Flexible Resource Allocation: A Comparison Of Linear Diophantine Analysis And Integer Programming

Robert F. Cope III, (E-mail: rcope2@selu.edu), Southeastern Louisiana University
Rachelle F. Cope, (E-mail: rcope@selu.edu), Southeastern Louisiana University
Avinash M. Waikar, (E-mail: awaiker@selu.edu), Southeastern Louisiana University

ABSTRACT

To help production managers cope with an ever changing and complex business environment, we investigate a flexible methodology for solving integer resource allocation problems. Solutions obtained to an example problem using the Linear Diophantine Equation (LDE) methodology presented are compared to solutions produced using Integer Programming. Tradeoffs are examined and discussed, and suggestions are made for managers facing resource decisions similar to the example studied.

INTRODUCTION

Businesses face increased competition characterized by customer demand for ever-improving quality, just-in-time delivery, increasing innovation, and ongoing innovation. Thus, today's production managers must make sound decisions in a variety of different areas of business. Rapid changes in customer requirements and quick delivery requirements also make it necessary that alternative solutions to resource allocation problems be available to managers. Alternate solutions may impact a company's revenues, operating expenses and more importantly customer satisfaction. Hence, we investigate a methodology to solve an integer resource allocation problem expressed in a structurally equivalent Linear Diophantine Equation (LDE).

AN EXAMPLE INTEGER RESOURCE ALLOCATION PROBLEM

Continuous process industries, such as paper and rolled steel, generally inventory their product in batches. The batches are then subdivided and distributed to order as finished products. Under just-in-time conditions, the subdivision of batch products can become inefficient when shocks get introduced into the supply chain. To manage such shocks, a production manager needs flexibility to continue efficient resource allocation. Therefore, methodologies to provide alternate solutions are valuable additions to the decision-making capability of a production manager. To illustrate, we explore the following integer resource allocation problem.

A production manager has several 1000 ft. batch rolls of kraft paper (or rolled steel) that must be subdivided into finished products. Certain customers, in general, require the finished products in rolls of 60 ft. and 80 ft. lengths. In how many different ways can this task be done?

A typical model used to represent the problem is the linear equation:

$$60X_1 + 80X_2 = 1{,}000, \quad (1)$$

where X_1 and X_2 respectively represent the number of rolls with 60 ft. and 80 ft. lengths and the solution set for the problem is restricted to a subset of integers.

Briefly, the LDE approach advanced by the authors to find the general solution to integer resource allocation problems requires:

- Step 1. An initial solution to the LDE be found,
- Step 2. A solution to the LDE's corresponding Homogeneous LDE (HLDE) be obtained, and
- Step 3. The Additive Theorem be employed to obtain the general solution.

For reference purposes, the LDE methodology is presented in detail in Appendix 2.

To begin the first step in solving the LDE for the production manager's problem, we use Blankinship's Version of the Euclidean Algorithm (BVEA) to find a solution to the LDE's corresponding greatest common divisor (gcd) equation. Employing the BVEA requires we define the matrix M below.

$$M = \begin{vmatrix} 60 & 1 & 0 \\ 80 & 0 & 1 \end{vmatrix}$$

Table I summarizes the actions of the BVEA.

Table I

BVEA Applied Using
$A_1 = 60$ and $A_2 = 80$

Pass i	Operator	Operand	Q	Matrix M_i	Compute
1	60	80	1	60 1 0 20 -1 1	for j = 1 to 3 $M_{2j} - Q*M_{1j}$
2	20	60	3	0 4 -3 20 -1 1	$M_{1j} - Q*M_{2j}$ for j = 1 to 3

From Table I we find the gcd of $A_1 = 60$ and $A_2 = 80$ is 20, and a solution to the gcd equation $60X_1 + 80X_2 = 20$ is $X_1^* = -1$ and $X_2^* = 1$.

To complete the first step, we find an initial solution to our integer allocation problem modeled by $60X_1 + 80X_2 = 1{,}000$. The value of k is found, as $k = C/D = 1{,}000/20 = 50$. Thus, the initial solution to the LDE is $X_1^{**} = kX_1^* = 50*(-1) = -50$ and $X_2^{**} = kX_2^* = 50(1) = 50$.

In the second step, we determine a solution to the HLDE:

$$60X_1 + 80X_2 = 0. \quad (2)$$

Since we have no coefficients equal to ±1, the *Unit Coefficient Reduction Algorithm* must be used. In addition, the algorithm requires the coefficients of the HLDE to be relatively prime. This condition is satisfied by factoring 20 from the coefficients of the HLDE to yield $3X_1 + 4X_2 = 0$. Table II contains the details of the algorithm.

Table II

Unit Coefficient Reduction Algorithm
Applied to $3X_1 + 4X_2 = 0$

HLDE	Iteration	Step	Action
$3X_1+4X_2 = 0$	1	1	Min = $A_1 = 3$ SMin = $A_2 = 4$
		2	By Division $4 = 3q + r$ $q = 1; r = 1$
$3(X_1'-X_2)+4X_2 = 0$ $3X_1'+X_2 = 0$		3	$X_1 = X_1' - X_2$ Substitute Simplify $r = 1$

Since $3X_1' + X_2 = 0$ has a unit coefficient, it can be solved as:

$$X_2 = -3X_1'. \quad (3)$$

By substituting $X_1 = X_1' - X_2$ and employing equation 3, we obtain:

$$X_1 = 4X_1'. \quad (4)$$

Thus, equations 3 and 4 for an arbitrary X_1' yield a solution to the HLDE.

In the third step, the general solution to the LDE is found by adding the initial LDE solution, X_1^{**} and X_2^{**}, to the HLDE solution of equations 3 and 4, respectively, to obtain:

$$X_1 = 4X_1' - 50, \text{ and} \quad (5)$$

$$X_2 = -3X_1' + 50. \quad (6)$$

The mathematical model for the problem requires the resource values of X_1 and X_2 to be nonnegative. Thus, the inequalities $4X_1' – 50 \geq 0$ and $-3X_1' + 50 \geq 0$ are solved. Applying basic algebra yields $X_1' \geq 12.5$ and $X_1' \leq 16.3$, and the integer values satisfying these inequalities are X_1' = 13, 14, 15, and 16.

Using these four values for X_1', and the general formulas for X_1 (equation 5) and X_2 (equation 6) yields the following four sets of integer resource allocation solutions:

- X_1 = 2-60 ft. rolls and X_2 = 11-80 ft. rolls,
- X_1 = 6-60 ft. rolls and X_2 = 8-80 ft. rolls,
- X_1 = 10-60 ft. rolls and X_2 = 5-80 ft. rolls, or
- X_1 = 14-60 ft. rolls and X_2 = 2-80 ft. rolls.

Thus, the LDE approach allows a production manager to efficiently subdivide a batch roll four different ways, providing the production manager with built-in flexibility that he might not have known prior to performing this analysis. Such flexibility can allow for the absorption of unforeseen shocks to the system.

COMPARISON TO TRADITIONAL SOLUTION METHODS

We compare the LDE methodology to an equivalent integer resource allocation problem formulated as an Integer Programming (IP) problem. For reference purposes, completed tableaus of the IP process can be found in

Appendix 3. The IP approach yielded the first solution provided by the LDE approach: 2-60 ft. rolls and 11-80 ft. rolls. In an attempt to obtain an alternate solution using the IP approach, the initial cutting plane strategy was altered to pivot around a different corner, namely variable X_1. The alternate pivot yielded the following solution: 14-60 ft. rolls and 2-80 ft. rolls. This was the fourth solution provided by the LDE approach. To obtain other solutions, strategies involving a "seek and destroy" methodology would have to be initiated with no guarantee of finding other integer solutions, indicating rigidity with the cutting plane technique.

Thus, a limitation of the IP approach to solving integer resource allocation problems has been revealed. The use of cutting plane algorithms to solve IP problems offers little control in obtaining a solution, much less full sets of solutions. The use of these pivoting algorithms may allow the search to cut alternative feasible region boundaries, but provides no mechanism for obtaining alternative solutions without an exhaustive search.

ISSUES AND IMPLICATIONS OF THE LDE METHODOLOGY

Continuing with the example problem, suppose a customer does not want to buy a complete batch roll cut to his requirements. Further, suppose the production manager generally stocks 50 ft. rolls for customers. The LDE approach is not only flexible with its resource allocation, but is also flexible in its use. The production manager can simply model this scenario as:

$$60X_1 + 80X_2 + 50X_3 = 1{,}000, \quad (7)$$

where X_3 represents the number of 50 ft. rolls to be cut from the batch roll. The LDE methodology would then be executed with a new set of solutions available for the production manager to analyze. For this example, the gcd would be 10 with one possible solution set being: $X_1 = 5$, $X_2 = 5$ and $X_3 = 6$. Here, the customer would receive 5-60 ft. rolls and 5-80 ft. rolls from the batch roll, while the production manager increases his inventory by 6-50 ft. rolls and incurs an inventory charge.

Another use for the LDE approach pertains to re-cuts. Suppose a new customer wants to order finished product rolls in 25 ft. and 40 ft. lengths. The production manager now has two options for applying the LDE approach. First, he can re-cut the inventoried 50 ft. rolls, but incurs scrap costs when re-cutting to 40 ft. lengths. The simple LDE for this option would be:

$$25X_1 + 40X_2 = 50. \quad (8)$$

Second, he can set-up a new 1000 ft. batch roll to be cut into the newly specified lengths. The LDE for this option would look like:

$$25X_1 + 40X_2 = 1{,}000. \quad (9)$$

Solutions to the LDE of equation 9 would not incur scrap, allowing the production manager to efficiently use the 1,000 ft. batch roll. For this example, the gcd would be 5 with multiple solution sets available for analyses. One possible solution set, $X_1 = 8$ and $X_2 = 20$, provides the customer with 8-25 ft. rolls and 20-40 ft. rolls.

Production managers are generally considered to be efficient planners. We believe that many would welcome the opportunity to look into the future and plan for inventory and scrap charges, making the LDE methodology an extremely useful and flexible tool.

CONCLUDING REMARKS

Hence, we have presented production managers with an alternative to an IP optimization-based solution. If no single optimum solution is required, the LDE methodology does have the flexibility of providing all possible alternate solutions to an integer resource allocation problem. Choosing the LDE approach allows production managers to perform their own sensitivity analysis, maybe even satisfying other important objectives that are overlooked by the use of optimization-based methods. For our example, shocks on the amount of finished product rolls needed in the supply chain can be absorbed through an alternate efficient cut of the batch rolls.

Another application could be a real estate developer who has a parcel of land with a certain length of frontage property. He must subdivide it into lots having small frontages and large frontages. Small frontages may be easier to sell, but large frontages might bring more profit. In a practical sense, the modeling of problems as LDEs in many cases can provide a more realistic view of decision tradeoffs.

The contributions of exploring this topic are threefold. First, a methodology is presented to confront problems requiring alternate solutions incurred by production managers. Many in academia realize that there is a need to incorporate multiple solutions to integer resource allocation problems in materials management literature. Solutions to integer allocation problems are not usually presented in great detail in most quantitative analysis texts (Kurosaka, 1986).

Second, in exploring this alternative algorithmic approach to solving LDEs, it has been found that a general solution to enumerate all possible integer solutions can be formulated for any LDE that has an initial solution. Naturally, the author’s advancement of a general solution to enumerate all possible LDE solutions is beneficial for any application in which multiple solutions are desirable.

The third purpose involved comparing the methodology of the LDE approach to the results obtained by solving the same resource allocation problem as an IP problem. Previously, problems in which integer solutions have been desired have traditionally been explored as IP problems (Elmaghraby and Elimam, 1980). By utilizing a cutting plane algorithm to solve the problem, it was shown that the more traditional approach does not provide the flexibility to production managers that the LDE method provides.

REFERENCES

1. Barnett, I.A., *Elements of Number Theory*. Prindle, Weber & Scmidt, Inc., Boston, Massachusetts, 1969.
2. Blankinship, W.A., A New Version of the Euclidean Algorithm, *American Mathematical Monthly*, Vol. 70 September, pp. 742-745, 1963.
3. Elmaghraby, S.E. and Elimam, A.A., Knapsack-Based Approaches to the Makespan Problem on Multiple Processors, *AIIE Transactions*, Vol. 12 Issue 1 (March), pp. 87-96, 1980.
4. Greenberg, H., *Integer Programming*. Academic Press, New York City, New York, 1971.
5. Gomory, R., An Algorithm for the Mixed Integer Problem, RM-2597 Rand Corporation (July), 1960a.
6. ——————, All Integer Programming Algorithm, RC-189 IBM Research Center Report (January) 1960b; also in Muth and Thompson (Ed. 1963) *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, New Jersey.
7. Hu, T.C., *Integer Programming and Network Flows*. Addison-Wesley, Reading, Massachusetts, 1969.
8. Ibaraki, T. and Katoh N., *Resource Allocation Problems: Algorithmic Approaches*. The MIT Press, Cambridge, Massachusetts, 1988.
9. Kaplan, W., *Ordinary Differential Equations*. Addison-Wesley, Reading, Massachusetts, 1958.
10. Kurosaka, R.T., Diophantine Equations, *Byte Magazine*, Vol. 11 March, pp. 343-350, 1986.
11. MacDuffee C.C., *Theory of Equations*. John Wiley, New York City, New York, 1954.
12. Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.
13. Rowe, N.C., Diophantine Inference on a Statistical Database, *Information Processing Letters*, Vol. 18 Issue 1 (January 20), pp. 25-31, 1986.
14. Stewart, B.M., *Theory of Numbers*. The MacMillan Company, New York City, New York, 1952.
15. Stewart, I., The Riddle of the Vanishing Camel, *Scientific American*, Vol. 266 (June), pp. 122-124, 1992.
16. Yan, S.Y., *Number Theory for Computing*, 2nd edition. Springer, Berlin, Germany, 1998.

APPENDIX 1: DIOPHANTINE EQUATIONS

A Diophantine equation is a rational integral equation in which the constants are integers and the solutions, or values of the variables that satisfy the equations, are integers (Greenberg, 1971). The word "Diophantine" is derived from the name of Diophantus (in the years 200-284) of the great city of Alexandria. He was one of the first to study equations in integers, and is referred to as the father of Algebra (Yan, 1998). He is perhaps best known as the writer of the book *Arithmetica* that has stimulated the study of Algebra, or more specifically, Diophantine Analysis (Yan, 1998).

A Diophantine equation may take on many different forms and either have no solution, a finite number of solutions, or an infinite number of solutions. In the infinite case, the solutions may be given in terms of one or more integer parameters. For our use, we employ *linear* Diophantine equations (*L*DE) and restrict them to positive values since resources are generally considered to be "real" and available.

Similar to other forms of Diophantine equations, an LDE is a linear equation with solutions in the set of integers (Barnett, 1969). Unlike most other Diophantine equations, LDEs can be solved algorithmically (Ibaraki and Katoh, 1988; Rowe, 1986; Stewart, 1992). Therefore, resource allocation problems may be analyzed as LDEs, and if a finite or infinite number of solutions exist, resource tradeoffs can be examined.

APPENDIX 2: THE LDE METHODOLOGY

The methodology developed here to find the general solution of an LDE involves three steps. Step one involves finding an initial solution to the LDE using Blankinship's Version of the Euclidean Algorithm as its basis. If an initial solution exists, it can be used to find all possible integer solutions. In the second step, a solution to the LDE's corresponding Homogeneous equation (HLDE) is required. Finally, we show that the general solution of the LDE is found by advancing the initial solution of the LDE found in step one with the solution to the HLDE obtained in step two through addition (general solution = initial LDE solution + HLDE solution).

A2.1 Theorems from Number Theory

The methodology is based on and begins with two theorems found in elementary number theory textbooks. Together they provide for the existence and determination of solutions to LDEs (Barnett, 1969; Stewart, 1952). In brief, the selection of D, the greatest common divisor (gcd) of an LDE, is important for finding an initial solution.

Theorem 1: Let $A_1X_1 + A_2X_2 + ... + A_nX_n = C$ (for $n \geq 2$) represent an LDE, and D represent the gcd of A_1, A_2, A_n. If D divides C, then the LDE has a solution in the set of integers.

Theorem 2: Assume the LDE $A_1X_1 + A_2X_2 + ... + A_nX_n = C$ has a solution, and D is the gcd of A_1, A_2, A_n. If X_1^*, X_2^*, ..., X_n^* represents a solution to the gcd equation $A_1X_1 + A_2X_2 + ... + A_nX_n = D$, then $X_1^{**} = kX_1^*$, $X_2^{**} = kX_2^*$, ..., $X_n^{**} = kX_n^*$ represents a solution to the LDE, where $k = C/D$.

A2.2 Blankinship's version of the Euclidean Algorithm for finding an initial LDE solution

In the previous subsection, Theorem 2 implied that to find an initial solution to an LDE one would first solve its related gcd equation $A_1X_1 + A_2X_2 + ... + A_nX_n = D$, where D is the gcd of A_1, A_2, ..., A_n. To assist with the selection of D, W.A. Blankinship's (1963) article unveiled a method for finding the gcd for more than two integer variables, and a solution to the corresponding gcd equation of *n* integers. Blankinship's Version of the Euclidean Algorithm is straight forward to execute and easily implemented using a computer.

Suppose we are interested in finding the gcd of A_1, A_2, ..., A_n and solving the gcd equation $A_1X_1 + A_2X_2 + ... + A_nX_n = D$. According to Blankinship's algorithm we first formulate an *n* by *(n+1)* matrix M defined as:

$$M = \begin{vmatrix} A_1 & 1 & 0 & 0 & . & . & 0 \\ A_2 & 0 & 1 & 0 & . & . & 0 \\ A_3 & 0 & 0 & 1 & . & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ A_n & . & . & . & . & . & 1 \end{vmatrix}$$

The basis of this algorithm consists of performing elementary row operations on the matrix according to the following procedure:

Step #1: Select the row with the smallest non-zero first element. Call this row the *operator row* and its first element the *operator*.

Step #2: Find all rows with non-zero first elements. Call these rows *operand rows* and their first elements *operands*.

(a) If one or more *operand rows* are found, then go to Step #3.

(b) If no *operand rows* are found the process terminates with the solution set for the problem found in the *operator row*. The gcd D is the *operator* while the solution to the gcd equation defined by X_1^*, X_2^*, ..., X_n^* is found in the corresponding columns *2* to *(n+1)* of the *operator row*.

Step #3: For each *operand row* determine the integer quotient Q equal to the row's *operand* divided by the *operator*.

Step #4: For each *operand row* replace each element of the row with the corresponding difference between itself and Q times the corresponding element of the *operator row*.

Step #5: Return to Step #1.

A2.3 Solving the HLDE

By definition, an HLDE is an LDE with its constant term, C, on the right hand side of the equation equal to zero. The second step of the process involves finding its solution. First, if one of the variable coefficients of the HLDE is ±1, a solution can be found by solving for that variable. If no unit coefficients exist in the HLDE, the *Unit Coefficient Reduction Algorithm* can be used to resolve the issue (MacDuffee, 1954).

The process that follows defines the *Unit Coefficient Reduction Algorithm*. First, assume the HLDE $A_1X_1 + A_2X_2 + \ldots A_nX_n = 0$ with $A_i \neq 0$ for i = 1 to n, and where the gcd of A_1, A_2, ..., A_n = 1. To invoke, follow these four steps.

Step #1: Choose Min = A_m to be the smallest A_i for i = 1 to n. Then let SMin = A_s represent the smallest of the remaining A_i's that is relative prime to Min (i.e. has gcd of 1).

Step #2: Apply the division algorithm and find the quotient, q, and the remainder, r, using Min as the divisor and SMin as the dividend. Namely, find integers q and r such that:

SMin = q*Min + r, where $0 \leq r < \text{Min}$.

Step #3: Substitute $X_m = X_m' - q*X_s$ in the HLDE. It is easily seen that this substitution yields a coefficient of r for X_s. Furthermore, repeated applications of Step #3 leads to the desired result of r = 1 or a unit coefficient for the HLDE.

Step #4:

(a) If r is not 1, then return to Step #1.

(b) If r = 1, then a solution to the HLDE can be found by doing the following.

- First, solve the HLDE for the variable that has a unit coefficient.
- Next, utilize the substitutions performed in Step #3 and the equation obtained in (1) above to determine the general solution of the HLDE.

A2.4 Advancing the Methodology through the Additive Theorem

Finally, the methodology for finding a general solution to an LDE is completed by combining the initial solution of the LDE to the solution of its HLDE. The process formulated is dependent on Theorem 3 below, which is based on a similar concept used to solve ordinary linear differential equations (Kaplan, 1958).

Theorem 3 (Additive Theorem): If $I_1, I_2, ..., I_n$ represents an initial solution to the LDE, $A_1X_1 + A_2X_2 + ... + A_nX_n = C$, and $G_1, G_2, ..., G_n$ represents a solution to its HLDE, $A_1X_1 + A_2X_2 + ... + A_nX_n = 0$, then $X_i^* = G_i + I_i$, for i = 1 to n, represents a general solution to the LDE $A_1X_1 + A_2X_2 + ... + A_nX_n = C$.

The proof for Theorem 3 is straight forward and can be obtained by contacting the lead author.

APPENDIX 3: SOLVING THE EXAMPLE PROBLEM USING PRIMAL INTEGER PROGRAMMING

There are many software packages on the market to solve IP problems, and many of these packages generally employ a branch-and-bound algorithm. Branch-and-bound procedures are basically enumeration schemes where certain solutions are discarded by showing that the value of the objective obtained with solutions from a particular class are larger than a provable lower bound. This lower bound is greater than or equal to the value of the objective function obtained earlier (Pinedo, 1995). Many have considered this form of solution methodology to be rigid. Additionally, offering an initial solution to a software package in an attempt to obtain an alternate solution on another branch is most often useless.

More traditional methodologies for solving IP problems employ a cutting plane technique based on the work of Gomory (1960a; 1960b). In addition, some software packages do considerable preprocessing on linear integer models adding constraint cuts to restrict the non-integer feasible region. The claim is that this technique greatly improves solution times for most IP models.

According to Hu (1969), the term "primal" applied to an IP algorithm denotes a method that proceeds to an optimal solution through a sequence of successively better solutions that are all feasible in the sense of satisfying both the integer and the linear restrictions on the variables. One of the prospective advantages of a primal algorithm is the possibility of stopping calculations before an optimal solution has been obtained and using the best solution that has been generated.

A general IP resource allocation problem is of the form:

Maximize/Minimize: $z(x) = f(C_1{*}X_1, C_2{*}X_2, ..., C_n{*}X_n)$

Subject to: $\sum_{j=1}^{n} C_j * X_j = N$

$X_j \geq 0$ (for j = 1, 2, ..., n).

Following this format, we formulate our example problem as a primal IP problem below.

Maximize/Minimize: $60X_1 + 80X_2$

Subject to: $60X_1 + 80X_2 = 1000$

for $X_1 \geq 0$ and $X_2 \geq 0$.

The initial tableau for the problem is illustrated in Tableau I.

Tableau I

Source Row: X_3
Pivot Column: $-X_2$
λ: 80

	1	**$-X_1$**	**$-X_2$**
X_0	0	-60	-80
X_1	0	-1	0
X_2	0	0	-1
X_3	1000	60	80
S_1	12	0	1

In Tableau II, note that an optimal solution has been found at $X_1 = 2$ and $X_2 = 11$ indicating that the production manager's batch roll has been subdivided into 2-60 ft. rolls and 11-80 ft. rolls. The objective function has been maximized to 1000 ft. indicating that the entire batch roll has been used.

Tableau II

Termination: Objective Function Value = 1000 ft.
Solution: $X_1 = 2$, $X_2 = 11$

	1	**$-S_4$**	**$-S_3$**
X_0	1000	20	0
X_1	2	3	-4
X_2	11	-2	3
X_3	0	-20	0
S_5			

In an effort to obtain an alternative solution, a different cutting plane strategy was used. It involved initially cutting around the X_1 variable (corner). Tableau III provides an alternate optimal solution at $X_1 = 14$ and $X_2 = 2$ indicating that the batch roll has been subdivided into 14-60 ft. rolls and 2-80 ft. rolls. The objective function has been maximized to 1000 ft. indicating that the entire batch roll has been used again.

Tableau III

Termination: Objective Function Value = 1000 ft.
Solution: $X_1 = 14$, $X_2 = 2$

	1	**$-S_1$**	**$-S_2$**
X_0	1000	0	20
X_1	14	4	-1
X_2	2	-3	1
X_3	0	0	-20
S_3			

Attempts to obtain other solutions would involve a "seek and destroy" strategy with no guarantee of finding other solutions.

NOTES