} A_{i,j} \cdot X_j \geq RHS_i, \text{ where } i=1,2,3,\ldots,M$$

Where:

C_j - Cost of one person working on shift j

X_j - Number of people working on shift j

$A_{i,j}$ - Equal 1 - shift j is on duty during workperiod i
Equal 0 - shift j is not on duty during workperiod i

RHS_i - Personnel requirement for workperiod i

M - Number of workperiods

This formulation allows for different costs and a varying coverage during each shift. A typical 9 to 5 shift will fit in this formulation; additionally, part-time and graveyard shifts will also fit.

The real problem with scheduling personnel is inherent with anything dealing with people. The solution must be in integer values. No one has come up with a way to have a third of a person on duty and, mathematically, this makes what would be a simple problem into something that is extremely difficult.

A typical solution to this problem is to drop the above formulation into any commercial Linear Programming(LP) code and then truncate the answers to the nearest integer. While this seems like a logical solution, in actuality some assumptions form the basis for this solution. The most important is that the feasible region for the problem includes your answer. Woolsey[25] contains two examples of why you must check the feasibility of your solution, this discussion contains: a washing machine problem that is solved using LP when the integer problem has no feasible solution and a US Air Force problem where they used LP and discovered the feasible solution was considerably different from their LP solution.

Another possible solution is to use an Integer Programming (IP) code and let the computer run until solved. The examples used in this tutorial have as many as ten shifts and forty workperiods, this is a large IP problem. Entering the data into the computer and verifying it was correctly input required about fifteen minutes per problem. Solving them using an IP code takes a few minutes. However, using the WWM, I reduced these problems and solved them by hand in less than five minutes per problem. Should the problem not reduce to an easily solvable set of equations, the reduced problem will be faster to enter into a computer to find your answer.

The IP algorithms for optimal solutions come in several varieties:

Branch and Bound - This literally computes the value of the objective function for every feasible value for each variable and gives you the optimal answer. The number of calculations involved could be huge. For example assume that each shift can have up to ten people, then each variable in the equation has a possible value from zero to ten. With ten shifts the number of possible combinations is 11^{10}.

Gomory - This can be much more efficient than the Branch and Bound method. The problem is that this algorithm functions in dual space and therefore is super-optimal and infeasible until it converges to the optimal answer. In other words, you do not have a usable primal solution until the algorithm is complete.

Watson-Woolsey Method

The WWM reduces the original problem down to the smallest equivalent problem. First the method eliminates redundant or non-binding constraints. A redundant constraint is one which is the duplicate of another, a non-binding constraint is one which never comes into play because another constraint is tighter. Second, the method eliminates variables in the primal problem; these are essentially redundant or non-binding constraints in the equivalent dual formulation. The dual formulation for this problem is as follows:

Maximize: $Total = \sum_{i=1}^{M} RHS_j \cdot Y_j$

Subject To: $\sum_{i=1}^{M} A_{j,i} \cdot Y_j \leq C_j$, where j=1,2,3,...,n

Where:

Total - Value of workforce to company

C_j - Cost of one person working on shift j

Y_j - Value (to employer) of one worker covering work period i

$A_{j,i}$ - Equal 1 - shift j is on duty during work period i

Equal 0 - shift j is not on duty during work period i

RHS_i - Personnel requirement for work period i

M - Number of workperiods

It is important to note the similarities between the Primal, or original problem, and the dual problem. The coefficient matrices are simply transposed, the RHS in each constraint becomes the objective function coefficient and the objective function coefficients become the RHS's in the dual. If the primal is a minimization then it's dual is a maximization problem. Similarly, the greater than constraints become less than constraints in the dual. Duality theory is weaker in integer programming than in linear programming due to an obvious lack of continuity in the linear variables. To emphasize this point it should be mentioned that Egon Balas, in a paper given at the Princeton Symposium on

Mathematical Programming [3], showed that for integer programming, the dual of the dual is not the primal, as is the case in linear programming. Therefore one should be particularly cautious in economic interpretation of either mixed-integer, all integer, or zero-one integer programming problems. To quote from the above paper, page 191, Egon Balas wrote :"If, however, the primal is a pure zero-one program, there is no use at all in going to the dual." Also, in a private communication to Dr. Woolsey [28], Dr. Balas pointed out that if a shift scheduling problem had "all one's sequentially" for each shift, (i.e. no "gaps" allowed in the sequence of one's) that the shift scheduling problem could be formulated as an out-of-kilter network flow problem. Unfortunately, for the problems of this thesis, this circumstance does not hold. There exists a one to one correspondence between variables in the primal formulation and constraints in the dual formulation. Therefore, when we eliminate constraints in the primal we are eliminating variables in the dual.

There are two groups of rules, row and column. As the grouping implies the first group applies to the rows or workperiods and the second applies to the columns or shifts. We are trying to minimize the cost of people working a number of different shifts over time periods with different demands for people. We start with a worksheet that shows work periods at a given time (rows) and shifts (columns) with ones put into time periods that the shift is on duty. Further the demand for people in each time period is given as the right hand side (RHS) in each row. There is a cost per person on each shift that can be different from shift to shift.

The key to the WWM is in locating similar or different left hand sides (LHS) and right hand sides (RHS). Each rule begins with a title in capital letters, e.g. SAME-SAME. Explicit justification for the rules are not presented here as they are explained in detail in Watson and Woolsey's papers in [22,23,24].

This indicates that the user should look for the rows containing LHS that are the same and RHS that are the same. The RHS is just one number and determining if the two rows have the same RHS is self explanatory. The LHS contains a different coefficient for each shift, compare these coefficients one at a time and verify that they are the same. Here are the rules:

SAME-SAME:

ELIMINATE ALL SUCH ROWS BUT ONE.

SAME-DIFFERENT:

IF RHS1>RHS2 then ELIMINATE ROW 2, otherwise ELIMINATE ROW 1.

DIFFERENT-SAME:

IF row 1 has ones everywhere that row 2 has ones,

BUT row 1 has one or more additional ones

ELIMINATE row 1.

DIFFERENT-DIFFERENT:

IF row 1 has ones everywhere that row 2 has ones,

BUT row 1 has one or more additional ones AND

IF RHS1<RHS2, then

ELIMINATE row 1.

The column rules are similar to those for the rows. Some exceptions occur in terminology and due to choosing the column with the smallest cost coefficients. e.g. SAME-SAME. This indicates that the user should look for the columns containing the same shift coverage(LHS), 1's and 0's in same rows, and a cost coefficient(RHS) that is the same. The cost coefficient is just one number and determining if the two columns have the same RHS is self explanatory. The LHS or shift coverage contains a different coefficient for each time period, compare these coefficients one at a time and verify that they are the same. Here are the rules:

COLUMN SAME-SAME:

ELIMINATE all such columns but one.

COLUMN SAME-DIFFERENT:

ELIMINATE the more costly column (shift).

COLUMN DIFFERENT-SAME:

IF column (shift) 1 has ones in the same rows that column (shift) 2 has ones,

BUT column (shift) 1 has one or more rows with additional ones

ELIMINATE column (shift) 2.

COLUMN DIFFERENT-DIFFERENT:

IF column (shift) 1 has ones everywhere that column (shift) 2 has ones ,

BUT column (shift) 1 has one or more rows with additional ones, AND

IF COST1<COST2 ELIMINATE column (shift) 2.

If the current worksheet has not been altered by this pass through the rules, stop and get the best answer you can for the worksheet that's left. Otherwise return to RULE 1.

Computer Aided Instruction

This tutorial is intended to bring this simple method to anyone that wants to learn. The method is taught to students in Integer Programming at CSM. While this seems an ample audience, the method could greatly simplify the tasks of many schedulers. Anyone can read the algorithm and many will understand how to put it to good use; however, reading how to do something is not the same as doing it. This tutorial will instruct its user in application of the Watson-Woolsey Method, step by step, it will point out mistakes and provide hints to the next application of a rule. It is hoped that this will be of use to anyone that needs a little hands on experience to master the method.

Background

Shift Scheduling is a major topic in the various Operations Research publications. Problems vary from scheduling telephone operators to harbor pilots [11, 26]. The solutions vary from complex detailed computer solutions to simple heuristics. All the problems require integer solutions, yet, the objectives vary: e.g. minimizing assignment costs, maintain adequate service levels, and maximizing a harbor pilot's time with their families [26]. The Watson-Woolsey Method is described in a Master's Thesis by James L. Watson Jr. [22] for the Colorado School of Mines. The method of reducing the shift scheduling problem described here appears unique. A computer assisted tutorial on the WWM has not been discovered.

Mr. Watson cited several sources, some of which I will reiterate here. Thompson, Tonge, and Zionts [20] demonstrated a concept for reducing the original problem by eliminating redundant and/or non-binding constraints. They focused on LP problems, which have continuous linear constraints, and applied a graphical approach. This limited the technique to a maximum of three variables because of the limitations of graphically depicting more than three dimensions.

Glover and Woolsey [12] demonstrated a surrogation technique for IP constraints. The idea is to combine all the constraints into one "strong" constraint. The method is useful for IP problems where the variables are restricted to values of zero and one.

Bartholdi [4] analyzed problems that have cyclical shift schedules and similar requirements. The technique introduces a round-off algorithm to a LP solution. This method allows calculation of an upper bound of the difference between the optimal integer solution and the solution provided by the round-off algorithm. Knowledge of the optimal integer solution is not required for this calculation.

Scheduling problems tend to be specific to each application. The optimization criteria vary according to who or what is determining the schedule. Nurse scheduling is aimed at optimizing the satisfaction of the nurse. The cost is relatively constant due to staffing requirements. When scheduling Harbor Pilots [26] we want to optimize Off-duty times. Pay is essentially constant and a "fair" schedule is important. Wermus and Pope state that "The algorithm must be simple and explainable to people with no college education" which describes the WWM perfectly.

Solutions to scheduling problems vary from LP solutions [4] to simple heuristics [8, 26] to complicated mathematical formulations [15] which combine IP, LP and network optimization to find a solution.

Constraint removal is discussed in Reklaitis, Ravindran, and Ragsdell [19], but, is limited to removing equality constraints by substitution.

The WWM's generic problem is easily modified to fit a specific application and the algorithm is simple to use, once the technique is learned. This tutorial is intended to help in that learning process.

User's Guide

Overview

This tutorial provides an interactive environment for learning the WWM. The WWM is a simple heuristic that reduces a constrained integer optimization problem to the smallest set of variables and their binding constraints. The WWM is easy to implement once you understand the rules and their usage. This tutorial is intended to enhance the user's knowledge of the WWM.

Several sample files are provided and they have been designed to provide examples of all the rules in the WWM.

Getting Started

Installing WW-Tutorial:

1. Insert the WW-Tutorial Setup disk in drive, determine if this is drive A: or B:.
2. In the Windows Program Manager, choose Run from the File menu.

   In the Command Line box, type a:\setup or b:\setup and then choose the OK button. Follow the instructions on the screen

You are now ready to run the tutorial. Just double click on the WW-Tutorial icon, and begin. Spend a little time familiarizing yourself with the form, see "Figure 1 - Sample Display", page 13.

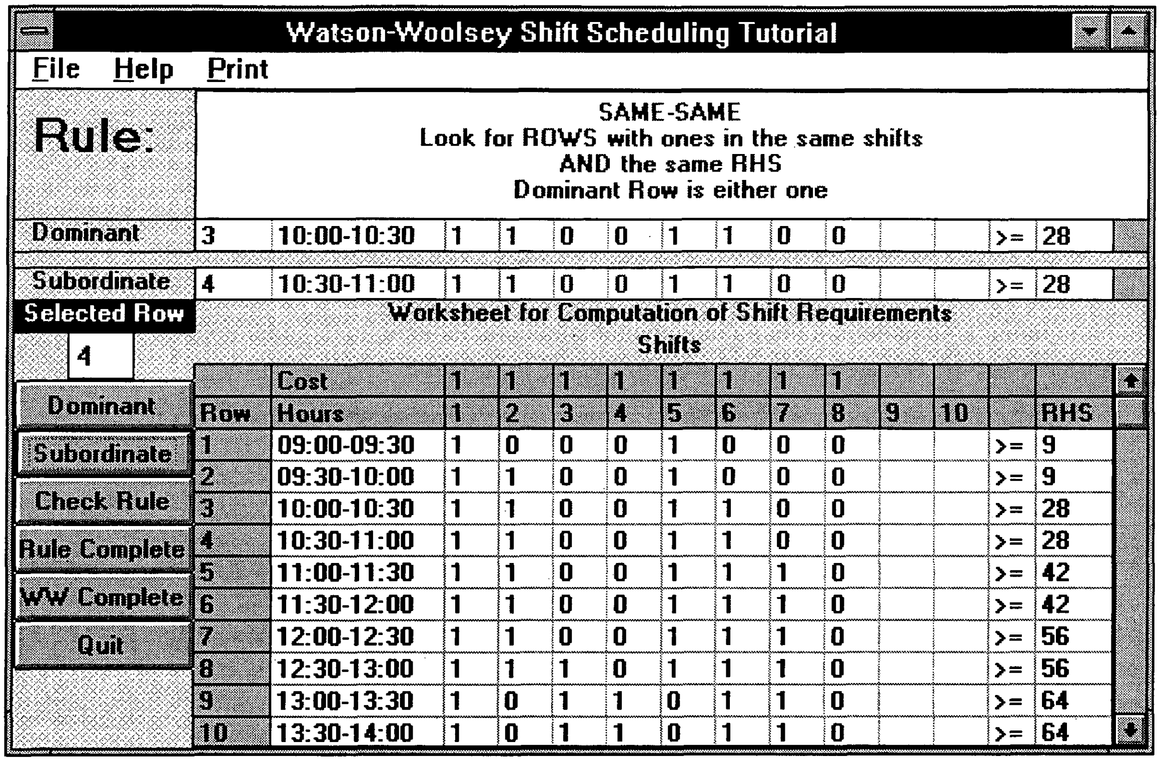


Figure 1 - Sample Display

- The current rule is displayed in a box at the top.
- A grid similar to a spreadsheet is the most prominent feature; this will display the worksheet while you work. The display will vary depending upon which rules are currently in use. Row rules will look as they are at startup, column rules are transposed to display an entire column on screen; they appear as rows of the

spreadsheet. A scroll bar appears on the right of the grid when there are rows below the bottom of the window.

- Below the rule are two single row grids that will contain your choices of Dominant and Subordinate rows.
- On the left side is a box labeled Selected Row, this is where the currently selected row or shift is displayed. Additionally, you may enter a row number. Please note that when the column rules are displayed the label for this box will change to Selected Shift, therefore it will be referred to as the Selected Row/Shift box.
- Below the Selected Row/Shift box are several command buttons, click to perform the following:

  Dominant - display selected row/shift in the Dominant grid

  Subordinate - display selected row/shift in the Subordinate grid

  Check Rule - verify correct application of rule

  Rule Complete - verify no remaining rows fit current rule

  WW Complete - verify that Method is complete

  Quit - terminate the tutorial

Starting the WW-Tutorial:

To begin the tutorial you need to open a sample worksheet. To do this, click on the File menu (similar to what you did in the Windows Program Manager), then choose

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

Open from this menu. A list of files will appear. The example that is shown in this thesis is located in "EXAMPLE.WWW" (all worksheet extensions are .WWW for Watson-Woolsey Worksheet), this file contains at least one application for each rule.

The remaining task is to choose rows based upon the displayed rule. There are two ways to choose a row:

- Place the mouse pointer over the row in the grid and click, your selection will appear in the Selected Row/Shift display area.
- Place the mouse pointer over the Selected Row/Shift display area and click, then you may enter the row/shift number from the first column of the grid.

Step-by-Step Description

A typical tutorial session will follow these steps:

I Startup WWM-Tutorial.

II Choose a sample shift scheduling problem from the File menu.

III For each rule displayed repeat the following steps.

1 - Rule X is displayed.

2 - Select a row.

3 - Click -Dominant- command button.

4 - Select row.

5 - Click -Subordinate- command button.

6 - Verify that your selections meet Rule X criterion.

7 - Click -Check Rule- command button.

8 - The tutorial will display a message:

Correct Answer

or - Incorrect Answer with a hint to problem.

9 - Return to beginning of Step III or continue to Step IV.

IV Click -Rule Complete -

1 - Correct Answer and advance to next Rule, go to Step III.

2 - or - Incorrect Answer with a hint to rows matching Rule X.

V Click -WW Complete -

1 - Correct Answer and wait for user to choose next action.

2 - or - Incorrect Answer with a hint.

The Rules

In Mr. Watson's thesis the rules were divided into EQUAL and UNEQUAL shift costs. While this does simplify the rules, it creates the impression that the task is considerably different depending upon your shift costs. This is not true, the only difference occurs in the column rules. For the EQUAL shift cost problems you can ignore the two rules (six and eight) that consider DIFFERENT shift costs.

WATSON-WOOLSEY METHOD FOR SHIFT SCHEDULING

This is a simple method to eyeball shift scheduling problems. We assume that you are trying to minimize the cost of people working a number of different shifts over time periods with different demands for people. We start with a worksheet that shows work periods at a given time (rows) and shifts (columns) with ones put into time periods that the shift is on duty. The demand for people in each time period is given as the right hand side (RHS) in each row. There is a cost per person on each shift that can be different from shift to shift.

Rule 1: **SAME-SAME**

Look for ROWS with ones in the same shifts

AND the same RHS

Dominant Row is either one

This rule eliminates identical time periods. If the constraints are identical it is simply a matter of removing multiple copies.

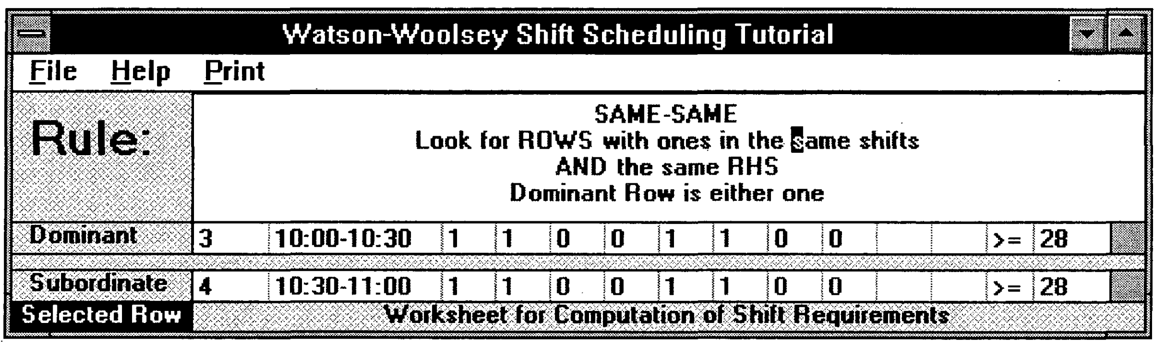


Figure 2 - SAME-SAME Example

Rule 2: SAME-DIFFERENT

Look for ROWS with ones in the same shifts

AND the different RHS

Dominant Row is row with largest RHS

This rule compares two time periods with identical shift coverage. Whatever shifts are chosen must cover the larger requirement, therefore the time period with the smaller requirement is dominated.

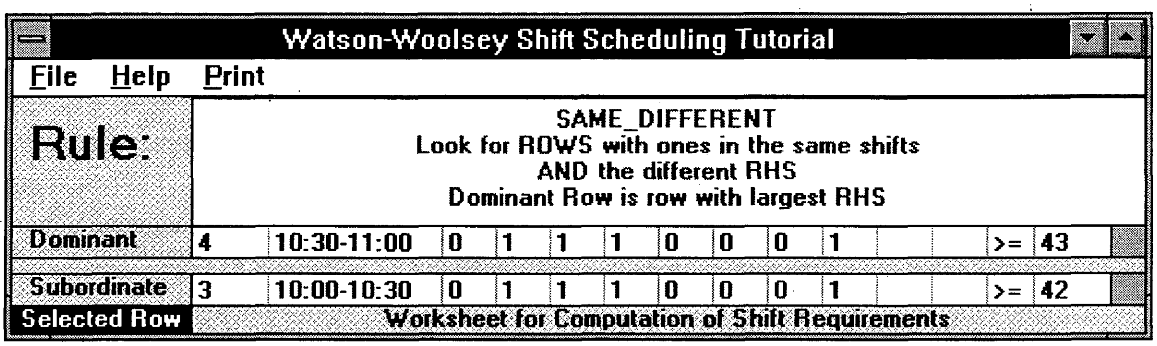


Figure 3 - SAME-DIFFERENT Example

Rule 3: DIFFERENT-SAME

Look for ROWS with the same RHS

if row 1 has ones everywhere as row 2, but, row 1 has additional ones

Dominant Row is row 2

This rule compares two time periods with identical requirements. Compare the two time periods shift by shift, if row 1 has the same shifts as row 2, except row 1 has additional shifts covering this time period. Then covering row 2 meets the requirements for row 1. Therefore, row 1 is dominated by row 2.

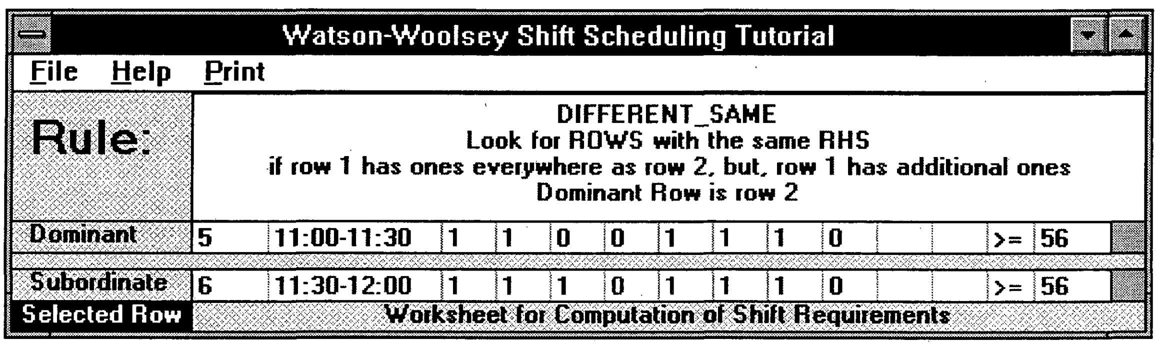


Figure 4 - DIFFERENT-SAME Example

Rule 4: DIFFERENT-DIFFERENT

Look for ROWS with different RHS's

if row 1 has ones everywhere as row 2, but, row 1 has additional ones

if RHS 1 < RHS 2 then row 2 is dominant

This rule compares two time periods with different shift coverage and different requirements. Compare the two time periods shift by shift, if row 1 has the same shifts as row 2, except row 1 has additional shifts covering this time period. And the requirements for row 1 are less than the requirements for row 2. Then covering row 2 meets the requirements for row 1. Therefore, row 1 is dominated by row 2.

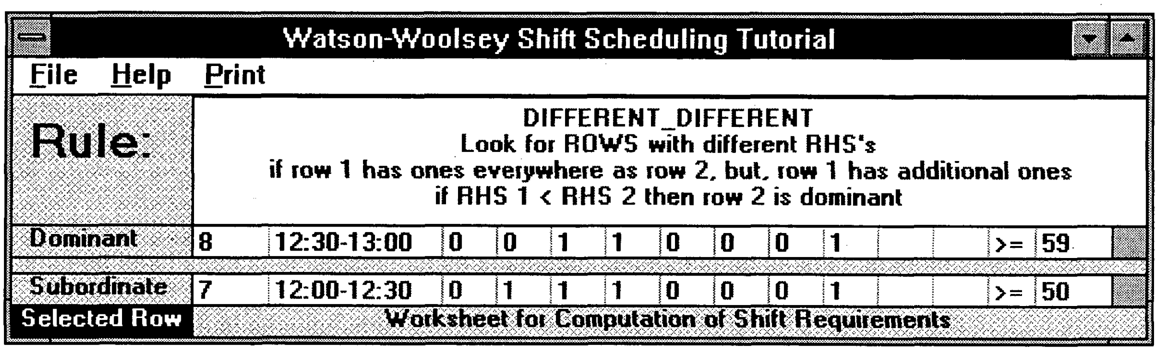


Figure 5 - DIFFERENT-DIFFERENT Example

Beginning of Column Rules

Next, we come to the column rules. A modification was necessary to allow the display of an entire column on the display screen. Figure 6 shows an example of Rule 5, the first of the column rules. The Dominant and Subordinate rows are actually the transposed columns of the coefficient matrix. On the left is the shift number followed by the zeroes and ones indicating shift coverage over the remaining work periods. On the right, where the RHS was for the previous rules, is the shift cost.

Rule 5: COLUMN SAME-SAME

Look for Shifts with ones in the same columns

AND the same RHS

Dominant Shift is either one

This rule eliminates identical shifts. If the shifts offer identical shift coverage, it is simply a matter of removing multiple copies.

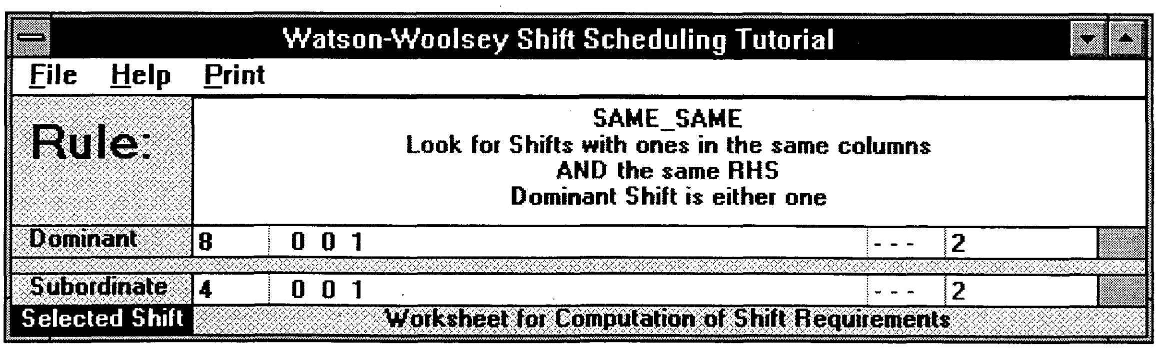


Figure 6 - COLUMN SAME-SAME Example

Rule 6: COLUMN SAME-DIFFERENT

Look for Shifts with ones in the same time periods

AND different Shift Costs

Dominant Shift is shift with smallest Shift Cost.

(The shift cost is on the right, where the RHS was for the Row rules.)

This rule compares two shifts with identical coverage. Whatever shifts have the lowest cost are chosen. NOTE: With EQUAL shift costs no two shifts can have different costs, this eliminates this rule from consideration.

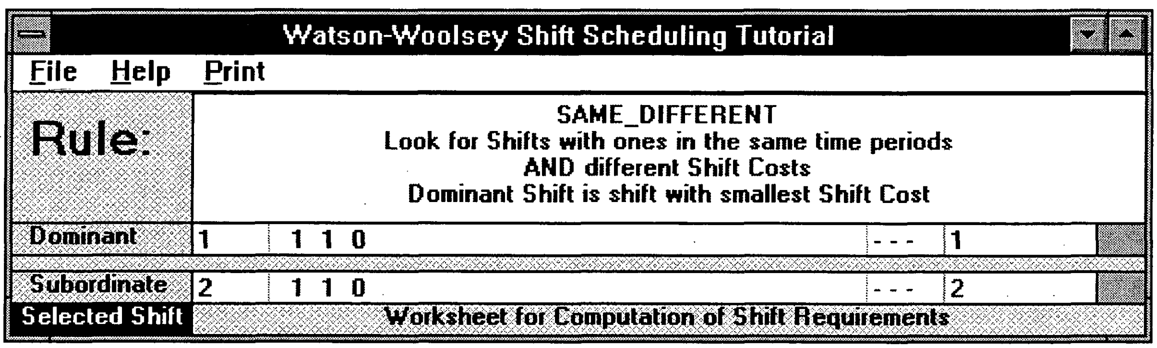


Figure 7 - COLUMN SAME-DIFFERENT Example

Rule 7: COLUMN DIFFERENT-SAME

Look for Shifts with the same Shift Cost

if shift 1 has ones everywhere as shift 2, but, shift 1 has additional ones

Dominant Shift is shift 1

(The shift cost is on the right, where the RHS was for the Row rules.)

This rule compares two shifts with identical shift costs. Compare the two shifts coverage, if row 1 has the same coverage as row 2, except row 1 covers additional time periods. Then shift 1 meets the requirements for shift 2 and costs the same amount. Therefore, row 2 is dominated by row 1.

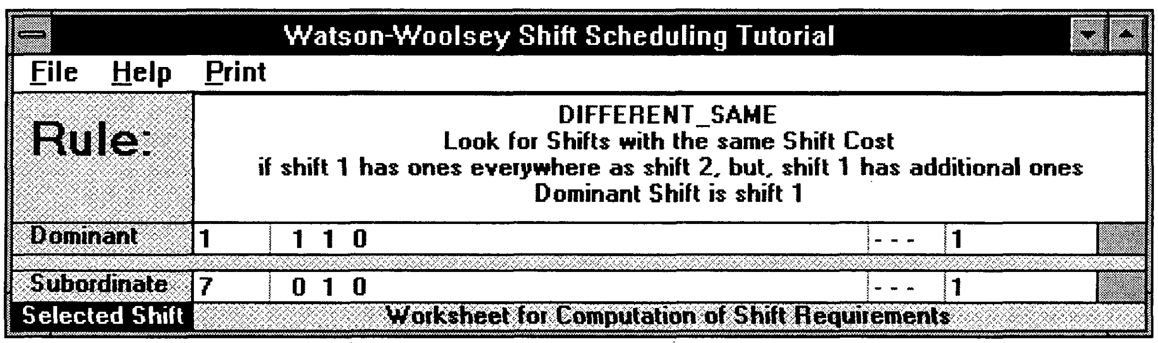


Figure 8 - COLUMN DIFFERENT-SAME Example

Rule 8: COLUMN DIFFERENT-DIFFERENT

Look for COLUMNS, which are transposed and displayed as rows, with

the same Shift Cost

if shift 1 has ones everywhere as shift 2, but, shift 1 has additional ones

if Shift Cost for 1 < Shift Cost for 2 then shift 1 is dominant

(The shift cost is on the right, where the RHS was for the Row rules.)

This rule compares two shifts with different coverage and different shift costs. Compare the two shifts coverage, if row 1 has the same coverage as row 2, except row 1 covers additional time periods. And shift 1 has a lower cost than shift 2. Then shift 1 meets the requirements for shift 2 and costs less. Therefore, row 2 is dominated by row 1. NOTE: With EQUAL shift costs no two shifts can have different costs, this eliminates this rule from consideration.

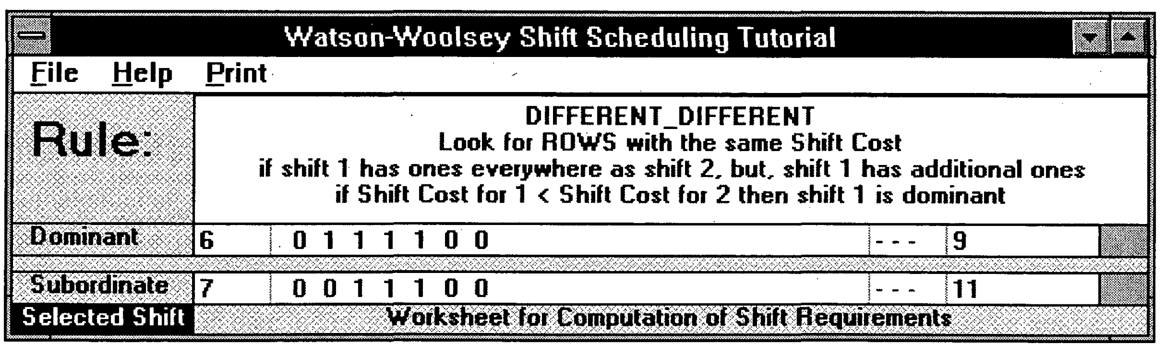


Figure 9 - COLUMN DIFFERENT-DIFFERENT Example

If the current worksheet has not been altered by this pass through the rules, stop and get the best answer you can for the worksheet that's left. Otherwise return to RULE 1.

An Example

The WWM-Tutorial is designed to be simple to use. The following example will walk through the "EXAMPLE.WWW" sample file. You should start the tutorial and work through this example as you read. Some steps will be left out, the example file includes at least one implementation for each of the eight rules. This text will not include every step required to complete the WWM, the tutorial should provide any necessary assistance required to complete this sample file.

The initial screen of the WW-Tutorial appears below in "Figure 10 - Opening screen". To begin the tutorial you load a shift scheduling problem file. Just click on the "File" command and a menu will be displayed, choose "Open" from this menu. A dialog box will be displayed showing the available data files. Choose "EXAMPLE.WWW" and press RETURN or click the OK button.

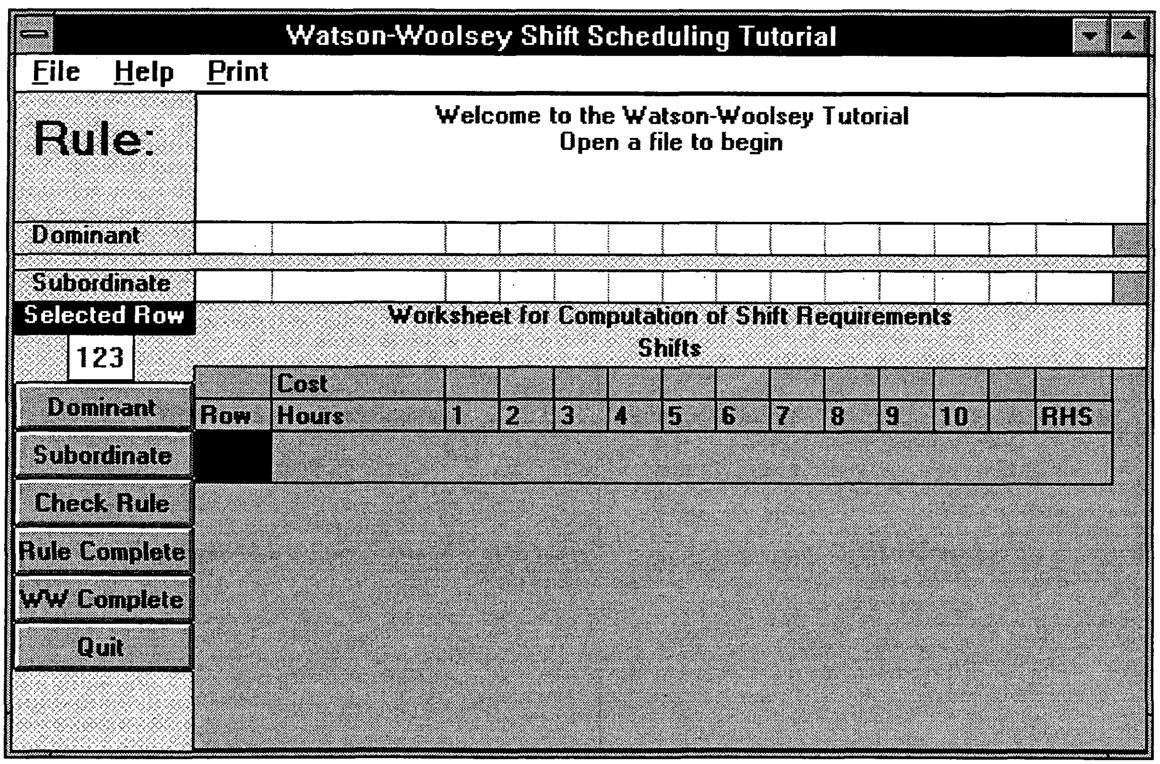
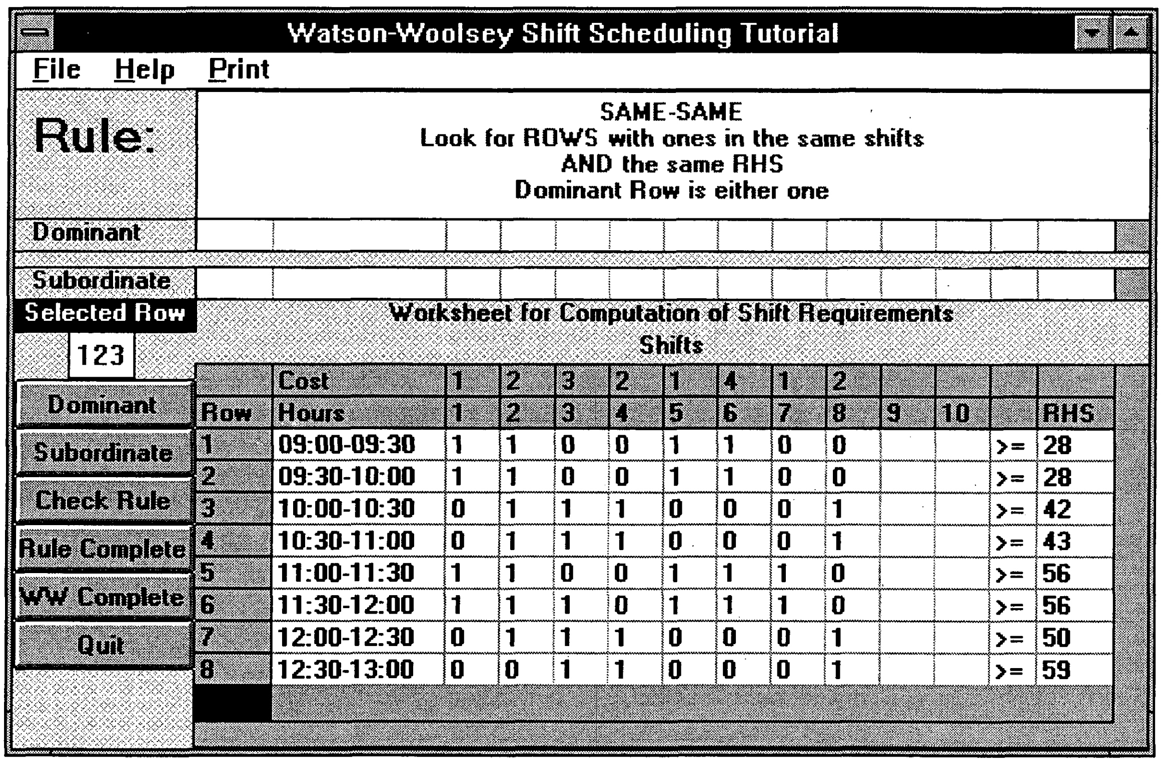


Figure 10 - Opening screen

The WW-Tutorial will load and display the example problem, as shown in "Figure 11 - After loading "EXAMPLE.WWW" data file", below.



	Cost	1	2	3	2	1	4	1	2				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	1	0	0	1	1	0	0			>=	28
2	09:30-10:00	1	1	0	0	1	1	0	0			>=	28
3	10:00-10:30	0	1	1	1	0	0	0	1			>=	42
4	10:30-11:00	0	1	1	1	0	0	0	1			>=	43
5	11:00-11:30	1	1	0	0	1	1	1	0			>=	56
6	11:30-12:00	1	1	1	0	1	1	1	0			>=	56
7	12:00-12:30	0	1	1	1	0	0	0	1			>=	50
8	12:30-13:00	0	0	1	1	0	0	0	1			>=	59

Figure 11 - After loading "EXAMPLE.WWW" data file

Rows one and two meet the criteria for Rule 1 - SAME-SAME, to select these rows place the arrow over row 1 in the grid (or click on the box under "Selected Row" and enter "1") then click on the Dominant button. Row 1 will appear in the Dominant row display area. Repeat this with row 2 as the subordinate row. "1", below, shows the screen at this point.

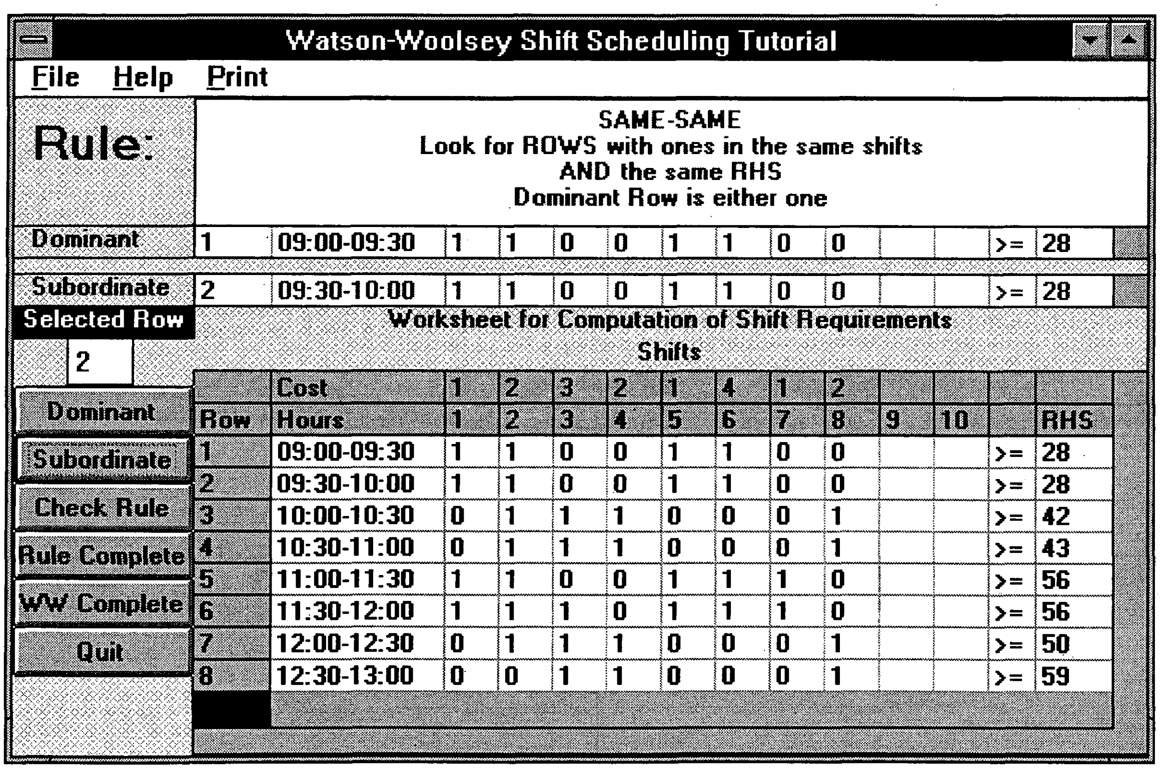


Figure 12 - After selection of Dominant and Subordinate Rows meeting Rule 1

"Figure 13 - Display showing response to correct selections", below, shows the screen after clicking the "Check Rule" button. "Correct Answer . . ." is displayed momentarily in the display area. Then the row is deleted and the rule is displayed.

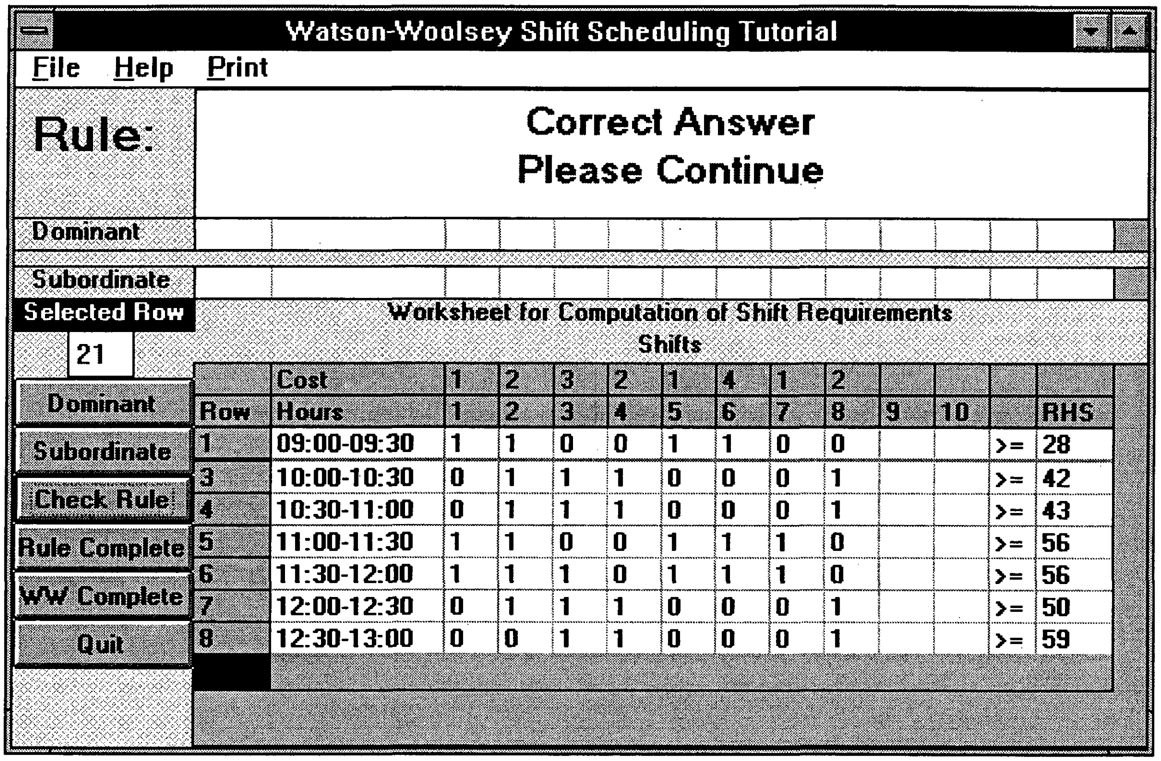
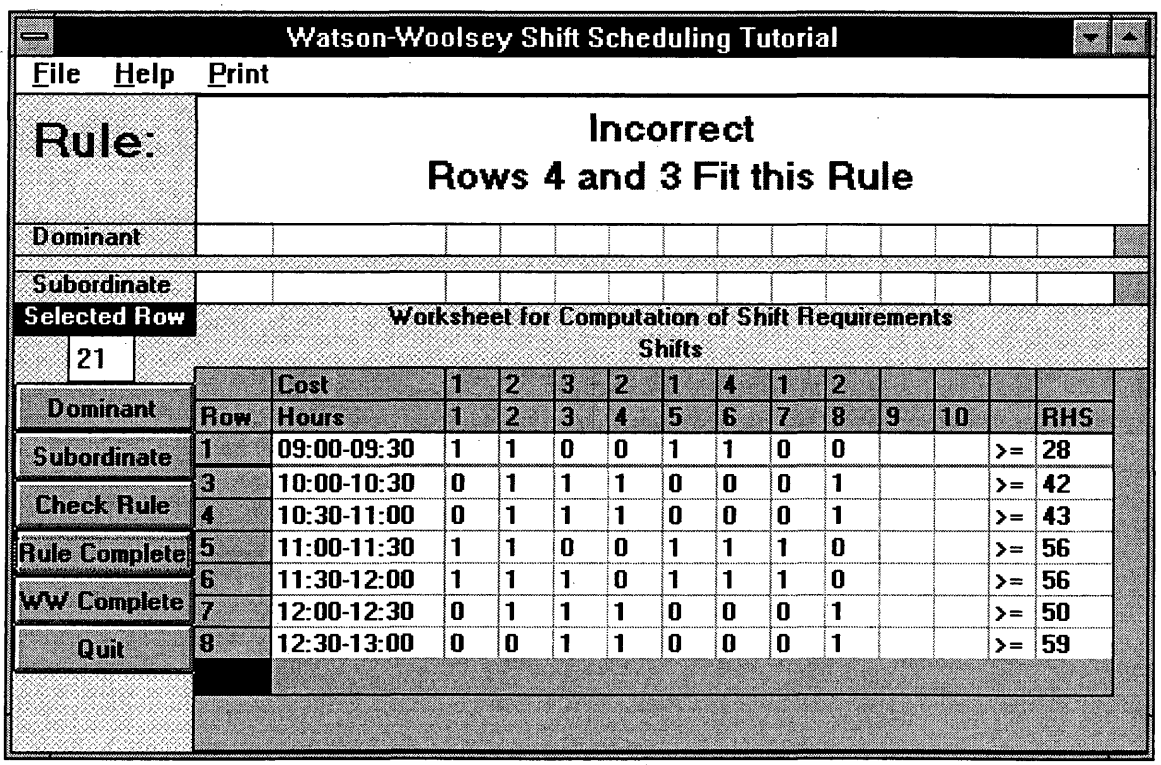


Figure 13 - Display showing response to correct selections

Click on the "Rule Complete" button and the display will be as shown in "Figure 14 - Display after selecting Rule Complete before Rule is done", below. This is where the hints for incomplete rule application are displayed, in this case the Dominant Row is number 4 and the Subordinate Row is number 3.



	Cost	1	2	3	2	1	4	1	2				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	1	0	0	1	1	0	0			>=	28
3	10:00-10:30	0	1	1	1	0	0	0	1			>=	42
4	10:30-11:00	0	1	1	1	0	0	0	1			>=	43
5	11:00-11:30	1	1	0	0	1	1	1	0			>=	56
6	11:30-12:00	1	1	1	0	1	1	1	0			>=	56
7	12:00-12:30	0	1	1	1	0	0	0	1			>=	50
8	12:30-13:00	0	0	1	1	0	0	0	1			>=	59

Figure 14 - Display after selecting Rule Complete before Rule is done

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

"Figure 15 - Display after choosing Dominant and Subordinate rows incorrectly", below, is an example of the display shown for an incorrect rule application. In this example the RHS's do not meet the rule criteria. The tutorial checks the RHS first and then checks each column from left to right to verify a correct application of a rule.

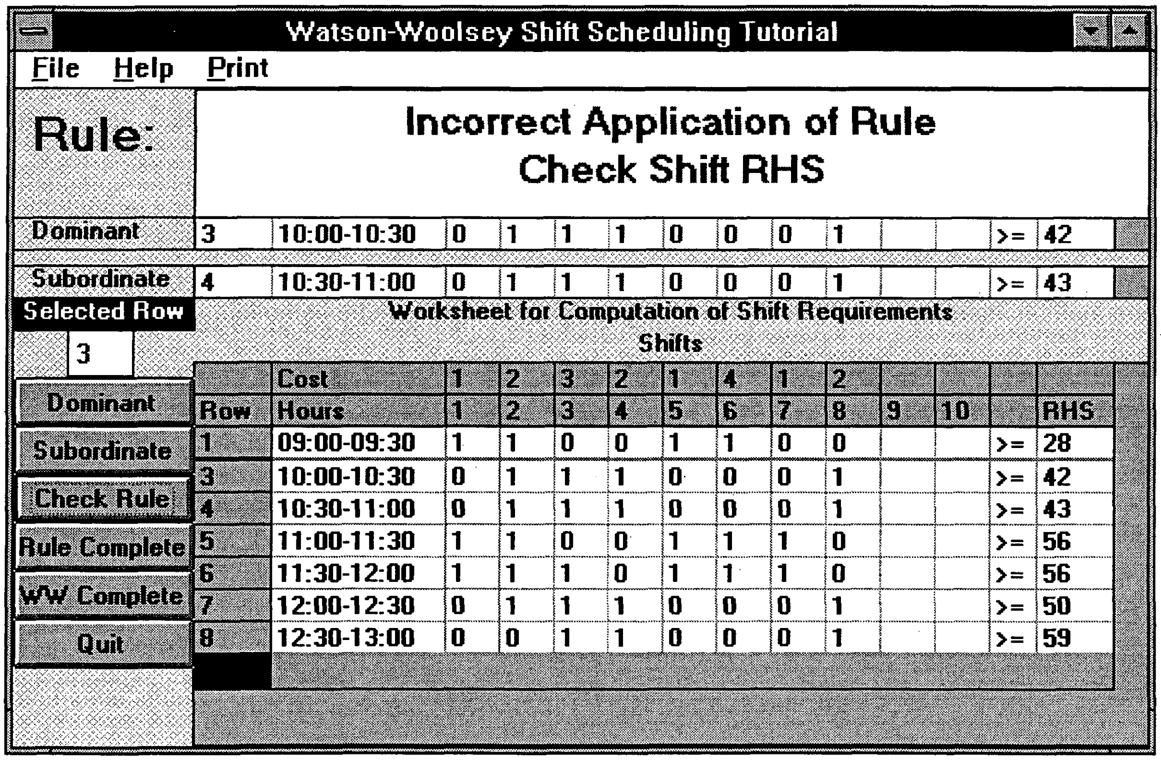


Figure 15 - Display after choosing Dominant and Subordinate rows incorrectly

In "Figure 16 - Display during Rule 4", page 34, the last row rule is displayed. This is the last screen shown prior to going into the column display format. Row 8 will be deleted when you click the "Check Rule" button, then after clicking "Rule Complete" the display will change as pictured in "Figure 17 - Display for Rule 5 (the first Column Rule)", page 35. Please compare these and verify that the information is the same, just shown after transposing the coefficient matrix.

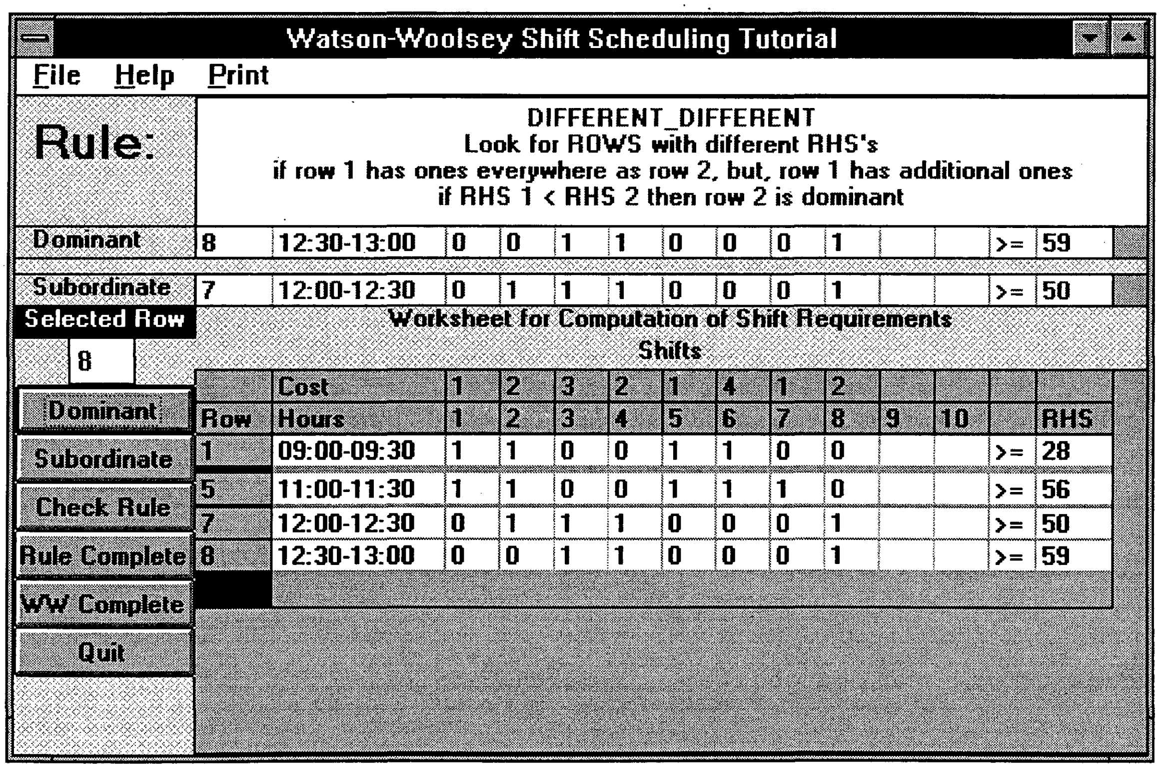


Figure 16 - Display during Rule 4

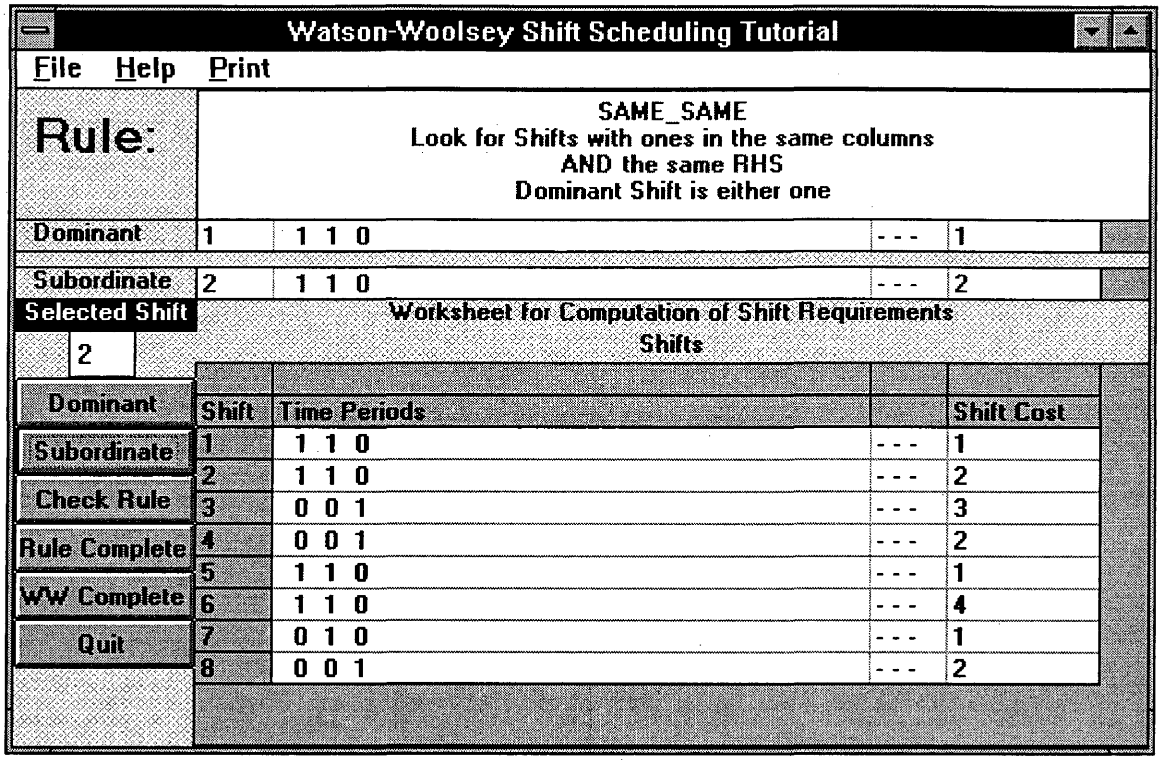


Figure 17 - Display for Rule 5 (the first Column Rule)

"Figure 18 - All Column rule eliminations complete", below, shows the display at the end of the column eliminations. Your display will appear like this after rule 7 - DIFFERENT-SAME is completed. This example uses equal shift costs therefore Rule 8 is not applicable.

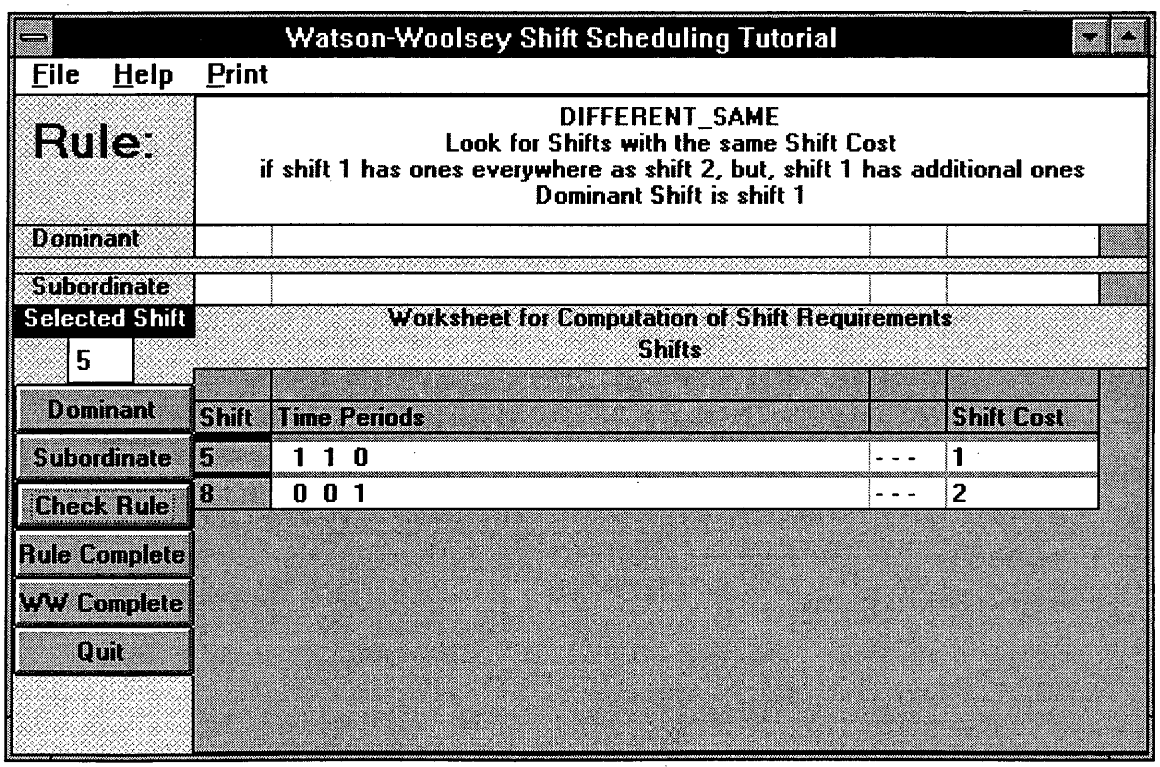


Figure 18 - All Column rule eliminations complete

In "Figure 19 - Example complete, final screen", below, is the display after completing the WWM. Note that this example problem has been reduced to two equations that are essentially equalities. Shift one needs 56 employees on duty and Shift two requires 59. The answer to the original problem will have no employees on the other shifts.

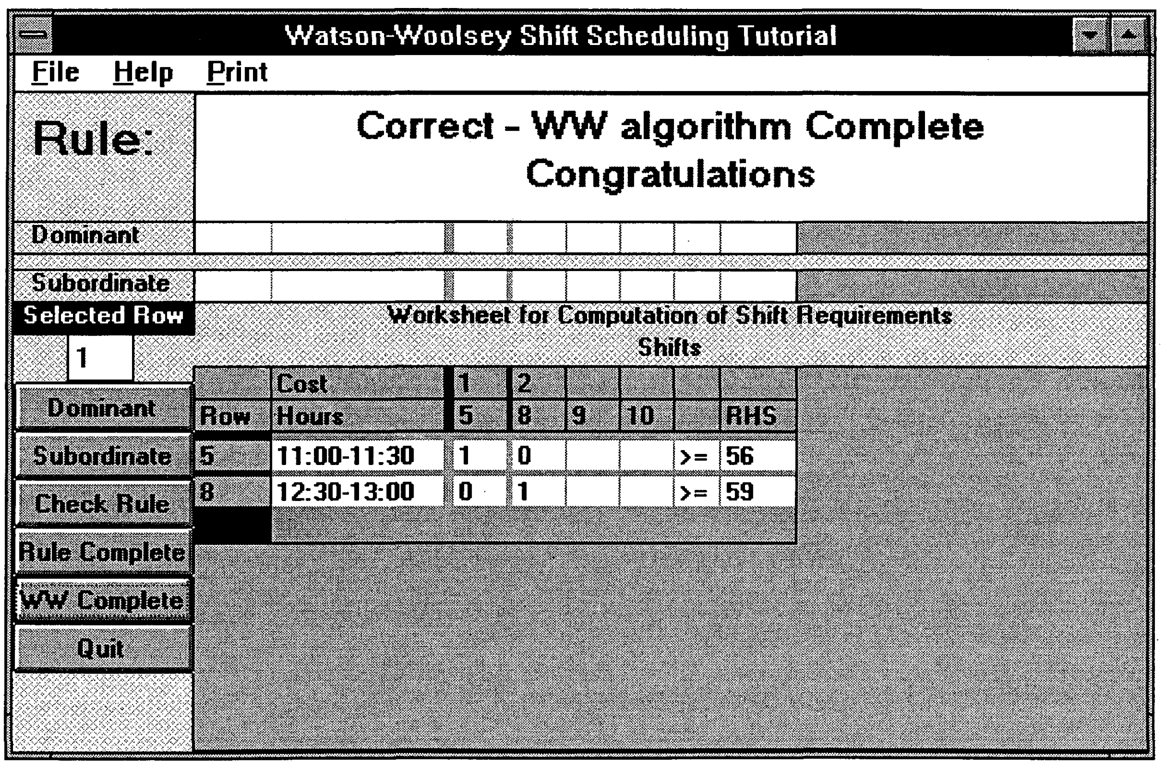


Figure 19 - Example complete, final screen

Sample Files

The sample files are presented in alphabetical order:

- DOMINGO.WWW
- FRIDAY.WWW
- JUEVES.WWW
- MIERCOLE.WWW
- MONDAY.WWW
- SABADO.WWW
- TUESDAY.WWW

Each group contains three items:

1. a figure showing the entire worksheet
2. a table showing dominant/subordinate pairs
3. a figure showing the final reduced problem

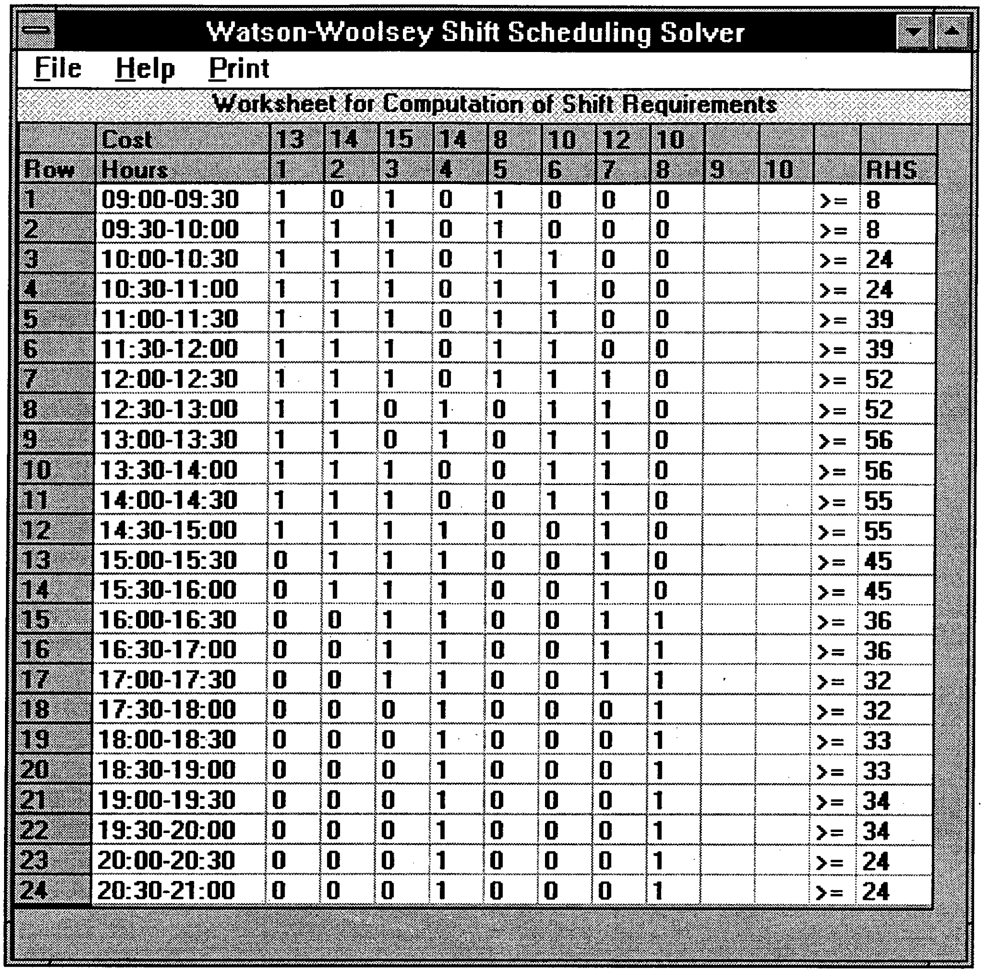
The figures are explained in an earlier section of the User's Guide. The table shows dominant/subordinate pairs of constraints. Rule numbers are across the top, below each rule are columns labeled "D" and "S", for dominant and subordinate, respectively. In these columns are listed the row/column number as they appear in the worksheet. The columns are further subdivided by pass number. This indicates the number of times you have "passed" through all eight rules. In other words, if "Pass 2" is indicated, you have already applied all the rules once and are now on the second "pass" through the rules.

Please note that the tables are NOT comprehensive. The order of your choosing dominant/subordinate pairs in the SAME-SAME rows/columns will affect which rows/columns get eliminated. Use these tables if you get have problems, but, keep in mind that you need to follow the table order.

e.g. enter dominant/subordinate pairs from rule 1, in order, the rule 2, in order, etc.

In this manner the same subordinate will be eliminated and the tables will be accurate.

DOMINGO.WWW sample file

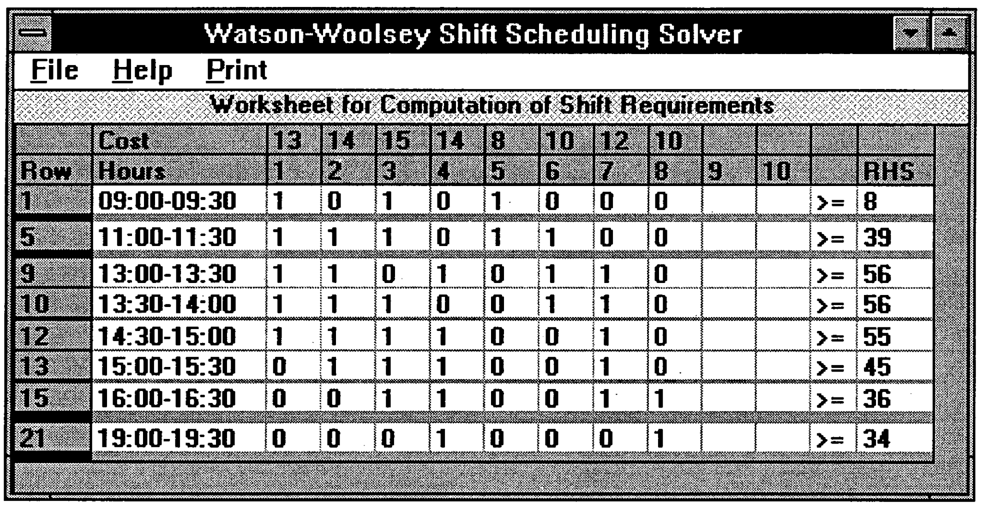


	Cost	13	14	15	14	8	10	12	10				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	1	0	1	0	0	0			>=	8
2	09:30-10:00	1	1	1	0	1	0	0	0			>=	8
3	10:00-10:30	1	1	1	0	1	1	0	0			>=	24
4	10:30-11:00	1	1	1	0	1	1	0	0			>=	24
5	11:00-11:30	1	1	1	0	1	1	0	0			>=	39
6	11:30-12:00	1	1	1	0	1	1	0	0			>=	39
7	12:00-12:30	1	1	1	0	1	1	1	0			>=	52
8	12:30-13:00	1	1	0	1	0	1	1	0			>=	52
9	13:00-13:30	1	1	0	1	0	1	1	0			>=	56
10	13:30-14:00	1	1	1	0	0	1	1	0			>=	56
11	14:00-14:30	1	1	1	0	0	1	1	0			>=	55
12	14:30-15:00	1	1	1	1	0	0	1	0			>=	55
13	15:00-15:30	0	1	1	1	0	0	1	0			>=	45
14	15:30-16:00	0	1	1	1	0	0	1	0			>=	45
15	16:00-16:30	0	0	1	1	0	0	1	1			>=	36
16	16:30-17:00	0	0	1	1	0	0	1	1			>=	36
17	17:00-17:30	0	0	1	1	0	0	1	1			>=	32
18	17:30-18:00	0	0	0	1	0	0	0	1			>=	32
19	18:00-18:30	0	0	0	1	0	0	0	1			>=	33
20	18:30-19:00	0	0	0	1	0	0	0	1			>=	33
21	19:00-19:30	0	0	0	1	0	0	0	1			>=	34
22	19:30-20:00	0	0	0	1	0	0	0	1			>=	34
23	20:00-20:30	0	0	0	1	0	0	0	1			>=	24
24	20:30-21:00	0	0	0	1	0	0	0	1			>=	24

Figure 20 - DOMINGO.WWW

Table 1 - DOMINGO.WWW rule application

File: DOMINGO.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	5	3	1	2	10	7								
	5	6	9	8												
	13	14	10	11												
	15	16	15	17												
	19	20	19	18												
	21	22	21	18												
	23	24	18	23												
			21	19												


Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	13	14	15	14	8	10	12	10				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	1	0	1	0	0	0			>=	8
5	11:00-11:30	1	1	1	0	1	1	0	0			>=	39
9	13:00-13:30	1	1	0	1	0	1	1	0			>=	56
10	13:30-14:00	1	1	1	0	0	1	1	0			>=	56
12	14:30-15:00	1	1	1	1	0	0	1	0			>=	55
13	15:00-15:30	0	1	1	1	0	0	1	0			>=	45
15	16:00-16:30	0	0	1	1	0	0	1	1			>=	36
21	19:00-19:30	0	0	0	1	0	0	0	1			>=	34

Figure 21 - DOMINGO.WWW, solution

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

FRIDAY.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

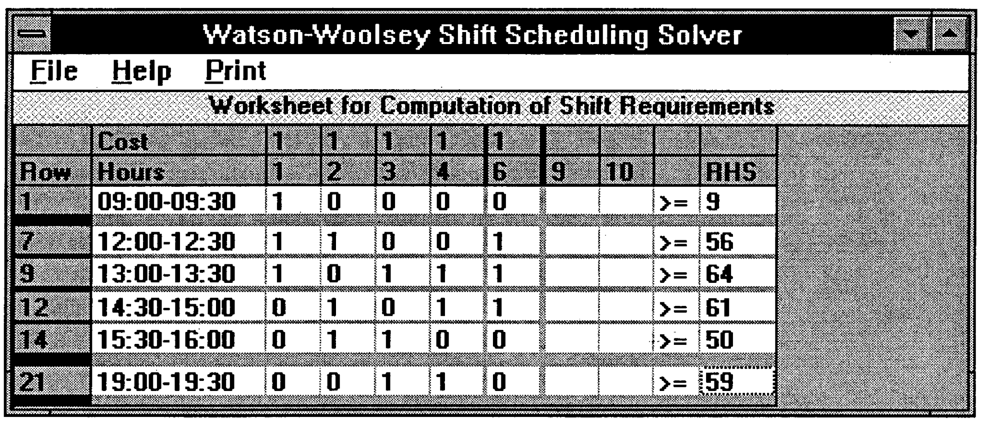
Worksheet for Computation of Shift Requirements

	Cost	1	1	1	1	1	1	1	1				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	0	1	0	0	0			>=	9
2	09:30-10:00	1	1	0	0	1	0	0	0			>=	9
3	10:00-10:30	1	1	0	0	1	1	0	0			>=	28
4	10:30-11:00	1	1	0	0	1	1	0	0			>=	28
5	11:00-11:30	1	1	0	0	1	1	1	0			>=	42
6	11:30-12:00	1	1	0	0	1	1	1	0			>=	42
7	12:00-12:30	1	1	0	0	1	1	1	0			>=	56
8	12:30-13:00	1	1	1	0	1	1	1	0			>=	56
9	13:00-13:30	1	0	1	1	0	1	1	0			>=	64
10	13:30-14:00	1	0	1	1	0	1	1	0			>=	64
11	14:00-14:30	1	1	1	1	0	1	1	0			>=	61
12	14:30-15:00	0	1	0	1	0	1	1	0			>=	61
13	15:00-15:30	0	1	1	0	0	0	1	0			>=	50
14	15:30-16:00	0	1	1	0	0	0	0	0			>=	50
15	16:00-16:30	0	1	1	1	0	0	0	0			>=	42
16	16:30-17:00	0	1	1	1	0	0	0	1			>=	42
17	17:00-17:30	0	1	1	1	0	0	0	1			>=	43
18	17:30-18:00	0	1	1	1	0	0	0	1			>=	43
19	18:00-18:30	0	1	1	1	0	0	0	1			>=	50
20	18:30-19:00	0	0	1	1	0	0	0	1			>=	50
21	19:00-19:30	0	0	1	1	0	0	0	1			>=	59
22	19:30-20:00	0	0	1	1	0	0	0	1			>=	59
23	20:00-20:30	0	0	1	1	0	0	0	1			>=	47
24	20:30-21:00	0	0	1	1	0	0	0	1			>=	47
25	21:00-21:30	0	0	1	1	0	0	0	1			>=	15
26	21:30-22:00	0	0	1	1	0	0	0	1			>=	15

Figure 22 - FRIDAY.WWW

Table 2 - FRIDAY.WWW rule application

File: FRIDAY.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	7	5	1	2	14	15					1	5		
	5	6	17	16	7	8							3	8		
	9	10	19	16	12	11							6	7		
	17	18	19	17	14	13										
	21	22	21	20	14	19										
	23	24	20	23												
	25	26	20	25												
Pass 2			7	3												


Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	1	1	1	1	1				
Row	Hours	1	2	3	4	6	9	10		RHS
1	09:00-09:30	1	0	0	0	0			>=	9
7	12:00-12:30	1	1	0	0	1			>=	56
9	13:00-13:30	1	0	1	1	1			>=	64
12	14:30-15:00	0	1	0	1	1			>=	61
14	15:30-16:00	0	1	1	0	0			>=	50
21	19:00-19:30	0	0	1	1	0			>=	59

Figure 23 - FRIDAY.WWW, solved

JUEVES.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

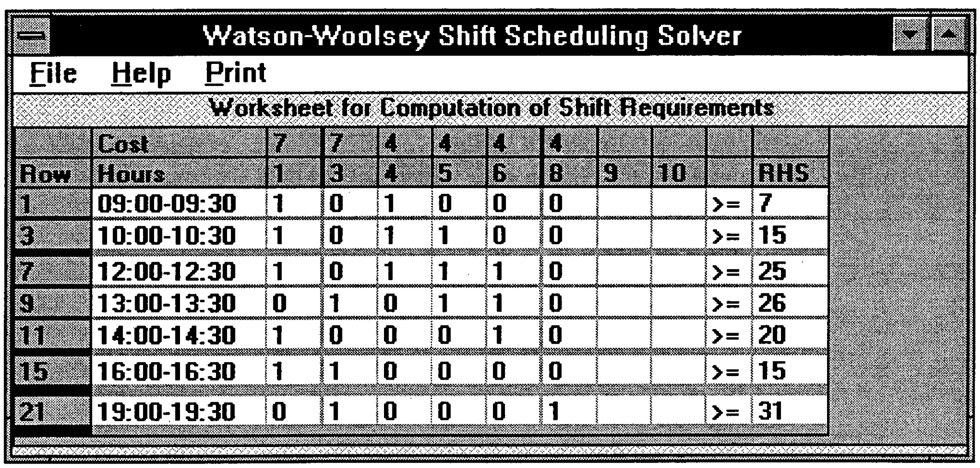
Worksheet for Computation of Shift Requirements

	Cost	7	7.5	7	4	4	4	4	4				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	1	0	0	0	0			>=	7
2	09:30-10:00	1	1	0	1	0	0	0	0			>=	7
3	10:00-10:30	1	1	0	1	1	0	0	0			>=	15
4	10:30-11:00	1	1	0	1	1	0	0	0			>=	15
5	11:00-11:30	1	1	0	1	1	1	0	0			>=	23
6	11:30-12:00	1	1	0	1	1	1	0	0			>=	23
7	12:00-12:30	1	1	0	1	1	1	1	0			>=	25
8	12:30-13:00	1	1	0	1	1	1	1	0			>=	25
9	13:00-13:30	0	0	1	0	1	1	1	0			>=	26
10	13:30-14:00	0	0	1	0	1	1	1	0			>=	26
11	14:00-14:30	1	1	0	0	0	1	1	0			>=	20
12	14:30-15:00	1	1	0	0	0	1	1	0			>=	20
13	15:00-15:30	1	1	1	0	0	0	1	0			>=	15
14	15:30-16:00	1	1	1	0	0	0	1	0			>=	15
15	16:00-16:30	1	1	1	0	0	0	0	0			>=	15
16	16:30-17:00	1	1	1	0	0	0	0	0			>=	15
17	17:00-17:30	0	1	1	0	0	0	0	1			>=	21
18	17:30-18:00	0	1	1	0	8	0	0	1			>=	21
19	18:00-18:30	0	0	1	0	0	0	0	1			>=	26
20	18:30-19:00	0	0	1	0	0	0	0	1			>=	26
21	19:00-19:30	0	0	1	0	0	0	0	1			>=	31
22	19:30-20:00	0	0	1	0	0	0	0	1			>=	31
23	20:00-20:30	0	0	1	0	0	0	0	1			>=	17
24	20:30-21:00	0	0	1	0	0	0	0	1			>=	17

Figure 24 - JUEVES.WWW

Table 3 - JUEVES.WWW rule application

File: JUEVES.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	21	19	1	2	21	17					6	7	1	2
	5	6	19	23	15	13										
	7	8														
	9	10														
	11	12														
	13	14														
	15	16														
	17	18														
	19	20														
	23	24														
Pass 2			7	5												



	Cost	7	7	4	4	4	4				
Row	Hours	1	3	4	5	6	8	9	10		RHS
1	09:00-09:30	1	0	1	0	0	0			>=	7
3	10:00-10:30	1	0	1	1	0	0			>=	15
7	12:00-12:30	1	0	1	1	1	0			>=	25
9	13:00-13:30	0	1	0	1	1	0			>=	26
11	14:00-14:30	1	0	0	0	1	0			>=	20
15	16:00-16:30	1	1	0	0	0	0			>=	15
21	19:00-19:30	0	1	0	0	0	1			>=	31

Figure 25 - JUEVES.WWW, solved

MIERCOLE.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

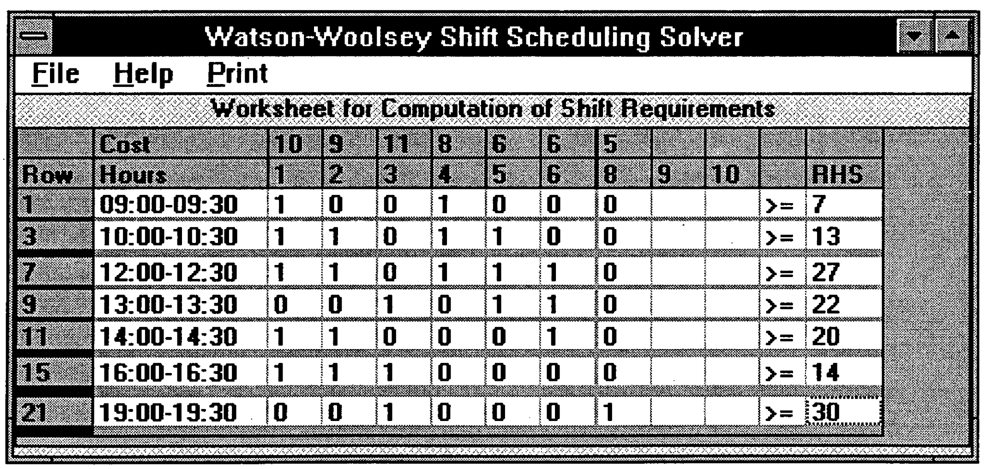
Worksheet for Computation of Shift Requirements

	Cost	10	9	11	8	6	6	6	5				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	1	0	0	0	0			>=	7
2	09:30-10:00	1	1	0	1	0	0	0	0			>=	7
3	10:00-10:30	1	1	0	1	1	0	0	0			>=	13
4	10:30-11:00	1	1	0	1	1	0	0	0			>=	13
5	11:00-11:30	1	1	0	1	1	1	0	0			>=	19
6	11:30-12:00	1	1	0	1	1	1	0	0			>=	19
7	12:00-12:30	1	1	0	1	1	1	1	0			>=	27
8	12:30-13:00	1	1	0	1	1	1	1	0			>=	27
9	13:00-13:30	0	0	1	0	1	1	1	0			>=	22
10	13:30-14:00	0	0	1	0	1	1	1	0			>=	22
11	14:00-14:30	1	1	0	0	0	1	1	0			>=	20
12	14:30-15:00	1	1	0	0	0	1	1	0			>=	20
13	15:00-15:30	1	1	1	0	0	0	1	0			>=	14
14	15:30-16:00	1	1	1	0	0	0	1	0			>=	14
15	16:00-16:30	1	1	1	0	0	0	0	0			>=	14
16	16:30-17:00	1	1	1	0	0	0	0	0			>=	14
17	17:00-17:30	0	1	1	0	0	0	0	1			>=	17
18	17:30-18:00	0	1	1	0	0	0	0	1			>=	17
19	18:00-18:30	0	0	1	0	0	0	0	1			>=	24
20	18:30-19:00	0	0	1	0	0	0	0	1			>=	24
21	19:00-19:30	0	0	1	0	0	0	0	1			>=	30
22	19:30-20:00	0	0	1	0	0	0	0	1			>=	30
23	20:00-20:30	0	0	1	0	0	0	0	1			>=	14
24	20:30-21:00	0	0	1	0	0	0	0	1			>=	14

Figure 26 - MIERCOLE.WWW

Table 4 - MIERCOLE.WWW rule application

File: MIERCOLE.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	21	19	1	2	21	17					6	7		
	5	6	19	23	15	13										
	7	8														
	9	10														
	11	12														
	13	14														
	15	16														
	17	18														
	19	20														
	21	22														
	23	24														
Pass 2			7	5												



Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	10	9	11	8	6	6	5				
Row	Hours	1	2	3	4	5	6	8	9	10		RHS
1	09:00-09:30	1	0	0	1	0	0	0			>=	7
3	10:00-10:30	1	1	0	1	1	0	0			>=	13
7	12:00-12:30	1	1	0	1	1	1	0			>=	27
9	13:00-13:30	0	0	1	0	1	1	0			>=	22
11	14:00-14:30	1	1	0	0	0	1	0			>=	20
15	16:00-16:30	1	1	1	0	0	0	0			>=	14
21	19:00-19:30	0	0	1	0	0	0	1			>=	30

Figure 27 - MIERCOLE.WWW, solved

MONDAY.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

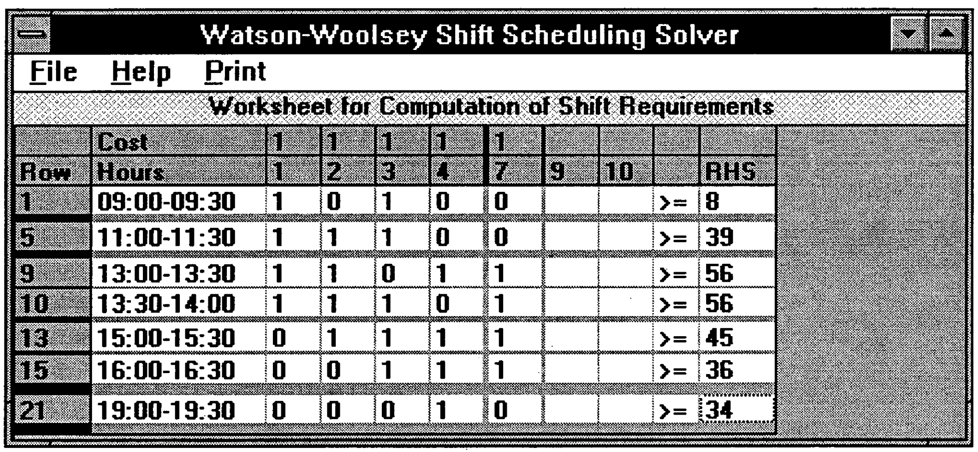
Worksheet for Computation of Shift Requirements

	Cost	1	1	1	1	1	1	1	1				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	1	0	1	0	0	0			>=	8
2	09:30-10:00	1	1	1	0	1	0	0	0			>=	8
3	10:00-10:30	1	1	1	0	1	1	0	0			>=	24
4	10:30-11:00	1	1	1	0	1	1	0	0			>=	24
5	11:00-11:30	1	1	1	0	1	1	0	0			>=	39
6	11:30-12:00	1	1	1	0	1	1	0	0			>=	39
7	12:00-12:30	1	1	1	0	1	1	1	0			>=	52
8	12:30-13:00	1	1	0	1	0	1	1	0			>=	52
9	13:00-13:30	1	1	0	1	0	1	1	0			>=	56
10	13:30-14:00	1	1	1	0	0	1	1	0			>=	56
11	14:00-14:30	1	1	1	0	0	1	1	0			>=	55
12	14:30-15:00	1	1	1	1	0	0	1	0			>=	55
13	15:00-15:30	0	1	1	1	0	0	1	0			>=	45
14	15:30-16:00	0	1	1	1	0	0	1	0			>=	45
15	16:00-16:30	0	0	1	1	0	0	1	1			>=	36
16	16:30-17:00	0	0	1	1	0	0	1	1			>=	36
17	17:00-17:30	0	0	1	1	0	0	1	1			>=	32
18	17:30-18:00	0	0	0	1	0	0	0	1			>=	32
19	18:00-18:30	0	0	0	1	0	0	0	1			>=	33
20	18:30-19:00	0	0	0	1	0	0	0	1			>=	33
21	19:00-19:30	0	0	0	1	0	0	0	1			>=	34
22	19:30-20:00	0	0	0	1	0	0	0	1			>=	34
23	20:00-20:30	0	0	0	1	0	0	0	1			>=	24
24	20:30-21:00	0	0	0	1	0	0	0	1			>=	24

Figure 28 - MONDAY.WWW

Table 5 - MONDAY.WWW rule application

File: MONDAY.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	5	3	1	2	10	7					1	5		
	5	6	9	8									1	6		
	13	14	10	11									4	8		
	15	16	15	17												
	19	20	19	18												
	21	22	21	18												
	23	24	18	23												
			21	19												
Pass 2							9	12								


Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

Row	Cost / Hours	1 / 1	1 / 2	1 / 3	1 / 4	1 / 7	9	10		RHS
1	09:00-09:30	1	0	1	0	0			>=	8
5	11:00-11:30	1	1	1	0	0			>=	39
9	13:00-13:30	1	1	0	1	1			>=	56
10	13:30-14:00	1	1	1	0	1			>=	56
13	15:00-15:30	0	1	1	1	1			>=	45
15	16:00-16:30	0	0	1	1	1			>=	36
21	19:00-19:30	0	0	0	1	0			>=	34

Figure 29 - MONDAY.WWW, solved

SABADO.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

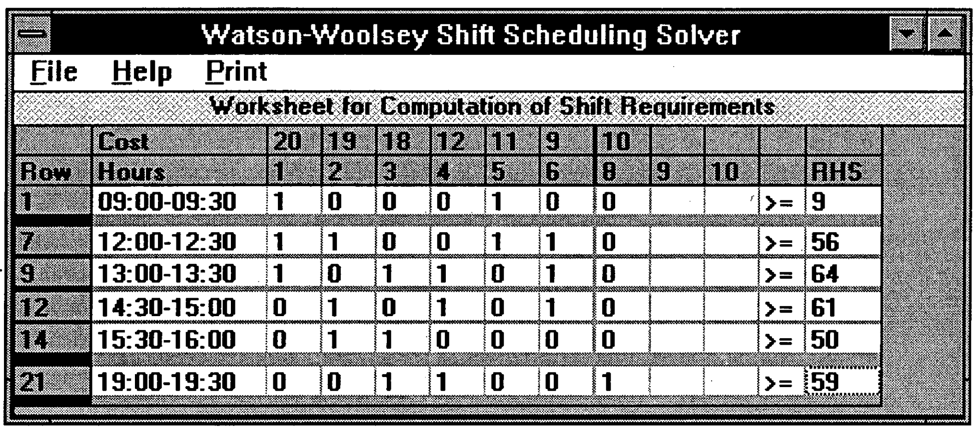
Worksheet for Computation of Shift Requirements

	Cost	20	19	18	12	11	9	11	10				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	0	1	0	0	0			>=	9
2	09:30-10:00	1	1	0	0	1	0	0	0			>=	9
3	10:00-10:30	1	1	0	0	1	1	0	0			>=	28
4	10:30-11:00	1	1	0	0	1	1	0	0			>=	28
5	11:00-11:30	1	1	0	0	1	1	1	0			>=	42
6	11:30-12:00	1	1	0	0	1	1	1	0			>=	42
7	12:00-12:30	1	1	0	0	1	1	1	0			>=	56
8	12:30-13:00	1	1	1	0	1	1	1	0			>=	56
9	13:00-13:30	1	0	1	1	0	1	1	0			>=	64
10	13:30-14:00	1	0	1	1	0	1	1	0			>=	64
11	14:00-14:30	1	1	1	1	0	1	1	0			>=	61
12	14:30-15:00	0	1	0	1	0	1	1	0			>=	61
13	15:00-15:30	0	1	1	0	0	0	1	0			>=	50
14	15:30-16:00	0	1	1	0	0	0	0	0			>=	50
15	16:00-16:30	0	1	1	1	0	0	0	0			>=	42
16	16:30-17:00	0	1	1	1	0	0	0	1			>=	42
17	17:00-17:30	0	1	1	1	0	0	0	1			>=	43
18	17:30-18:00	0	1	1	1	0	0	0	1			>=	43
19	18:00-18:30	0	1	1	1	0	0	0	1			>=	50
20	18:30-19:00	0	0	1	1	0	0	0	1			>=	50
21	19:00-19:30	0	0	1	1	0	0	0	1			>=	59
22	19:30-20:00	0	0	1	1	0	0	0	1			>=	59
23	20:00-20:30	0	0	1	1	0	0	0	1			>=	47
24	20:30-21:00	0	0	1	1	0	0	0	1			>=	47
25	21:00-21:30	0	0	1	1	0	0	0	1			>=	15
26	21:30-22:00	0	0	1	1	0	0	0	1			>=	15

Figure 30 - SABADO.WWW

Table 6 - SABADO.WWW rule application

File: SABADO.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	7	5	1	2	14	15							6	7
	5	6	17	16	7	8										
	9	10	19	16	12	11										
	17	18	19	17	14	13										
	21	22	21	20	14	19										
	23	24	20	23												
	25	26	20	25												
Pass 2			7	3												


Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	20	19	18	12	11	9	10				
Row	Hours	1	2	3	4	5	6	8	9	10		RHS
1	09:00-09:30	1	0	0	0	1	0	0			>=	9
7	12:00-12:30	1	1	0	0	1	1	0			>=	56
9	13:00-13:30	1	0	1	1	0	1	0			>=	64
12	14:30-15:00	0	1	0	1	0	1	0			>=	61
14	15:30-16:00	0	1	1	0	0	0	0			>=	50
21	19:00-19:30	0	0	1	1	0	0	1			>=	59

Figure 31 - SABADO.WWW, solved

TUESDAY.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	1	1	1	1	1	1	1	1				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	1	0	0	0	0			>=	7
2	09:30-10:00	1	1	0	1	0	0	0	0			>=	7
3	10:00-10:30	1	1	0	1	1	0	0	0			>=	15
4	10:30-11:00	1	1	0	1	1	0	0	0			>=	15
5	11:00-11:30	1	1	0	1	1	1	0	0			>=	23
6	11:30-12:00	1	1	0	1	1	1	0	0			>=	23
7	12:00-12:30	1	1	0	1	1	1	1	0			>=	25
8	12:30-13:00	1	1	0	1	1	1	1	0			>=	25
9	13:00-13:30	0	0	1	0	1	1	1	0			>=	26
10	13:30-14:00	0	0	1	0	1	1	1	0			>=	26
11	14:00-14:30	1	1	0	0	0	1	1	0			>=	20
12	14:30-15:00	1	1	0	0	0	1	1	0			>=	20
13	15:00-15:30	1	1	1	0	0	0	1	0			>=	15
14	15:30-16:00	1	1	1	0	0	0	1	0			>=	15
15	16:00-16:30	1	1	1	0	0	0	0	0			>=	15
16	16:30-17:00	1	1	1	0	0	0	0	0			>=	15
17	17:00-17:30	0	1	1	0	0	0	0	1			>=	21
18	17:30-18:00	0	1	1	0	0	0	0	1			>=	21
19	18:00-18:30	0	0	1	0	0	0	0	1			>=	26
20	18:30-19:00	0	0	1	0	0	0	0	1			>=	26
21	19:00-19:30	0	0	1	0	0	0	0	1			>=	31
22	19:30-20:00	0	0	1	0	0	0	0	1			>=	31
23	20:00-20:30	0	0	1	0	0	0	0	1			>=	17
24	20:30-21:00	0	0	1	0	0	0	0	1			>=	17

Figure 32 - TUESDAY.WWW

Table 7 - TUESDAY.WWW rule application

File: TUESDAY.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	21	19	1	2	21	17					1	2		
	5	6	19	23	15	13							1	4		
	7	8											3	8		
	9	10											6	7		
	11	12														
	13	14														
	15	16														
	17	18														
	19	20														
	21	22														
	23	24														
Pass 2			7	5			21	9					1	5		
							21	15					1	6		
Pass 3			3	1												
			7	1												
			11	1												
			7	3												
			11	3												
			7	11												

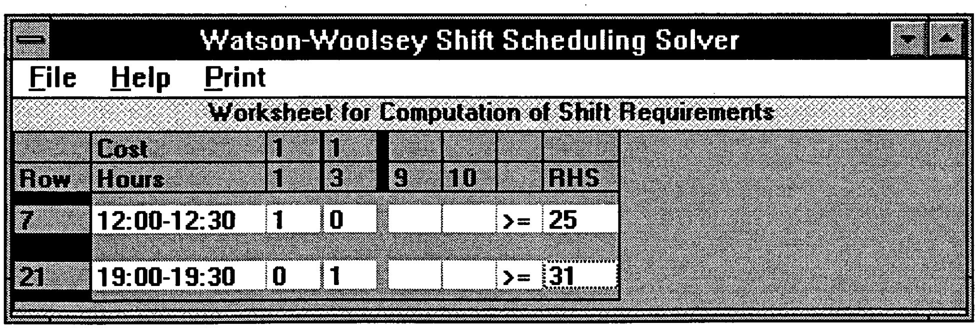


Figure 33 - TUESDAY.WWW, solved

WEDNESDY.WWW sample file

Watson-Woolsey Shift Scheduling Solver

File Help Print

Worksheet for Computation of Shift Requirements

	Cost	1	1	1	1	1	1	1	1				
Row	Hours	1	2	3	4	5	6	7	8	9	10		RHS
1	09:00-09:30	1	0	0	1	0	0	0	0			>=	7
2	09:30-10:00	1	1	0	1	0	0	0	0			>=	7
3	10:00-10:30	1	1	0	1	1	0	0	0			>=	13
4	10:30-11:00	1	1	0	1	1	0	0	0			>=	13
5	11:00-11:30	1	1	0	1	1	1	0	0			>=	19
6	11:30-12:00	1	1	0	1	1	1	0	0			>=	19
7	12:00-12:30	1	1	0	1	1	1	1	0			>=	27
8	12:30-13:00	1	1	0	1	1	1	1	0			>=	27
9	13:00-13:30	0	0	1	0	1	1	1	0			>=	22
10	13:30-14:00	0	0	1	0	1	1	1	0			>=	22
11	14:00-14:30	1	1	0	0	0	1	1	0			>=	20
12	14:30-15:00	1	1	0	0	0	1	1	0			>=	20
13	15:00-15:30	1	1	1	0	0	0	1	0			>=	14
14	15:30-16:00	1	1	1	0	0	0	1	0			>=	14
15	16:00-16:30	1	1	1	0	0	0	0	0			>=	14
16	16:30-17:00	1	1	1	0	0	0	0	0			>=	14
17	17:00-17:30	0	1	1	0	0	0	0	1			>=	17
18	17:30-18:00	0	1	1	0	0	0	0	1			>=	17
19	18:00-18:30	0	0	1	0	0	0	0	1			>=	24
20	18:30-19:00	0	0	1	0	0	0	0	1			>=	24
21	19:00-19:30	0	0	1	0	0	0	0	1			>=	30
22	19:30-20:00	0	0	1	0	0	0	0	1			>=	30
23	20:00-20:30	0	0	1	0	0	0	0	1			>=	14
24	20:30-21:00	0	0	1	0	0	0	0	1			>=	14

Figure 34 - WEDNESDY.WWW

Table 8 - WEDNESDY.WWW rule application

File: WEDNESDY.WWW																
Rule	1		2		3		4		5		6		7		8	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S	D	S
Pass 1	3	4	21	19	1	2	21	17					1	2		
	5	6	19	23	15	13							1	4		
	7	8											3	8		
	9	10											6	7		
	11	12														
	13	14														
	15	16														
	17	18														
	19	20														
	21	22														
	23	24														
Pass 2			7	5			21	9					1	5		
							21	15					1	6		
Pass 3			3	1												
			7	1												
			11	1												
			7	3												
			11	3												
			7	11												

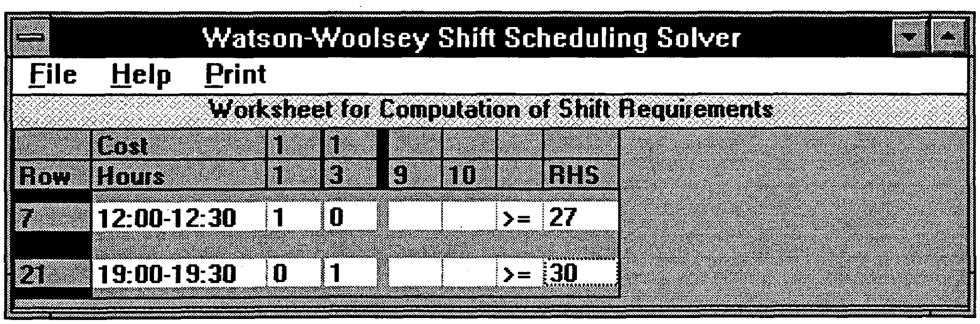


Figure 35 - WEDNESDY.WWW, solved

Recommendations for User

The first time user is encouraged to work through the EXAMPLE.WWW sample file along with the tutorial. From that point it is suggested that you concentrate on the files containing the rules which seem the most difficult to understand. The purpose of the tutorial is to make it second nature to spot rows which meet the method's criteria for removal. It is possible to create your own problems and enter them into this tutorial. The file formats are defined in the Appendix, modifying an existing file is the simplest way to create a new file. Fewer errors will occur with this technique.

ARTHUR LAKES LIBRARY
COLORADO SCHOOL OF MINES
GOLDEN, CO 80401

Conclusions

Why was this tutorial necessary?

The purpose of this tutorial is to create a 'learn by doing' or 'hands on' tool to assist in performing the Watson-Woolsey Shift Scheduling Method. Learning this method in a classroom with an instructor may prove adequate for some students. For those of us who would like or need some additional practice in performing this method the tutorial provides sample problems and verification of correct application of the WWM.

Reading about a method is a simple task. Learning that method from a book, without someone to ask when you have a problem, without any feedback about your performance, is extremely difficult. This tutorial is designed to provide an 'expert' to guide you, encourage you, or just provide assurance that you are doing it right.

Further Recommendations

There is an entire world of topics which fit this category. They fit into some neat and clean areas:

1. Adding to the tutorial
2. Adding solutions to the program

3. Expanding the method

In the first category there are many subjects that could use a good tutorial.

- Linear Programming Simplex Algorithm
- Gomory Cutting Planes
- Lawler and Bell
- Balas algorithm for Zero-One Integer Programming problems
- Geometric Programming techniques

The second category might include modifying the program to reduce the problem and then provide an algorithm to solve the reduced IP problem. Or you could modify it to output in a format that an LP solver like STORM could read.

The third category might be the most important. It might be possible to use this reduction method on problems where the matrix coefficients (the $A_{i,j}$'s) were other numbers than just zero or one? The simplest would be for positive integers, then expand for positive and negative integers. And then on to the real numbers! This type of expansion of the method would be useful with any type of problem. One might argue, after a reasonable search of the relevant literature, that the utility of removing redundant or bounded constraints has been underestimated or overlooked by the Operations Research community in general.

The Watson-Woolsey Method is simply a preprocessing step to simplify the original problem to the smallest equivalent. This thesis has presented a user-friendly tutorial process to aid in the learning and use of this algorithm

References

1) Andrews, B. H., and Parsons, H. L., 1989. "L. L. Bean chooses a Telephone Agent Scheduling System," *Interfaces*, Vol. 19, No. 6, (November-December), pp. 1-9.

2) Balas, E.,1965. "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research*, Vol. 13, pp. 517-547.

3) ______.,1970. "Duality in Discrete Programming," *Proceedings of the Princeton Symposium on Mathematical Programmming*, (eds. H. Kuhn), Princeton University Press, pp. 179-197.

4) Bartholdi, J. J., 1981. "A Guaranteed Accuracy Round Off Algorithm for Cyclic Scheduling and Set Covering," *Operations Research*, Vol. 29, No. 3 (May-June), pp. 501-510.

5) Bechtold, S. E.,Janaro, R. E., and Sumners, D. L. 1984. "Maximization of Labor Productivity through Optimal Rest-Break Schedules," *Management Science*, Vol. 30, No. 12 (December), pp. 1442-1458.

6) Brownwell, W. S., Lowerre, J. M., 1976. "Scheduling of Work Forces Required in Continuous Operation Under Alternative Labor Policies," *Management Science*, Vol. 22, No. 5 (January), pp. 597-605.

7) Burns, R. N., and Carter, M. W., 1985. "Work Force Size and Single Shift Schedules with Variable Demands," *Management Science*, Vol. 31, No. 5, (May), pp. 599-607.

8) Choypeng, P., Puakpong, P., and Rosenthal, R. E., 1986. "Optimal Ship Routing and Personnel Assignment for Naval Recruitment in Thailand," *Interfaces*, Vol. 16, No. 4 (July-August), pp. 47-52.

9) Churchman, C. W., Ackoff, R. L., and Arnold, E. L., 1961. Chapter 12 in Introduction to Operations Research. John Wiley and Sons.

10) Cline, A. K., King, D. H., and Meyering, J. M., 1992. "Routing and Scheduling Coast Guard Buoy Tenders," *Interfaces*, Vol. 22, No. 3, (May-June), pp. 56-72.

11) Gaballa, A., and Pearce, W., 1979. "Telephone Sales Manpower Planning at Qantas," *Interfaces*, Vol. 9, No. 3 (May), pp. 1-9.

12) Glover, F., and Woolsey, R. E. D., 1970. "Aggregating Diophantine Equations," Management Science Report Series, University of Colorado, No. 70-4, (October).

13) Gomory, R, "An Algorithm for Integer Solutions to Linear Programs," Recent Advances in Mathematical Programming (eds.: Graves and Wolfe), New York, McGraw-Hill, 1963.

14) ______, "All-Integer Integer Programming Algorithm," in Industrial Scheduling (eds: Muth and Thompson), Englewood Cliffs, NJ, Prentice-Hall, 1963.

15) Holloran, T. J., and Byrn, J. E., 1986. "United Airlines Station Manpower Planning System," *Interfaces*, Vol. 16, No. 1, (January-February), pp. 39-50.

16) Lawler, E. L., and Bell, R. L., "A Method for Solving Discrete Optimization Problems," *Operations Research,* Vol. 14, pp. 1098-1112.

17) Miller, H. E., Pierskalla, W. P., and Rath, G. J., 1976. "Nurse Scheduling using Mathematical Programming," *Operations Research*, Vol. 24, No. 5 (September-October), pp. 857-870.

18) Mulvey, J. M., 1979. "Strategies in Modeling: A Personnel Scheduling Example," *Interfaces*, Vol. 9, No. 3 (May), pp. 66-76.

19) Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M., 1983. Chapter 11 in Engineering Optimization, Methods and Applications. John Wiley and Sons.

20) Thompson, G. L., Tonge, F. M., and Zionts, S., 1966. "Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems," *Management Science*, Vol. 12, No. 7 (March), pp. 588-608.

21) Tien, J. M., and Kamiyama, A., 1982. "On Manpower Scheduling Algorithms," *SIAM Review*, Vol. 24, No. 3 (July), pp. 275-287.

22) Watson, James L. , "Efficient Solution of Shift Scheduling Problems by Determining the Minimal Sets of Constraints and Variables," in Colorado School of Mines Thesis (T-4014), 1991.

23) _____ & Woolsey, R.E.D.; "Solving Shift Scheduling Problems With Common Sense and . . . I," *AGIFORS 30th Annual Symposium 1990*, pp. 243-269.

24) ______ & Woolsey, R.E.D.; "Solving Shift Scheduling Problems With Common Sense and . . . II," *AGIFORS 30th Annual Symposium 1990*, pp. 269-296.

25) Woolsey, R.E.D.,1973. "A Novena to St. Jude, or Four Edifying Case Studies in Mathematical Programming," *Interfaces*, Vol. 4, No. 1, (November), pp. 32-39.

26) Wermus, M., and Pope, J. A., 1994. "Scheduling Harbor Pilots," *Interfaces*, Vol. 24, No. 2, (March-April), pp. 44-52.

27) Wu, N. L., and Coppins, R., 1981. Linear Programming and Extensions, McGraw-Hill.

28) Balas, Egon, 1994. Private Communication with Dr. Woolsey.