AMO - Advanced Modeling and Optimization, Volume 19, Number 1, 2017

A Spreadsheet Approach to Solving NLP and ILP Formulations of a Modified Partition Problem

Lonnie Turpin, Jr.[*,†] Vipin Menon[‡] Gerard Ornas, Jr.[§]

Abstract

Given a countable set of positive integers under a proposed set of conditions, we develop and analyze both nonlinear programming (NLP) and integer linear programming (ILP) formulations for a strict partition into $N = 2$ subsets with equal summation. This note adds to the list of operations research (OR) applications in spreadsheet modeling by constructing each of these formulations in a manner that makes use of the readers intuition without the use of formal proofs, and uses available Excel spreadsheet functions without other advanced software. As such, we hope this is a useful classroom tool as well as a general interest article on optimization for a broader audience.

Keywords: Partition Problem, Spreadsheet Model, Nonlinear Programming, Integer Linear Programming

1 Introduction

Optimization is becoming increasingly important in curricula ranging from mathematics and computer science to business and economics. Tools such as Microsoft Excel offer an attractive alternative to specialized software primarily because of their ubiquitous nature. Excel is key not only for the ever growing need for spreadsheet modeling skills

[*]Corresponding author.

[†]Department of Management, Marketing, and Business Administration. McNeese State University, Lake Charles, Louisiana, 70609. Email: lturpin@mcneese.edu.

[‡]Department of Electrical Engineering and Computer Science. McNeese State University, Lake Charles, Louisiana, 70609. Email: vmenon@mcneese.edu.

[§]Department of Mathematical Sciences. McNeese State University, Lake Charles, Louisiana, 70609. Email: gornas@mcneese.edu.

AMO - Advanced Modeling and Optimization. ISSN: 1841-4311

required for any business discipline, but because it offers a simplified interface without the complexity associated with these specialized programs. Noteworthy research by [15] provides useful guidelines for using spreadsheets to teach students about specific operations research (OR) models, while [5] discusses the efficacy of the spreadsheet approach to operations research education in general.

Within OR, Excel has been used to solve widespread problems such as the multi-criteria decision problem [18] as well as more complicated problems as in the resource-constrained project scheduling problem [20], among others. In this research, we add to the list of OR applications by constructing formulations for a modified partition problem in a manner that

1. makes use of the readers intuition without the use of formal proofs (to make it palatable for cross-disciplines); and
2. uses available Excel spreadsheet functions without visual basic (VBA) or advanced software.

The partition problem, known in the form of the number partitioning problem (NPP) and the set partitioning problem (SPP), is a classic NP-complete problem (in NPP) or NP-hard (in SPP) in computer science with the task of deciding whether a given multiset (NPP) or set (SPP) of positive integers can be partitioned into N finite subsets such that the difference in sums of the numbers in each subset are minimized. For $N = 2$, a perfect partition is realized when the difference is 0 (if the sum of all the integers is even) or 1 (if the sum of all the integers is odd). Many methods exist to solve both types of problems including the Kamarkar-Karp [10] differencing heuristic (see [4] for a complete analysis and [12] for a modified version), the use of constrainedness to locate phase transition behavior [7], optimization via simulated annealing [9], constraint programming [14], and tree search [17], among others. For practical applications of each, see [1] and [19]. Of specific interest to this research is the work [11] which proposes a generalized NPP for $N = 2$ as well as an integer linear programming formulation of the problem. In the following sections, we use similar logic and show both nonlinear programming (NLP) and integer linear programming (ILP) formulations for a special case of an SPP.

To begin, let Ω represent a grand set. This research presents NLP and ILP formulations to partition Ω into $N = 2$ sets X_1 and X_2 where $\sum X_1 = \sum X_2$. For exposition, we give the cardinality $|\Omega| \gg N$ to prevent loss of generality, and consider the following assumptions (filters on the data set).

Assumption 1. *To create the input numbers, we let every element ω in Ω be a random integer in $\{\omega \in \mathbb{Z}_{>0} \mid 1 \leq \omega \leq U\}$ for all $\omega \in \Omega$, where U is a definable finite upper* *bound.*

Assumption 2. *The sum of Ω is a multiple of N, expressed as $\sum_{\omega\in\Omega}\omega \in N\mathbb{Z}_{>0}$. For brevity, we write $\sum\Omega \in N\mathbb{Z}_{>0}$.*

Assumption 3. *Each element is unique, that is no two elements in Ω are identical. Formally, the multiplicity m of any element is one, expressed as $m_\omega = 1$ for all $\omega \in \Omega$.*

Should Assumption 3 be suspended, Ω could be analyzed as a multiset (see [3]). It should be clear that $U > |\Omega|$ to satisfy Assumption 3. The logic is easily shown by contradiction. Suppose $U = 25$ and $|\Omega| = 26$. It is impossible to create a data set with each element satisfying the integer interval $[1, 25]$ without repeating at least one number.

Assumption 4. *No element may be assigned to more than one set expressed as $\{\bigvee_N \omega \in X_N, \omega \in \Omega\}$. By Assumption 3, this results in the implication $\bigcap_N X_N = \emptyset$.*

This paper is organized as follows. Section 2 presents the integer linear and nonlinear programming models for $N = 2$ subsets. Section 3 provides the Excel formulations of the two models in a four-step procedure where the final step gives a brief overview of the solutions. Finally, Section 4 concludes the paper with a general discussion.

2 Formulations for $N = 2$

Let ω_i represent the i-th number assigned. We begin by assigning the largest value to the first set X_1, that is forcing $\omega_1 = \max_{\omega\in\Omega}\omega$ and assigning to X_1. Define Ω' as the remainder of Ω after ω_1 as $\Omega' = \Omega - \max_{\omega\in\Omega}\omega$.

Remark 1. *It should be obvious that $\max_{\omega\in\Omega}\omega \le \sum\Omega/2$ for a perfect partition to exist. For clarity, consider the contradiction $\max_{\omega\in\Omega}\omega > \sum\Omega/2$. This directly implies $\sum X_1 > \sum X_2$, resulting in no solution. Following Assumption 3 and generalizing for $N > 2$ results in the bounds $|\Omega| < U \le \sum\Omega/N$.*

Now let $\lambda_\omega \in \{0, 1\}$ represent the decision variables where $\lambda_\omega = 1$ if $\omega \neq \max_{\omega\in\Omega}\omega$ is assigned to X_1, and $\lambda_\omega = 0$ if not. The binary nature of the variables satisfies Assumption 4. By the preceding logic, we replace ω_1 with $\max_{\omega\in\Omega}\omega$ to define $\max_{\omega\in\Omega}\omega + \sum_{\omega\in\Omega'}\lambda_\omega\omega \equiv \sum X_1$ and correspondingly set $\sum X_2 \equiv \sum\Omega/2$ to satisfy the partition. Considering the known inputs $\sum\Omega/2$ and all $\omega \in \Omega$, we present the argument

$$\min \left| \left(\max_{\omega\in\Omega}\omega + \sum_{\omega\in\Omega'}\lambda_\omega\omega \right) - \frac{\sum\Omega}{2} \right| \quad \text{s.t.} \quad \lambda_\omega \in \{0,1\}, \quad \text{for all} \quad \omega \in \Omega' \tag{NLP}$$

allowing us to determine the optimal partitioning of Ω into X_1 and X_2.

Remark 2. *Removing the absolute value bounds on the formulation in (NLP) would force the decision variable solution $\lambda_\omega = 0$ for all $\omega \in \Omega'$. The proof is straightforward and therefore omitted.*

The absolute value argument in (NLP) means that this objective is not linear, but it may undergo a transformation into a linear objective with additional linear constraints and additional variables. Because of modulus, it is clear that one set of variables is not enough to make the objective function linear. We introduce the new decision variables $\psi_\omega \in \{0,1\}$ representing whether or not ω is assigned to X_2 for all $\omega \in \Omega'$. It is immediately clear that $\psi_\omega \equiv 1 - \lambda_\omega$ for all $\omega \in \Omega'$. We set $\sum_{\omega\in\Omega'} \psi_\omega\omega \equiv \sum X_2$ and maintain $\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega\omega \equiv \sum X_1$.

We can now transform of the nonlinear system in (NLP) to a linear system via the following program in (ILP).

$$
\begin{aligned}
\min \quad & \left(\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega\omega\right) - \sum_{\omega\in\Omega'} \psi_\omega\omega \\
\text{s.t.} \quad & \left(\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega\omega\right) - \sum_{\omega\in\Omega'} \psi_\omega\omega \;\geq\; 0 \\
& \max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega\omega \;\leq\; \frac{\sum\Omega}{2} \\
& \sum_{\omega\in\Omega'} \psi_\omega\omega \;=\; \frac{\sum\Omega}{2} \\
& \psi_\omega \;=\; 1-\lambda_\omega, \quad \text{for all} \quad \omega\in\Omega' \\
& \lambda_\omega \;\in\; \{0,1\}, \quad \text{for all} \quad \omega\in\Omega'
\end{aligned}
\tag{ILP}
$$

The mechanics are relatively straightforward, so we do not need to give a formal proof. Instead, we opt for a basic explanation. Let the bounds on the decision variables be $\lambda_\omega, \psi_\omega \in \{0,1\}$ for all $\omega \in \Omega'$. To begin, we break down (NLP) into $\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega\omega \leq \sum\Omega/2$ and $\sum_{\omega\in\Omega'} \psi_\omega\omega$ whereby we satisfy the minimization argument by the indirect maximization of $\sum_{\omega\in\Omega'} \lambda_\omega$, rendering the reordering of terms in the objective unnecessary.

It remains to linearize the absolute value on the bounds $|\lambda_\omega|$ for all $\omega \in \Omega'$. Suppose we let $|\lambda_\omega|$ such that $|\lambda_\omega| \leq \psi_\omega$ for all $\omega \in \Omega'$ by the identity of $\sum_{\omega\in\Omega'} \psi_\omega\omega$. Then, with a bit of algebra we admit $\lambda_\omega - \psi_\omega \leq 0$ and $-\lambda_\omega - \psi_\omega \leq 0$, representing the linear equivalency of $|\lambda_\omega| \leq \psi_\omega$ for all $\omega \in \Omega'$. We reduce the above equations to $\lambda_\omega + \psi_\omega \leq 1$ for all $\omega \in \Omega'$.

Replacing $\lambda_\omega + \psi_\omega \leq 1$ with $\psi_\omega = 1 - \lambda_\omega$ allows for the change $\lambda_\omega, \psi_\omega \in \{0,1\}$ to $\lambda_\omega \in \{0,1\}$ for all $\omega \in \Omega'$. This eliminates the possibility of any element ω from being unassigned to any subset (that is, $\lambda_\omega = 0$ and $\psi_\omega = 0$ for some ω). The binary nature of the λ_ω variable coupled with $\psi_\omega = 1 - \lambda_\omega$ forces $\psi_\omega \in \{0, 1\}$ yielding a single changing variable and the desired result.

Remark 3. *Following* (ILP)*, if* $\lambda_\omega, \psi_\omega = 1$ *for all* $\omega \in \Omega'$ *then* $\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} (\lambda_\omega + \psi_\omega) = |\Omega|$ *and the objective function* $\left(\max_{\omega\in\Omega} \omega + \sum_{\omega\in\Omega'} \lambda_\omega \omega\right) - \sum_{\omega\in\Omega'} \psi_\omega \omega = 0$*, resulting in a perfect partition of* Ω *into* $N = 2$ *subsets with equal summation.*

3 Spreadsheet Modeling

Step 1. *Creating a table and input data for the desired* N *and cardinality* $|\Omega|$*.*

Procedure. First, we need to create the table. For simplicity, we use $N = 2$ and $|\Omega| = 50$. Initially, we label the three columns necessary to perform the analysis as follows. Type "Value" in A1, "ω" in B1, and "λ_ω" in C1. Next, type "=1" in A2 and "=2" in A3. Highlight A2 and A3 simultaneously and click the fill button at the bottom right hand corner. Drag the formula down to cell A51. In A51, the result should be the number "50", giving us the desired cardinality $|\Omega| = 50$.

Second, we create the input data. We begin by simulating a data set in a distant column (any distant column will work, but we will use column H). To satisfy Assumption 1, type "=RANDBETWEEN(0,100)" in cell H2. It might seem logical to highlight H2 and, using the fill button at the bottom right hand corner, drag the formula down to cell H51. However, to satisfy Assumption 3, we need to guarantee that the function will not produce a duplicate value. To do this, we perform the following. In cell H3, type "=LARGE(ROW($1:$100)*NOT(COUNTIF($H$2:H2,ROW($1:$100))),RANDBETWEEN(1,(100+1)-ROW(H2)))". Before pressing enter, click on the end of the formula and press "[Ctrl]+[Shift]+[Enter]". This will enter the formula as an array function. Now fill H3 down to H51.

Next, in cell A53, type "$\sum \Omega$" and in B53, type "=SUM(B2:B51)". Before we realize B53, type "=SUM(H2:H51)" in H53. If H53 is an even number, then Assumption 2 is satisfied. If not, simply press "F9" and recalculate until an even number is realized yielding $\sum \Omega \in 2\mathbb{Z}_{>0}$.

Finally, copy and paste the (evenly summed) values from H2:H51 and paste them (as values) in B2:B51. This prevents the simulated values from changing every time a key is entered in the spreadsheet. □

Step 2. *Finding* $\max_{\omega\in\Omega} \omega$ *and formulating* (NLP)*.*

Procedure. First, we need to find the maximum value in Ω. Select B2:B51 and click on the Data tab. In the Sort & Filter group, select "Sort Z to A" which sorts the data in decreasing order. This places $\max_{\omega\in\Omega} \omega$ in cell B2.

Next, type "=ABS(B2+SUMPRODUCT(B3:B51,C3:C51)-(B53/2))" to represent min $\left|\left(\max_{\omega\in\Omega}\omega+\sum_{\omega\in\Omega'}\lambda_\omega\omega\right)-\sum\Omega/2\right|$. In the Solver Parameters dialog box, enter the following.

- Set Objective: "$B$55"
- To: Min (to minimize the objective function value)
- By Changing Variable Cells: "$C$3:$C$51" (note that we leave B2 out since it is directly assigned to X_1)
- Subject to the Constraints: "$C$3:$C$51=binary" (to guarantee $\lambda_\omega \in \{0,1\}$ for all $\omega \in \Omega'$)
- Select a Solving Method: "GRG Nonlinear"

Since all decision variables are binary, it is not necessary to check the box next to "Make Unconstrained Variables Non-Negative" to allow for a solution. □

Step 3. *Formulating* (ILP).

Procedure. Begin by copying A1:C55 in a new worksheet. Type "ψ" in cell D1. We satisfy the constraint $\psi_\omega = 1-\lambda_\omega$ for all $\omega\in\Omega'$ by simply entering "1-C3" in D3 and filling down to D51. This guarantees that $\psi_\omega\in\{0,1\}$ for all $\omega\in\Omega'$.

Then, type "=(B2+SUMPRODUCT(B3:B51,C3:C51))-SUMPRODUCT(B3:B51,D3:D51)" in cell B55 to represent the objective min $\left(\max_{\omega\in\Omega}\omega+\sum_{\omega\in\Omega'}\lambda_\omega\omega\right)-\sum_{\omega\in\Omega'}\psi_\omega\omega$. Next, type "≥" in C55 and "0" in D55 to setup the initial constraint $\left(\max_{\omega\in\Omega}\omega+\sum_{\omega\in\Omega'}\lambda_\omega\omega\right)-\sum_{\omega\in\Omega'}\psi_\omega\omega\geq 0$.

To realize the inequality constraint $\left(\max_{\omega\in\Omega}\omega+\sum_{\omega\in\Omega'}\lambda_\omega\omega\right)\leq\sum\Omega/2$, type "Inequality" in cell A56. Then, type "=B2+SUMPRODUCT(B3:B51,C3:C51)" in B56, "≤" in C56, and "=B53/2" in D56.

For the equality constraint $\sum_{\omega\in\Omega'}\psi_\omega\omega=\sum\Omega/2$, begin by typing "Equality" in cell A57. Next, type "=SUMPRODUCT(B3:B51,D3:D51)" in B57, "=" in C57, and "=B53/2" in D57.

In the Solver Parameters dialog box, enter the following.

- Set Objective: "$B$55"
- To: Min (to minimize the objective function value)
- By Changing Variable Cells: "$C$3:$C$51" (again, we leave B2 out since it is directly assigned to X_1)
- Subject to the Constraints: "$C$3:$C$51=binary" (to guarantee $\lambda_\omega\in\{0,1\}$ for all $\omega\in\Omega'$ just as in (NLP); note that $\psi_\omega\in\{0,1\}$ for all $\omega\in\Omega'$ is guaranteed by the formulas in D3:D51, so we do not need to add "$D$3:$D$51=binary"); "$B$55>=$D$55"; "$B$56<=$D$56"; "$B$57=$D$57"

- Select a Solving Method: "Simplex LP"

As with (NLP), all decision variables are binary, so it is not necessary to check the box next to "Make Unconstrained Variables Non-Negative" to allow for a solution. □

Step 4. *Analyzing the solutions.*

Procedure. An HP EliteOne 800 desktop computer was used to run each formulation. The computer was equipped with an Intel Core i5-4570S (2.9 GHz) processor with 16 GB of memory and a 500 GB hard drive. Table 1 shows the resulting performance measures.

Formulation	Solver Engine	Solution Time (Seconds)	Iterations	Subproblems
(NLP)	GRG Nonlinear (see [13])	46.863	2	110
(ILP)	Simplex LP (see [16])	0.093	0	46

Table 1: Performance measures for (NLP) and (ILP)

It is clear that each formulation is attractive in its own way. (NLP) has an advantage on formulation time, while (ILP) has an advantage on run time. A complete analysis is unnecessary since 1) post-optimality (sensitivity) analysis is not available in Solver for 0-1 problems, and 2) it is beyond the scope of this research to dial in the settings and parameters of Solver. We refer the reader to [21] for a detailed description of the GRG algorithm, [2] for a brief overview on the classic Simplex method, and [6] for a treatise on Solver. □

4 Discussion

This paper develops and analyzes a simple SPP for a strict partition of a countable set of positive integers under a proposed set of conditions into $N = 2$ subsets with equal summation. We undertake the partitioning by formulating nonlinear and integer linear programs using an Excel spreadsheet. Each formulation is constructed in four-step process that makes use of the readers intuition without the use of formal proofs. This in turn makes it a useful classroom aid as well as an easy read for a general audience with minimal exposure to OR and only a basic working knowledge of Excel. What results is a simple procedure that adds to the list of OR applications in spreadsheet modeling, and uses basic Excel functions without relying on other advanced software. Further work can be directed toward modifying the proposed formulations for extending the base case of $N = 2$ to the general case for any finite $N > 2$ such that $\sum X_1 = \sum X_2 = \ldots = \sum X_N$, as well as investigating each in terms of a NPP multiset.

References

[1] Bauke, H., Mertens, S., & Engel, A. (2003). Phase Transition in Multiprocessor Scheduling. Physical Review Letters, 90(15), 158701-1-4.

[2] Bertsimas, D., & Tsitsiklis, J. N. (1997). Introduction to Linear Optimization (Vol. 6, pp. 479-530). Belmont, MA: Athena Scientific.

[3] Blizard, W. D. (1988). Multiset Theory. Notre Dame Journal of formal logic, 30(1), 36-66.

[4] Boettcher, S., & Mertens, S. (2008). Analysis of the Karmarkar-Karp Differencing Algorithm. The European Physical Journal B, 65(1), 131-140.

[5] Cochran, J. J. (2009). Pedagogy in Operations Research: Where Has the Discipline Been, Where is it Now, and Where Should it Go? ORiON, 25(2), 161-184.

[6] Fylstra, D., Lasdon, L., Watson, J., & Waren, A. (1998). Design and Use of the Microsoft Excel Solver. Interfaces, 28(5), 29-55.

[7] Gent, I. P., & Walsh, T. (1998). Analysis of Heuristics for Number Partitioning. Computational Intelligence, 14(3), 430-451.

[8] Johnson, D. S. (1973). Near-optimal Bin Packing Algorithms (Doctoral dissertation, Massachusetts Institute of Technology).

[9] Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1991). Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. Operations Research, 39(3), 378-406.

[10] Karmarkar, N., & Karp, R. M. (1982). The Differencing Method of Set Partitioning. Technical Report UCB/CSD 82/113, Computer Science Division, University of California, Berkeley.

[11] Kojic, J. (2010). Integer Linear Programming Model for Multidimensional Two-way Number Partitioning Problem. Computers & Mathematics with Applications, 60(8), 2302-2308.

[12] Korf, R. E. (1998). A Complete Anytime Algorithm for Number Partitioning. Artificial Intelligence, 106(2), 181-203.

[13] Lasdon, L. S., Waren, A. D., Jain, A., & Ratner, M. (1978). Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming. ACM Transactions on Mathematical Software (TOMS), 4(1), 34-50.

[14] Muller, T. (1998). Solving Set Partitioning Problems with Constraint programming. Proc. of PAPPACT98, 313-332.

[15] Munisamy, S. (2009). A Spreadsheet-based Approach for Operations Research Teaching. International Education Studies, 2(3), 82-88.

[16] Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. The Computer Journal, 7(4), 308-313.

[17] Pedroso, J. P., & Kubo, M. (2010). Heuristics and Exact Methods for Number Partitioning. European Journal of Operational Research, 202(1), 73-81.

[18] Perzina, R., & Ramik, J. (2014). Microsoft Excel as a tool for solving multicriteria decision problems. Procedia Computer Science, 35, 1455-1463.

[19] Rushmeier, R. A., Hoffman, K. L., & Padberg, M. (1995). Recent Advances in Exact Optimization of Airline Scheduling Problems. Dept. of Operations Research and Operations Engineering, George Mason University, Working Paper.

[20] Trautmann, N., & Gnagi, M. (2015, December). On an application of Microsoft Excel's evolutionary solver to the resource-constrained project scheduling problem RCPSP. In Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on (pp. 646-650). IEEE.

[21] Vanni, T., Karnon, J., Madan, J., White, R. G., Edmunds, W. J., Foss, A. M., & Legood, R. (2011). Calibrating Models in Economic Evaluation. Pharmacoeconomics, 29(1), 35-49.