9-1-1986

Using Management Science to Assign Judges to Casetypes

Stuart S. Nagel

Miriam K. Mills

Special Issues and Topics

Using Management Science to Assign Judges to Casetypes

STUART S. NAGEL* AND MIRIAM K. MILLS**

The purpose of this article is to discuss the relevance of management science to assigning judges to types of cases.[1] The article shows how one can utilize modern management science to assign judges to cases more systematically than with the traditional rotation system.[2] The methodology is versatile in taking into consideration any constraints that relate to (1) the range of trial hours each judge works on the average per week, (2) the range of trial hours for different types of cases in an average week, and (3) constraints that relate to equity, efficiency, effectiveness, and other considerations. The object is to assign judges to cases in light of their track records, their interests, and the caseload. A law firm could use similar methods to assign lawyers more efficiently to casetypes.

The most relevant management science method, in this context, is called linear programming.[3] The relations are "linear" on the assumption that diminishing marginal returns or diminishing incremental output does not occur over a relatively short span of time, such as ten trial hours a week. The term "programming" in the

---

* Professor of Political Science, University of Illinois at Urbana-Champaign.

** Associate Professor of Organizational Science, New Jersey Institute of Technology.

1. For a debate on whether systematic judicial assignment should be adopted, see Nagel, *A Response to the Responses*, 70 JUDICATURE 73 (1986); Nagel, *Systematic Assignment of Judges: A Proposal*, 70 JUDICATURE 79 (1986); Polansky, *Systematic Assignment in the Urban Court*, 70 JUDICATURE 76 (1986); Ryan, *Judicial Assignment, Efficiency, and Politics*, 70 JUDICATURE 78 (1986); Slate, *Can and Should There be Systematic Assignment of Judges to Casetypes?*, 70 JUDICATURE 77 (1986).

2. For a discussion of how judges are traditionally assigned to cases, see ABA COMMISSION ON STANDARDS OF JUDICIAL ADMINISTRATION, STANDARDS RELATING TO TRIAL COURTS 86-93 (1976); PRESIDENT'S COMMISSION ON LAW ENFORCEMENT AND ADMINISTRATION OF JUSTICE, TASK FORCE REPORT: THE COURTS 88-90, 165-67 (1967). The most common assignment method is rotation of cases across judges.

3. For literature on linear programming and its variations as applied to assigning people to tasks, see W. ERIKSON & O. HALL, COMPUTER MODELS FOR MANAGEMENT SCIENCE 23-52, 71-86 (1983); S. LEE, LINEAR OPTIMIZATION FOR MANAGEMENT 1-246, 307-32 (1976); S. RICHMOND, OPERATIONS RESEARCH FOR MANAGEMENT DECISIONS 277-85, 314-82 (1968). None of the literature examined describes linear programming as applied to judicial assignment.

phrase linear programming is synonymous with optimizing. In other words, finding an optimum allocation of scarce resources in the sense of maximizing some goal or goals. In this case, the scarce resource is judicial time. The main goal is to allocate judges to different cases in light of how well each performs in handling various casetypes.

I. THE PROBLEM OF ASSIGNING TWO JUDGES TO TWO CASETYPES

A. *Expressing the Problem*

Table 1 shows the basic data for illustrating the management science methodology and the competing methodology of intelligent trial and error. We have two judges named Judge Fox and Judge Wolf. They are each expected to put in 10 trial hours per week. The rest of their 40-hour weeks are spent in nontrial activities. In an average week, there are 20 hours of trial time, with about 40% attributable to criminal cases and 60% attributable to civil cases.

Hypothetically, each judge was included in a survey of lawyers who practice in this court system. The lawyers were asked to score the expertise of the judges in criminal and civil cases on a five-point scale. The five points can be expressed in words or symbols consisting of:

1. - - or highly incompetent;
2. — or mildly competent;
3. 0 or neither competent nor incompetent;
4. + or mildly competent; or
5. + + or highly competent in handling this casetype.

Judge Fox received an average score of 4 in criminal cases and a 3 in civil cases. Judge Wolf received a 2 in criminal cases and a 3 in civil cases.

The scores on the 1-5 scale do not have to refer merely to competency as determined by a lawyer survey. They can also be composite scores that take into consideration judges' interests, their speed in handling certain types of cases, their records on appeal, seniority, or whatever else is considered relevant to the systematic assignment of cases to judges. An ideal assignment system involves some tests, indicators, or other methods for predicting how well judges will do in various types of cases before they are assigned to a casetype or a set of casetypes. Predicting judicial performance is done more in judicial systems such as the French system where judges undergo special training and testing before beginning their judicial careers.[4]

4. *See* R. DAVID & J. BRILEY, MAJOR LEGAL SYSTEMS IN THE WORLD TODAY 21-118

TABLE 1. THE PROBLEM OF ALLOCATING 20 HOURS TO TWO JUDGES FOR TWO CASETYPES

CASES / JUDGE	CRIMINAL		CIVIL		Hours per Judge
	Quality Score	Hours Assigned	Quality Score	Hours Assigned	
FOX	4	a	3	b	10
WOLF	2	c	3	d	10
Hours per Casetype		8		12	20

NOTES:

1. The allocation system is shown in its simplest form with two judges and two casetypes.
2. Each judge is expected to put in 10 hours a week to satisfy the average weekly total of 20 hours of trial time.
3. Criminal cases constitute 40% of the total or 8 hours, and civil cases constitute 60% or 12 hours.
4. Judge Fox received scores of 4 and 3 on the two casetypes, and Judge Wolf received scores of 2 and 3.

For each judge in each type of case, one can calculate an hours-times-quality product. The problem is how to allocate the 20 trial hours between the two judges so as to maximize the sum of all the hours-times-quality products. In algebraic terms, we would like to solve for a, b, c, and d in Table 1.

Table 2 states the problem algebraically or in terms of symbolic logic as a goal or objective function to be maximized subject to various constraints. The goal to be maximized is the sum of the four products of 4a, 3b, 2c, and 3d. Each judge should be assigned ten hours, which means a + b = 10, and c + d = 10. Of the 20 hours, 8 hours or 40% should be criminal and 12 hours noncriminal, or a + c = 8, and b + d = 12.

In order to communicate the objective function and the constraints to a microcomputer, one needs to express the problem in terms of the coefficients or multipliers of the variables in the objective function and in the four constraints. The variables are a, b, c, and d. Their coefficients in the objective function are 4, 3, 2, and 3, as indicated in Table 2. The first coefficient in the first constraint is a 1 because the variable "a" is in effect multiplied by 1. The second coefficient is also a 1. The third coefficient is a 0 because variable c is not present in the first constraint. Likewise, the fourth coefficient is also 0. The right-hand side of the first constraint is 10 hours. One can easily determine the coefficients for the other three constraints by noting that if a letter or a variable appears in a constraint without a multiplier, then its coefficient is 1. If a variable does not appear at all in a constraint, then its coefficient is 0 in that constraint.

B. *The Solutions*

Table 3 shows all the alternative possibilities for allocating 20 hours to two judges for two casetypes. Manipulating Cell A determines the set of alternative possibilities. In other words, once any one cell is determined in a four-cell table, the numerical values of the other three cells are automatically determined in order for the rows to add to 10, and for the columns to add to 8 and 12. Alternative #1 sets Cell A at 8 hours, Alternative #2 sets Cell A at 7 hours, and so on through Alternative #9 which sets Cell A at 0 hours.

To aid in determining which of the nine alternatives is best, the upper left corner of each cell indicates the quality score that the judge received on the casetype (e.g., 4). The assigned hours (e.g., 8) are given in the center of each cell. The lower right corner of each cell

(1968); H. LIEBESNY, FOREIGN LEGAL SYSTEMS: A COMPARATIVE ANALYSIS 283-84 (4th rev. ed. 1981).

TABLE 2. CONVERTING THE JUDICIAL ASSIGNMENT PROBLEM INTO A SET OF LINEAR PROGRAMMING STATEMENTS AND COEFFICIENTS

VARIABLES / STATEMENTS	Algebraic Form	Linear Programming Coefficients				
		a (1)	b (2)	c (3)	d (4)	Right-Hand Side
Objective Function	4a + 3b + 2c + 3d	4	3	2	3	
Fox Hours	a + b = 10	1	1	0	0	10
Wolf Hours	c + d = 10	0	0	1	1	10
Criminal Hours	a + c =8	1	0	1	0	8
Civil Hours	b + d = 12	0	1	0	1	12

NOTES:

1. Any variable in an equation in the algebraic form column that has no coefficient is considered to have a coefficient of 1 for that equation. Any variable not in an equation is considered to have a coefficient of 0 for that equation.
2. The concept of "statements" in the first column includes statements of equality, greater than or equal to, less than or equal to, and statements concerning the objective function to be maximized.
3. Variables in the right-hand columns are shown with both letters and numbers. The letters are needed to communicate with the computer because the computer refers to the variables by number.

indicates the product of the assigned hours times the quality score (e.g., 32). The sum of the four products is then shown in the lower right of the four-cell table (e.g., 68). For example, in Alternative #1, the assignments are 8, 2, 0, and 10 hours for the four cells, which generates a grand total of 68.

From the nine alternative allocations, one can easily see that Alternative #1 is the best choice because it has a grand total of 68 points in terms of quality-weighted hours. The other alternatives all have less than 68 points. In addition, the alternatives diminish by 2 points whenever Cell A is lowered by 1 hour. Thus, the change slope for Cell A is +2 which means that whenever Cell A rises by 1 hour, the grand total or objective function also increases by 2 hours. In common sense terms, the analysis shows we should give as many hours as possible to Cell A because it has the highest quality score. Hence, Alternative #1, in which Cell A receives an 8, is the most advantageous distribution. Another way of attaining the same result is to say Cell C should receive the fewest hours possible because it has the lowest quality score. Under this method, Cell C receives 0 hours which is also delineated in Alternative #1.

One might also note that Alternative #9 can be considered the worst or malimum alternative, as contrasted to Alternative #1 which is the optimum alternative. Knowing the malimum allocation is helpful because one can see how close the actual allocation is to both the worst and the best allocation. The malimum can be arrived at by following the opposite rules of those used to arrive at the optimum. In other words, Cell C is given the most hours because it is the cell with the lowest quality score. The most hours that Cell C can be given in order to stay within the column total is 8 hours. Once these hours have been assigned, then the other cell entries are automatically determined in order to properly add the row and column totals. Thus, the malimum allocation is 0, 10, 8, and 2 for a total of 52.

The malimum serves the useful purpose of showing what the overall score would be should the cases be assigned on a random rotation basis instead of an optimizing basis. The result from random assignment can be determined by averaging the optimum and the malimum because there is a symmetrical pattern present in Table 3. The result from random assignment would be an overall score of 60 quality hours as opposed to 68 hours. It can also be arrived at by averaging the nine overall scores of Table 3 because they are all equally probable under a random assignment. A more complicated approach involves random assignment via a Monte Carlo computer

TABLE 3. THE ALTERNATIVE POSSIBILITIES FOR ALLOCATING 20 HOURS TO TWO JUDGES FOR TWO CASETYPES

#1 The Optimum

	C	N	
F	4 / 8 / 32	3 / 2 / 6	10
W	2 / 0 / 0	3 / 10 / 30	10
	8	12	68

#2

	C	N	
F	4 / 7 / 28	3 / 3 / 9	10
W	2 / 1 / 2	3 / 9 / 27	10
	8	12	66

#3

	C	N	
F	4 / 6 / 24	3 / 4 / 12	10
W	2 / 2 / 4	3 / 8 / 24	10
	8	12	64

#4

	C	N	
F	4 / 5 / 20	3 / 5 / 15	10
W	2 / 3 / 6	3 / 7 / 21	10
	8	12	62

#5

	C	N	
F	4 / 4 / 16	3 / 6 / 18	10
W	2 / 4 / 8	3 / 6 / 18	10
	8	12	60

#6

	C	N	
F	4 / 3 / 12	3 / 7 / 21	10
W	2 / 5 / 10	3 / 5 / 15	10
	8	12	58

#7

	C	N	
F	4 / 2 / 8	3 / 8 / 24	10
W	2 / 6 / 12	3 / 4 / 12	10
	8	12	56

#8

	C	N	
F	4 / 1 / 4	3 / 9 / 27	10
W	2 / 7 / 14	3 / 3 / 9	10
	8	12	54

#9 The Malimum

	C	N	
F	4 / 0 / 0	3 / 10 / 30	10
W	2 / 8 / 16	3 / 2 / 6	10
	8	12	52

NOTES:

1. Table 3 shows the nine possible combinations of hours that would lead to an overall caseload of 20 hours per judge. The Table anticipates 10 hours per judge, 8 hours for criminal and 12 hours in civil cases.
2. The best combination is to give Judge Fox 8 hours of criminal cases per week and 2 hours of civil. Judge Wolf should get 0 hours of criminal and 10 hours of civil. Doing so yields a sum of hours-times-quality products equal to 32 + 6 + 0 + 30 or a total of 68. No other combination gives a higher total.
3. The three numbers in each cell are:
   (a) The number in the upper left corner shows how well each judge does on each casetype on a 1-5 scale.
   (b) The number in the center of each cell shows the number of hours that could be given to each judge on that casetype and be consistent with the other cell allocations, the row totals, and the column totals.
   (c) The number in the lower right corner is the product of the quality score times the hours in the cell. The number beneath the lower right corner of each four-cell table is the sum of the hours-times-quality products in the cells.
4. "C" stands for criminal cases and "N" stands for noncriminal or civil cases. "F" stands for Judge Fox and "W" stands for Judge Wolf.

routine.[5] This approach would also assign the 8 criminal hours and the 12 civil hours to the judges in such a way that each gets 10 hours. The important point to note is that by systematically, rather than randomly, assigning the cases, there is an improvement of 13% in the total of quality hours achieved. In fact, there is always a greater total of quality hours achieved through systematic assignment than through random assignment.

Listing all the alternative possibilities, however, is too laborious a method of determining the optimum alternative. A simpler way of determining the optimum alternative is to insert the coefficients from Table 2 into a linear programming microcomputer routine in response to a series of on-screen questions that ask for the coefficients for each variable in each programming statement. The routine generates the results shown in Table 4. Those results indicate the coefficients in the objective function were 4, 3, 2, and 3. The results also indicate that the right-hand side of the constraints were 10, 10, 8, and 12, respectively. Those scores are useful for indicating that the computer was given the proper input. The output refers to those numbers as the original coefficients and the original right-hand side because both are subject to change in the subsequent sensitivity analysis to see what effect various changes might have on the optimum allocation. The most important section of the results are the optimum assignments which are 8, 2, 0, and 10. They yield a grand total of 68 quality-weighted hours.

The results also show the change slopes for the variables that are assigned 0 hours. Thus variable C or Cell C has a change slope of -2, meaning that whenever Cell C goes up one hour, the grand total goes down 2 points. An alternative way of expressing that change slope or coefficient sensitivity is to say that in order to move Cell C from an assignment of 0 to an assignment of 1, we would have to be willing to accept a reduction of 2 points in the grand total. Further, one could similarly calculate a change slope for the other three variables by adding one hour to the amount of hours originally placed in Cell A, Cell B, and Cell C. Where constraints interfere with a positive rate of return they should be relaxed.

A special version of linear programming, the transportation algorithm, is easier to apply to the judicial assignment problem than other linear programming methodologies. This model treats each

5. The Monte Carlo technique is a randomization system. For information on the program, see E. BUFFA & J. DYER, MANAGEMENT SCIENCE/OPERATIONS RESEARCH: MODEL FORMULATION AND SOLUTION METHODS 471-520 (1981); S. RICHMOND, OPERATIONS RESEARCH FOR MANAGEMENT DECISIONS 433-35 (1968).

TABLE 4. THE RESULTS OF THE LINEAR PROGRAMMING ANALYSIS AND THEIR SENSITIVITY

** RESULTS **

VARIABLE	VARIABLE VALUE	ORIGINAL COEFF.	COEFF. SENS.	CONSTRAINT NUMBER	ORIGINAL RHS
x 1	8	4	0	1	10
x 2	2	3	0	2	10
x 3	0	2	2	3	8
x 4	10	3	0	4	12

OBJECTIVE FUNCTION VALUE: 68

SENSITIVITY ANALYSIS
OBJECTIVE FUNCTION COEFFICIENTS

VARIABLE	LOWER LIMIT	ORIGINAL COEFFICIENT	UPPER LIMIT
x 1	2	4	NO LIMIT
x 2	NO LIMIT	3	5
x 3	NO LIMIT	2	4
x 4	1	3	NO LIMIT

NOTES:

1. The above computer output indicates that the optimum allocation of hours to cells a, b, c, and d is 8, 2, 0, and 10, respectively.
2. Those results are based on the fact that the quality scores for those cells were 4, 3, 2, and 3.
3. The results also recognize the fact that the row-column totals or constraints are 10, 10, 8, and 12.
4. The objective function value is 68 as determined by adding (8 × 4) to (2 × 3) to (0 × 2) to (10 × 3).
5. The column labeled "coefficient sensitivity" tells us the change slope for any cell whose optimum is zero. It says that if Cell C were to go up one unit (i.e., one hour), then the objective function value would go down by two units.

judge as a warehouse or a supplier of goods and each casetype as a destination or a receiver of goods. The numbers in the cells represent distances from the suppliers to the receivers. Thus, each judge has ten units available to supply. The criminal receiver wants 8 units, and the civil receiver wants 12 units. The solution that minimize transportation distances or times takes 8 units from the Fox warehouse for the criminal receiver, 2 units from the Fox warehouse for the noncriminal receiver, and 10 units from the Wolf warehouse for the noncriminal receiver. The advantage of using the transportation algorithm is that one only has to insert into the microcomputer the eight numbers shown in Table 1, not the 24 numbers shown in Table 2. One can generate the optimum solution by programming the microcomputer to maximize the objective function, or the overall sum of the quality scores times the hours of each cell.

II. VARYING THE COEFFICIENTS AND CONSTRAINTS

A. *Varying the Quality Coefficients*

An important benefit of the microcomputer approach to assigning judges to casetypes (or people to tasks) is the additional sensitivity analysis shown in Table 5. The insensitivity range denotes that the quality score of Judge Fox on criminal cases could range from a low of 2 to a high of infinity without affecting the optimum allocation. Actually, because the quality scale is a 1-5 scale, the insensitivity range only covers a range of 2 through 5. To test that insensitivity range, one can insert a 2, 3, or 5 in place of a 4 in Cell A in Table 1, and consider what the value of the new optimum should be. Under any of those changes, Cell A will continue to have a quality score that is as high or higher than Cell C. It would, therefore, still make sense to assign Judge Fox all 8 criminal hours, especially with a score of 3, 4, or 5. Moreover, even with a score of 2, which equalizes Fox and Wolf, one could assign all 8 hours to Fox and 0 to Wolf (or any combination of positive numbers that adds to 8) without affecting the choice of which alternative is best. The changes will only affect the size of the grand total of Alternative #1; it will not affect how the total compares to the other possible grand totals. If the size of the grand total for Alternative #1 is lowered by giving Judge Fox a score lower than 4, then all the grand totals in Table 4 will be lowered. Yet, Alternative #1 will still have the highest grand total. The other insensitivity ranges around the original coefficients in the objective function can be similarly interpreted. This shows that there is substantial room for error in scoring the quality of the judges on each casetype without affecting the optimum allocation.

B. *Varying the Hours per Judge*

With some linear programming problems, there are insensitivity ranges around the numbers that are on the right-hand side of the constraints. That is not the case with this problem because if any of the right-hand numbers, 10, 10, 8, and 12, change, then the allocations will also change. For example, if Judge Fox is no longer expected to have 10 hours of trial time, but instead may have an average of 8, then he will not be able to do 8 hours of criminal cases and 2 hours of noncriminal cases in an average week. The allocations must be consistent with the row and column totals. The row and column totals must also be consistent with each other.

One could ask the sensitivity analysis question of what would be the optimum total hours (other than 10 apiece) to assign Judge Fox and Judge Wolf in light of their four quality scores and the fact that a typical week involves 8 criminal hours and 12 noncriminal hours. As shown in Table 5-A, the optimum allocation for Cells A, B, C, and D is 8, 6, 0, and 6, respectively. This allocation generates a total of 14 hours for Judge Fox and 6 hours for Judge Wolf. Those figures could have been arrived at through a linear programming routine using the same objective function of 4a + 3b + 2c + 3d. Actually, any combination of Cells B and D that add to 12 would result in the maximum objective function of 68 because the quality scores for Cells B and D are the same. In a tie like that, one can split the column total of 12 among the tied cells, although some computer programs may give all 12 hours to the first cell.

When only dealing with two judges and two casetypes, one does not need a computer. One can arrive at the optimum allocations by (1) giving as many hours as possible to the high-scoring cells, (2) giving as few hours as possible to the low-scoring cells, (3) giving an intermediate number of hours to the intermediate-scoring cells, and (4) making small trial and error adjustments after following those three basic rules. If we add a fifth rule that says minimize the number of casetypes per judge, then the allocation would be 8, 0, 0, and 12. That way, no judge would have more than one casetype. We could also add a sixth rule encouraging the assignment of roughly equal judicial caseloads where doing so does not decrease the overall objective function. This rule would cause the optimum allocation to be 8, 2, 0, and 10. Both judges would then have 10 hours apiece while still achieving the maximum objective function of 68 and satisfying the column constraints.

TABLE 5. SOLUTIONS FOR VARIATIONS ON THE CONSTRAINTS

A. Constraints Per Casetype But Not Per Judge

	C	N	
F	4 a 8 32	3 b 6 18	
W	2 c 0 0	3 d 6 18	
	8	12	68*

B. Constraints Per Judge But Not Per Casetype

	C	N	
F	4 a 10 40	3 b 0 0	10
W	2 c 0 0	3 d 10 30	10
			70*

C. Constraint on Total Cases But Not Per Judge or Per Casetype

	C	N	
F	4 a 20 80	3 b 0 0	
W	2 c 0 0	3 d 0 0	20
			80*

D. Constraints Per Casetype, Per Judge, and on a Cell

	C	N	
F	4 a 8 32	3 b 2 6	10
W	2 c 0 0	3 d 10 30	10
	8	12	68*

NOTES:

1. The underlined numbers are the right-hand side of the equality constraints. Thus for Situation A, the only constraints are a + c = 8 and b + d = 12.
2. The italicized number in the lower right corner of each table is the grand total of the sum of the four products of the quality scores multiplied by the assigned hours.
3. The quality scores are in the upper left corner of each cell. The assigned hours are in the middle of each cell. The quality-times-hours products are in the lower right corner of each cell.

C. *Varying the Hours per Casetype*

One could also ask the sensitivity analysis question of what the optimum hours of criminal and noncriminal cases would be if we operate only under the constraints that require Judges Fox and Wolf to be assigned 10 hours each. The answer, as shown in Table 5-B, is 10, 0, 0, and 10. This solution can be reached by dropping the last two rows or constraints of that table using the microcomputer linear programming approach of Table 2. With two judges and two casetypes, one could simply assign each judge 10 hours of whatever casetype he or she does best.[6]

Table 5-B shows 10 criminal and 10 noncriminal hours in the average week instead of 8 and 12. Where a court system lacks judges who are especially competent in certain casetypes or lacks a sufficient quantity of judges regardless of their competency, the system can seek to divert certain casetypes to other methods of resolution. The 12 noncriminal hours per week could, therefore, be reduced to 10 by diverting or discouraging the extra 2 hours per week. Likewise, if the system has an especially competent judge hearing criminal cases, the system could encourage more criminal cases to go to trial. Of course this necessitates raising the original 8 hours per week to 10 hours.

The situation shown in Table 5-B may be the more typical work assignment situation than the one shown in Table 3. Work specialization typically involves each person being assigned to one job rather than to multiple jobs. The assignment algorithm is a variation on the linear programming routine which generally works with exactly as many tasks as there are people to be assigned.

D. *Other Variations*

The question of optimum allocation is reformulated when one drops all constraints per casetype or per judge and yet maintains a total constraint of 20 hours per week. As shown in Table 5-C, the optimum allocation then is 20, 0, 0, and 0. All 20 hours would be assigned to Judge Fox and the 20 hours would only consist of criminal cases. The set of constraints may, however, be unreasonable if they allow for many judges to have no trial caseload at all. Those judges could still do legal research and other activities. Yet, they could rightfully complain that they are not being allowed to be judges if they had no trial caseload. This might be contrary to not only their election mandate, but also to the constitutional/statutory definition of their offices.

The variations are carried one step in the other direction as constraints are added in Table 5-D. This table demonstrates what would happen if in addition to having column constraints for the casetypes and row constraints for the judges, we also add a cell constraint. A logical cell constraint system would be one where any judge who receives a score of 2 or 1 on a particular casetype should not hear any cases of that casetype, because a 2 means the judge is mildly incompetent on that casetype and a 1 means the judge is severely incompetent. With a four-cell table, adding a cell constraint automatically deter-

6. This same analysis can be applied if the constraints are any pair of numbers other than 10 hours assigned to each judge.

mines what the other cells will have to be without requiring any reiterative trial and error to arrive at the optimum allocations. If a 0 is assigned to Cell C in Table 5-D, then the optimum allocation is 8, 2, 0, and 10.

Another variation on Table 1 is to express the hours per judge and/or the hours per casetype in terms of ranges, rather than specific hours. For example, instead of stating that each judge should have 10 trial hours per week, one could express the constraint as "from 5 to 20." That is the same as saying Cell A plus Cell B should be greater than or equal to 5 and less than or equal to 20. Likewise, instead of saying the casetypes should average 8 and 12 hours apiece, one can note that criminal cases range from 3 to 15 trial hours per week, and noncriminal cases range from 8 to 16.

Given that information, what is the best allocation for maximizing the sum of the products of the quality scores times the hours assigned? The answer, arrived at either through the linear programming routine or through logical reasoning is an allocation of 15, 5, 0, and 11. The logical reasoning approach seeks to push the constraints to their zenith because as more hours are allocated, more benefits will be generated, just as in a budgeting problem where one generally lobbies for as big a budget as possible. The maximum number of hours per week would be 31 which involves 15 criminal and 16 noncriminal hours. Of those 31 hours, 20 should be given to Judge Fox because the sum of his quality scores is higher than the sum of Judge Wolf's scores. Of those 20 hours, Cell A must receive 15, the maximum number possible. The other cells are thereby determined to be 5, 0, and 11. Similar reasoning can be applied in arriving at an optimum allocation of hours to judges and casetypes with other numbers and other combinations of constraints such as equations, ranges, and less-than and greater-than inequalities.

E. *Conflicting Constraints*

One can easily introduce conflicting constraints by stating (1) Cell C should be 0 in Table 5-D (because Judge Wolf scored only 2 on criminal cases), (2) Cell D should be 6, meaning that Judge Wolf should get about half the noncriminal cases, and (3) Judge Wolf should have 10 trial hours per week. Whenever there are conflicting constraints, the computer will indicate that the problem is unsolvable. A knowledgeable insider, however, may be able to resolve the conflict by prioritizing the constraints in light of the purposes of the system. For example, it is more important to have 0 in Cell C than 6 in Cell D. Thus, we could drop the Cell D constraint more easily than the

Cell C constraint. The 0 in Cell C is more important for achieving a higher score in the objective function. It is also more important in view of the desire to avoid assigning judges to casetypes that they do not handle well. A similar analysis can be done wherever conflicting constraints occur.

Requiring more hours of work than there is work available is a common type of conflicting constraint that occurs when assigning people to tasks. This constraint can be resolved by reducing the work requirement or by increasing the demand. Or, an alternative method can be utilized if neither one of those approaches is feasible. The objective function can be maximized by completing the available work and having the equivalent of partial layoffs for some of the people. One does not necessarily layoff the person with the lowest overall score, however, because that person might be needed to cover a hard-to-cover specialty. In other words, reduce some of the 10-hour figures, but not by the same amount or percentage for each person or for each judge.

Another common conflict between constraints is to require fewer hours of work than there is work to be done. This conflict can be resolved by each person increasing his work output or by decreasing the demand. If these approaches are not available, then one may selectively add overtime to each individual in order to maximize the objective function. Of course, a new budgetary consideration is added if overtime will require extra pay. Still another alternative is to hire more people. This approach is the opposite of the previous conflict that, as discussed above, can be resolved by firing some personnel.

A third conflicting constraint involves the use of a competency or qualitative constraint. This rule would prohibit people from performing a given task unless they are qualified to do it. Competency is ascertained through a minimum scoring device. This conflict can be resolved by relaxing the qualifications required, or by rejecting requests for work for which qualified people are not available. One can also hire additional people or arrange for overtime for those who are qualified.

In order to maximize the objective function, all three types of conflicting constraints can be resolved by manipulating the inputs into the linear programming routine. One can also experiment, using manual calculations, with a subset of the people and tasks in order to obtain insights into the effects different variations have on these conflicts. The more promising variations can then be tried on a broader spectrum of people and tasks.

III. The Problem of Assigning Six Judges to Four Casetypes

A. *Expressing the Problem*

Table 6 expands the original problem of only two judges and two casetypes to six judges and four casetypes. The new problem is more realistic, and at first glance appears substantially more difficult. The problem involves providing a solution for twenty-four unknowns because six judges and four casetypes require twenty-four cells or allocations. These allocations must also satisfy the six row constraints representing hours per judge, together with the four column constraints representing hours per casetype. The casetypes here are criminal, personal injury, family, and miscellaneous.

The problem can be solved with a microcomputer using a linear programming routine. This would involve an objective function of the form, 4a + 3b + 4c + 3.67d, and so on through 4u + 1v + 4w + 3.00x. Furthermore, the six row and four column constraints must be expressed in the same manner as the two row and two column constraints are expressed in Table 2. Although this is a substantial amount of information, it can be quickly inputted into the computer in response to a series of simple questions concerning the linear programming coefficients that appear on the computer screen.

B. *The Solutions*

Table 7-B sets forth the optimum solution derived from the microcomputer program whereas Table 7-A shows one "trial and error" solution. That solution is found by following the common sense rules noted at the bottom of the table which include (1) giving many hours to cells that have a 5 for a quality score, (2) giving few hours to cells that have a 3 and 0 hours to cells with a 2 or 1, and (3) giving a moderate number of hours to cells that have a 4 for a quality score. The trial and error solution shown in Table 7-A is reached by following those rules, and making adjustments in light of the row constraints and the column constraints.

The overall optimum solution of the objective function is 250 points which is derived from the sum of the products of the quality scores times the hours assigned for each cell. At 246 points, the overall objective function for the trial and error method is extremely close to the optimum solution. One might say that if trial and error can regularly come within 95 percent of the optimum, then it may not be worthwhile to obtain the optimum through a microcomputer program. Furthermore, by adding an additional rule that attempts to

TABLE 6. THE PROBLEM OF ALLOCATING 60 HOURS TO SIX JUDGES FOR FOUR CASETYPES

CASES / JUDGE	CRIMINAL		PERSONAL INJURY		FAMILY		MISCELLANEOUS		Hours per Judge
	Q	H	Q	H	Q	H	Q	H	
BROWN	4	a	3	b	4	c	3.67	d	10
GREEN	5	e	4	f	2	g	3.67	h	10
JONES	4	i	2	j	5	k	3.67	l	10
MILLER	2	m	3	n	4	o	3.00	p	10
SMITH	1	q	4	r	3	s	2.67	t	10
WOOD	4	u	1	v	4	w	3.00	x	10
Hours per Casetype		24		18		12		6	60

NOTES:

1. There are six judges in Table 6 and four casetypes, as contrasted to two judges and two casetypes in Table 1. Table 6 and Table 7, which immediately follows, are thus designed to illustrate the application of the allocation system to a larger problem.
2. The object is to solve for 24 unknowns in order to maximize the sum of the products of the quality scores (Q) times the hours assigned (H) for each judge, subject to the row and column constraints.
3. The miscellaneous quality score is derived by averaging the other three quality scores for each judge.
4. An additional constraint might be that no judge should be assigned any hours for a casetype unless the judge receives a score of at least 3 on the casetype.

TABLE 7. ALTERNATIVE WAYS OF ALLOCATING 60 HOURS TO SIX JUDGES AND FOUR CASETYPES

Cell Letter	Hours Assigned	Quality Score	Quality Times Hours	TOTALS
A. A Trial-and-Error Solution				
a	7	4	28	
b	3	3	9	
e	10	5	50	
k	10	5	50	
n	5	3	15	
o	1	4	4	
p	4	3	12	
r	10	4	40	
u	7	4	28	
w	1	4	4	
x	2	3	6	
				246
B. Optimum Solution				
a	4	4	16	
d	6	3.67	22*	
e	10	5	50	
k	10	5	50	
n	8	3	24	
o	2	4	8	
r	10	4	40	
u	10	4	40	
				250

* Rounded to nearest whole number.

NOTES:

1. The trial and error solution is derived by utilizing rules such as the following:
   (a) Give as many hours as possible to cells that have a quality score of 5.
   (b) Give as few hours as possible to cells that have a quality score of 3.
   (c) Give a middling number of hours to cells that have a quality score of 4.
   (d) The exact number of hours to give as part of rules 1, 2, or 3 depends on the row and column constraints and what has already been allocated or could be allocated.
   (e) In allocating hours to judges, try to minimize the number of casetypes per judge. This makes it easier to find the optimum, and it is in conformity with the idea that specialization is efficient where there are no diminishing returns.
2. The optimum solution is derived with a microcomputer using what is known as the simplex algorithm which guarantees optimality in linear programming situations.
3. Both solutions are feasible because they satisfy the row and column constraints. The optimum solution is better because it generates a sum of the quality-times-hours products equal to 250 rather than only 246.
4. With a reasonably big problem of substantially more than two casetypes and more than two judges, it may be impossible, or at least difficult, and quite time consuming to find the optimum solution through trial and error as contrasted to using the simplex algorithm in computer or noncomputer form.

minimize the number of casetypes per judge, one can easily move from the trial and error solution of Table 7-A to the optimum solution of Table 7-B. The trial and error solution in Table 7-A requires that some judges hear three different types of cases, whereas the optimum solution requires that no judge hear more than two casetypes. Working with a matrix like the one illustrated in Table 6, and adding the additional rule makes it rather easy to find the optimum solution through trial and error.

IV. Some Conclusions

One should not interpret the analysis above as an indication that the microcomputer approach is not worth pursuing. It may be quite worthwhile in substantially larger problems, such as one involving 100 lawyers in a huge law firm with 40 casetypes or assigning 435 members of Congress to 30 committees. Of course, unless one uses a computer program, or the simplex algorithm behind the program, that guarantees an optimum solution, one can never be sure that one has found the optimum or is within five percentage points. The main benefit of the trial and error approach is that it does provide the user with insights into the analysis which are missed by simply accepting the computer output. On the other hand, the big advantage of the computer output is not that it guarantees an optimum, but rather that microcomputers can do sensitivity analysis extremely well. Although one may be able to arrive at the optimum solution through trial and error, it is far more difficult to arrive at the lower limit and the upper limit of the insensitivity range for every quality coefficient. It is also quite difficult to arrive at change slopes or coefficient sensitivity scores for every quality coefficient.[7]

The computer is also quite helpful in quickly demonstrating how the optimum allocation would change if alterations are made in the constraints per judge, per casetype, the hours assigned to any cell, or the overall quantity of hours. The computer is at its best not so much in finding the initial solution, but in providing useful information on how sensitive that solution is to various changes in the coefficients and constraints. By using that capability, we can arrive at an optimum allocation in light of a possible optimum set of constraints. Thus, the computer analysis not only provides answers, but it also enables us to improve the questions we ask.

Using management science to assign judges to types of cases may enable court systems to operate more effectively, efficiently, and equi-

7. *See* Table 4, *supra* p. 1325.

tably. Effectiveness is improved by assigning judges to casetypes for which they are best equipped. This is measured by the objective function of maximizing the products of the quality scores times the hours assigned. Efficiency is improved by building into the quality score the ability to decide cases quickly and thus reduce delay and expense in the system. Equity is improved by treating judges, whose interests can also be taken into consideration in the quality scores, more fairly in proportion to their abilities. To the extent that equity also means equal treatment, the optimum assignment rules can include equality constraints as well. Not only is there a need for an increased use of linear programming for optimum assignment of judges to casetypes, but there is also a need for greater use of common sense ideas on which the programming routines are based.