USING A QUADRATIC PROGRAMMING APPROACH TO SOLVE SIMULTANEOUS RATIO AND BALANCE EDIT PROBLEMS

Katherine J. Thompson, James T. Fagan, Brandy L. Yarbrough, Donna L. Hambric[1]

U.S. Census Bureau

ABSTRACT

Often, items collected by the Economic Census are first subjected to ratio edit tests and associated corrections to ensure consistency within the questionnaire and within the industry and are then subsequently "balanced" to ensure additivity with other data items. With this two-step procedure, the balanced data often fail to satisfy the original ratio edit tests. In most cases, this does not pose a data quality problem. However, there are situations in which it is undesirable to "undo" the original ratio edit corrections to satisfy additivity: examples include marginal totals in two-dimensional balance complexes and derived items that combine previously ratio-edited items with unedited items. We present an approach that first uses non-linear (quadratic) programming to find data "adjustment" solutions that satisfy such simultaneous balancing and ratio editing problems and then performs controlled rounding to obtain integer solutions.

Key words: non-linear program, constrained optimization

I. Introduction

Often, items collected by the Economic Census are first subjected to ratio edit tests and associated corrections to ensure consistency within the questionnaire and within the industry and are then subsequently "balanced" to ensure additivity with other data items. With this two-step procedure, items that appear in both ratio and balance edits may fail to satisfy the original ratio edit tests. In most cases, this does not pose a data quality problem. However, there are situations in which it is undesirable to ignore the original ratio edit corrections to satisfy additivity. This paper presents two such situations and provides our editing solution, couching the editing problem as a quadratic program.

In Section 3, we discuss an application used for the Manufacturing sector of the 1997 Economic Census to edit the dollar value of assets, capital expenditures, and retirements collected from the canvassed establishments. The data collected for Assets, Capital Expenditures, and Retirements are summarized in the (4 *x* 4) table shown below; for the remainder of this paper, we refer to this data matrix as the "Assets Complex." The body of the table provides the shorthand mnemonic of each distinct data item. The table also illustrates the required horizontal and vertical balance relationships. Specifically:

(1) In all rows, the TOTAL item should always = the BUILDING item + the MACHINERY item; and
(2) In the TOTAL, BUILDING, and MACHINERY columns row *d* should always = $a + b - c$.

Table 1: Assets Complex Data for the Manufacturing Sector of the Economic Census

Row	Description	TOTAL	BUILDING	MACHINERY
a	Assets at the beginning of the year	TAB	BAB	MAB
b	Capital Expenditures	TCE	CBE	CME
c	Retirements during the year	TRT	BRT	MRT
d	Assets at the end of the year	TAE	BAE	MAE

Besides these two-dimensional balance constraints, there are (industry-specific) ratio constraints on three marginal row totals: Ending Assets (TAE), Capital Expenditures (TCE) and Retirements (TRT).

The excellent results from the Assets Complex application motivated our second application. In Section 4, we describe our application of this approach to edit Gross Margin and Gross Profit for the Wholesale Trade sector of the 2002 Economic Census. This sector collects data on Payroll, Employment, Receipts, Operating Expenses, Purchases, and Inventories items from all establishments, as well as data on Commissions and Gross Selling Value when available. Gross Margin and Gross Profit values are derived from these items as

Gross Margin = Sales – Gross Selling Value – Beginning Inventories – Purchases + Ending Inventories
Gross Profit = Gross Margin – Operating Expenses + Commissions

Additionally, there are (industry-specific) ratio constraints on all of the component items.

[1] The authors thank Debra Coaxum, Scot Dahl, David Dickerson, Rita Petroni, and Richard Sigman for their useful comments on earlier versions of this manuscript. This report is released to inform interested parties of research and to encourage discussion.

This paper presents a non-linear (quadratic) programming approach that finds data "adjustment" solutions that satisfy such simultaneous balancing and ratio editing problems and then performs controlled rounding to obtain integer solutions. In Section II, we describe the general set-up for our quadratic programs, introducing some terminology and motivating the approach taken in the applications described in Sections III and IV. Section V provides our final observations and some recommendations for future applications.

II. General Set-Up

Consider a set of five data items: A, B, C, D, and E. The edited values of the first four items **must** satisfy two ratio relationships: $L_{AB} \leq A/B \leq U_{AB}$ and $L_{CD} \leq C/D \leq U_{CD}$. In addition, the sum of the edited values of A, C, and D must equal E (balance constraint). Finally, the edited values of all five items must be non-negative. Assume that all input data items have been subjected to ratio edits, and that some item values may have been previously imputed. Our objective is to minimize change from input to output data while satisfying the ratio and balance constraints.

Deming (1943) proposes a least-squares estimation solution to the data adjustment problem. For this, Deming assumes that observations contain random measurement error; that is $Y_i=E(Y_i)+\ _i$, where Y_i is the original unadjusted value for item i, $\ _i$ is a random error term for data item i with mean zero and measurement variance $\ _i^2$ ($\sum \ _i^2 = \ ^2$, the unit measurement error), and $E(Y_i)$ is the unobserved true measurement for item i. The adjusted values for the Y_i, denoted Y_i^*, are the least-squares estimators for the $E(Y_i)$ that minimize the following weighted sum of the residuals

$$Q=\sum_i\left(\frac{\sigma^2}{\sigma_i^2}\right)\left(Y_i-Y_i^*\right)^2=\sum_i\alpha_i\left(Y_i-Y_i^*\right)^2 \qquad (2.1)$$

without violating any of the constraints. Often, the least squares estimation minimization problem solution is obtained via Lagrange multipliers. The values that minimize the (constrained) Q are unbiased and consistent estimators. Note that in a "pure" least squares estimation setting, $\alpha_i \approx \ / \ _i$: items with larger variance will have weights close to 1, and items with small variance will have very large weights. Since this is a minimization problem, items with large variance will be adjusted before the less variable items.

The **same** adjustment problem can be couched as a quadratic program in Operations Research (OR) terminology. Our example defines the following quadratic program:

$$\min\left[\alpha_A(A-A^*)^2+\alpha_B(B-B^*)^2+\alpha_C(C-C^*)^2+\alpha_D(D-D^*)^2+\alpha_E(E-E^*)^2\right] \qquad (2.2)$$

[2]

subject to

$A^* + C^* + D^* - E^* = 0$	Balance constraint
$(B^*)(L_{AB}) - A^* \leq 0$	1st ratio edit constraint
$A^* - (B^*)(U_{AB}) \leq 0$	
$(D^*)(L_{CD}) - C^* \leq 0$	2nd ratio edit constraint
$C^* - (D^*)(U_{CD}) \leq 0$	
$A^*, B^*, C^*, D^*, E^* \geq 0$	Non-negativity constraints

The objective functions that we use in our applications are very similar to the least squares estimation objective function (2.1), varying only in the α_i. Our α_i are **subjectively defined** item-weights, designed to control the edit outcome. Note that in practice, each original non-linear (ratio) constraint is changed to two linear constraints, although we include the original ratio constraints in the quadratic programs in Sections 3 and 4 for clarity.

In (2.2), the minimization program solution can modify **any** of the five input data items. This may not be a desirable result. If, for example, the edit-implementer wished to preserve the ratio relationship between items A and B but did **not** want to modify item B's input value, then the $\alpha_B(B - B^*)^2$ term would be dropped from the objective function ("fixing" item B), and the $(B)(L_{AB})$ term in the first constraint and $(B)(U_{AB})$ term in the second constraint would be defined as constants. Note that dropping a variable from the objective function while retaining it as a constraint variable can increase the probability of an infeasible solution. In the applications discussed in Sections II and III, infeasibility was more often caused by intractable data than unsatisfiable constraints.

We used the simplex algorithm (Wolfe 1959) to obtain the non-integer solution to the quadratic programs. See the appendix for more details. After obtaining the non-integer solutions, we used the controlled rounding technique described in Cox and Ernst (1982) to obtain the final integer solutions that satisfied the additivity constraints.

---

[2] $\min\left[\sum_{i=A}^{E}\alpha_i \bullet i^{*2} - 2\sum_{i=A}^{E}\alpha_i(i \bullet i^*)^2 + \text{constant}\right]$, where $\text{constant} = \sum_{i=A}^{E}\alpha_i \bullet i^2$.

III. Example from the 1997 Census of Manufactures: Assets Complex

Prior to the 1997 Economic Census, the Assets Complex data were edited using a legacy program first developed in the early 1970's. While this program had yielded acceptable results during its initial years of usage, it was poorly documented and little understood by those who inherited it. Unfortunately, the end-result of the 1992 census application was unsatisfactory. The legacy program failed to provide suitable results and, facing resource and time-constraints, the only recourse was to correct the data via ad hoc methods such as custom-coded batch updates or interactively on a record-by-record basis, not a pleasant solution for a census of 350,000 establishments. Clearly, a more efficient means for editing these data was required for the 1997 census.

The most difficult aspect of developing a new edit for the Assets Complex turned out to be devising a way to maintain the horizontal and vertical balance among the various data items. After some preliminary discussion on how best to achieve this goal, it quickly became clear that "If-Then-Else" logic (used in the legacy system) would simply be impractical due to the overwhelming number of scenarios that might result in an imbalanced matrix. A proof-of-concept test conducted jointly between research and methodology staff and the edit developers illustrated that quadratic programming techniques could be used to successfully balance the Assets matrix. Ultimately, the new Asset Complex edit was a hybrid of traditional ratio editing and quadratic programming. These two editing phases are described below.

Phase 1: Ratio Editing

The data items Total Ending Assets (TAE), Total Capital Expenditures (TCE), and Total Retirements (TRT) were first edited independently from the other items in the matrix. Specifically, they were subjected to the following sequence of tests (in order):

$lb_{1i} \le TAE/x \le ub_{1i}$ (3.1.r)
(x represents the value of payroll or shipments depending on the industry)
$lb_{2i} \le TCE/TAE \le ub_{2i}$ (3.2.r)
$lb_{3i} \le TRT/TAE \le ub_{3i}$ (3.3.r)

where lb_i and ub_i are industry-specific lower and upper limits. At the point that these ratio tests were performed, payroll and shipments values were previously edited; their values were fixed. If TAE failed the first ratio test (3.1.r), its reported value was replaced by a rounded[3] or a value imputed from the previously fixed value of payroll or shipments. Similarly, the reported values of TCE and TRT could change if they failed their respective ratio test against the fixed edited value of TAE.

TAE, TRT, and TCE were selected as candidates for ratio editing due to their relatively high reliability (as compared to the other items in the assets matrix) and the availability of a reasonable amount of quality data with which to construct industry-specific parameters. Total Beginning Assets (TAB) could also have been subjected to ratio editing but it was not necessary to do so since, as shown below, the balance constraint TAE=TAB + TCE - TRT in the quadratic program effectively ensures its' consistency.

Phase 2: Quadratic Programming

After ratio editing TAE, TRT, and TCE, we implemented the following quadratic program:

$$\min [\alpha(TAB-TAB^*)^2 + \alpha(TCE-TCE^*)^2 + \alpha(TRT-TRT^*)^2 + \alpha(TAE-TAE^*)^2 + \alpha(MAB+ MAB^*)^2 + \alpha(CME-CME^*)^2 + \alpha(MRT-MRT^*)^2 + \alpha(MAE-MAE^*)^2 + \alpha(BAB-BAB^*)^2 + \alpha(CBE-CBE^*)^2 + \alpha(BRT-BRT^*)^2 + \alpha(BAE-BAE^*)^2] \quad (3.1)$$

Where $\alpha = \begin{cases} 10{,}000 \text{ if reported} \\ 1 \text{if imputed} \end{cases}$

s.t.

Ratio constraints (from Phase 1)

$lb_{1i} \le TAE^*/x \le ub_{1i}$ (from Phase 1) (3.1.r)
$lb_{2i} \le TCE^*/TAE^* \le ub_{2i}$ (from Phase 1) (3.2.r)
$lb_{3i} \le TRT^*/TAE^* \le ub_{3i}$ (from Phase 1) (3.3.r)
$lb_{4i} \le CME^*/TCE^* \le ub_{4i}$ (3.4.r)
$lb_{5i} \le MRT^*/TRT^* \le ub_{5i}$ (3.5.r)
$lb_{6i} \le MAE^*/TAE^* \le ub_{6i}$ (3.6.r)

Balance constraints

$TAE^*=TAB^* + TCE^* – TRT^*$ (3.1.b)
(column 1 additivity)
$BAE^*= BAB^* + CBE^* – BRT^*$ (3.2.b)
(column 2 additivity)
$MAE^*= MAB^* + CME^* – MRT^*$ (3.3.b)
(column 3 additivity)
$TAB^*= BAB^* + MAB^*$ (3.4.b)
(row 1 additivity)
$TCE^*= CBE^* + CME^*$ (3.5.b)
(row 2 additivity)
$TRT^*= BRT^* + MRT^*$ (3.6.b)

[3] reported value/1000, considered "reported data"

(row 3 additivity)
$TAE^* = BAE^* + MAE^*$ (3.7.b)
(row 4 additivity)
plus all non-negativity constraints on each variable

The widely disparate values of α ensure that imputed data items will be changed before reported ones. The additional three ratio constraints – combined with their associated additivity constraints -- guarantee that the adjusted detail items cannot be larger than their corresponding row totals [In tests (3.4.r)- (3.6.r), the lower bounds are greater than or equal to zero and the upper bounds are less than or equal to one]. Exactly one of the balance constraints listed above is redundant. To improve performance, we remove one constraint ahead of time.

Overall, the hybrid approach used for editing the Assets Complex during the 1997 Census of Manufactures was highly successful and a marked improvement over the experiences of the previous census. Most notably, the quadratic programming techniques that were utilized virtually eliminated one of the largest obstacles in developing a practical edit– the horizontal and vertical balance problem. Once implemented, there were a few production problems. First, the methodology did not always work very well for "non-normal" records– e.g., part-year operators, idle plants, out-of-business establishments. However, this was largely a function of their not conforming to the ratio parameters. Second, there were a handful of "unsolvable" cases that had to be dealt with manually. Many of these were often due to extremities or oddities in the data that did not get handled properly in other parts of the edit system (i.e., a bad payroll or shipments value was input to the assets edit). A few others were a result of the software requiring greater numerical precision than the compiler allowed in order to obtain a solution.

IV. Example from the Wholesale Trade Sector of the 2002 Economic Census: Gross Margin/Gross Profit

The Wholesale Trade sector of the Economic Census comprises approximately 400,000 establishments. Prior to the 2002 Economic Census, the derived values of Gross Margin and Gross Profit were automatically checked with ratio tests. Analysts reviewed edit-failing records and **manually** adjusted component data so that the final component data, Gross Margin, and Gross Profit values were all within their industry-specific bounds. In doing this, analysts attempted to adjust values of Operating Expenses, Purchases, then lastly Sales; the other data item values were changed when a feasible solution was otherwise impossible.

For 2002, we wanted to automate this adjustment procedure. Formulating the quadratic program was straightforward: in addition to the two balance constraints defining the two derived items, there were several existing industry-specific ratio edit constraints for most of the component data items. The challenge lay in developing item-specific weights that controlled the sequencing of data-item value adjustment.

We use the following mneumonics in the quadratic program below:

Mneumonic	Data Item
RCP	Receipts/sales
APR	Annual Payroll
OPX	Operating Expenses
EINV	Ending Inventories
BINV	Beginning Inventories
PURCH	Purchases
GSV	Gross Selling Value
COMM	Commissions
GM	Gross Margin
GP	Gross Profit

We initially tried to edit both derived items simultaneously, using an objective function that included all component data with the following constraints:

Ratio Constraints (Industry-Specific)

$lb_{1i} \le RCP/APR \le ub_{1i}$ (4.1.r)
$lb_{2i} \le OPX/APR \le ub_{2i}$ (4.2.r)
$lb_{3i} \le RCP/OPX \le ub_{3i}$ (4.3.r)
$lb_{4i} \le EINV/BINV \le ub_{4i}$ (4.4.r)
$lb_{5i} \le RCP/BINV \le ub_{5i}$ (4.5.r)
$lb_{6i} \le RCP/PURCH \le ub_{6i}$ (4.6.r)
$lb_{7i} \le GM/(RCP - GSV) \le ub_{7i}$ (4.7.r)
$lb_{8i} \le GP/RCP \le ub_{8i}$ (4.8.r)

Ratio Constraints (Item-Specific)

$0 < GP/GM < 1$ (4.9.r)
$0 \le COMM/GSV \le 0.15$ (4.10.r)

Additivity (Balance) Constraints

$GM = RCP – GSV – BINV – PURCH + EINV$ (4.1.b)
$GP = GM – OPX + COMM$ (4.2.b)

where the item-weights (α_i) in the objective function had two different values for each data item:

Where	If Reported...	If Imputed...
α_{RCP} =	17	3
α_{GSV} =	49	49
α_{BINV} =	25	10
α_{PURCH} =	15	1
α_{EINV} =	26	11
α_{OPX} =	2	1
α_{COMM} =	50	50

Subject-matter experts were not pleased with the solutions from this formulation. Often, several items were changed by a small amount rather than changing one or two values by a large amount, affecting imputation rates for too many items. More important, this approach was not automating the desired correction procedure. We needed more control over **which** data items were adjusted, and the desired sequencing of item-adjustment depended on the derived item. If Gross Margin failed its ratio test, then we wanted to first adjust values of Purchases, then Sales. If after correcting Gross Margin, Gross Profit still failed its ratio test, then we wanted to adjust values of Operating Expenses, then Purchases, and lastly Sales.

The only way to guarantee this type of control was to split to original problem into two separate quadratic programs (one for Gross Margin and one for Gross Profit), develop different item-weights for each program, and then run the two programs sequentially. At this point, we decided to drop the distinction between reported and imputed values in the item-weights, using one α_i per **item** per **quadratic program**. We attempted to control the relative adjustment to the variables through the increased item-weights (e.g., assigning weights that were larger by a factor of 10^9 to the variables whose value we did not want to change).

This second formulation gave much improved – but not completely satisfactory – results. The objective functions for both programs still include **all** component data items, and even though the item-weights on Sales, Purchases, and Operating Expenses were **much** larger than the other items', the quadratic program solutions still changed other component data. At this point, we realized that the only way to prevent this from happening was to remove the other four data items – Gross Selling Value, Beginning Inventories, Ending Inventories, and Commissions – from both objective functions and treat them as constants; we also removed ratio constraints (4.4.r) and (4.10.r).[4] Equations 4.1 and 4.2 below provide the final (production) quadratic programs.

Quadratic Program 1: Gross Margin

$$\min[\alpha_{RCP}(RCP - RCP^*)^2 + \alpha_{PURCH}(PURCH - PURCH^*)^2] \quad (4.1)$$

subject to ratio constraints (4.1.r), (4.2.r), (4.3.r), (4.5.r), (4.6.r), and (4.7.r) and balance constraint (4.1.b)

where $_{RCP} = 1,000,000,000$
$_{PURCH} = 1$

Quadratic Program 2: Gross Profit

$$\min[\alpha_{RCP}(RCP - RCP^*)^2 + \alpha_{PURCH}(PURCH - PURCH^*)^2 + \alpha_{OPX}(OPX - OPX^*)^2 + \alpha_{GM}(GM - GM^*)^2] \quad (4.2)$$

subject to ratio constraints (4.1.r)- (4.3.r) and (4.5.r) - (4.9.r) and balance constraints (4.1.b) and (4.2.b)

where $_{RCP} = 1,000,000,000$
$_{PURCH} = 1,000,000$
$_{OPX} = 1$
$_{GM} = 1,000,000$

Implementing these programs in the 2002 Economic Census greatly reduced the proportion of cases reviewed by analysts (referral rates) for these two items from 1997: for Gross Margin, the 2002 Economic Census referral rate dropped to 1.5% from 4.2% (the 1997 referral rate), and for Gross Profit, the 2002 Economic Census referral rate dropped to 1.7% from 5.6% (the 1997 referral rate). It is possible that other factors such as the introduction of electronic reporting in 2002 could have reduced the Gross Margin/Gross Profit edit failure rates. However, prior to 2002, all records that failed Gross Margin or Gross Profit were referred to an analyst for manual correction. With the quadratic program approach in 2002, only the truly intractable cases were referred to an analyst.

V. Discussion

In both of the applications described above, the quadratic programming approach yielded highly

[4] There were some initial implementation problems caused by the ratio constraint on Sales/Beginning Inventories (4.5.r): if the respondent reported no inventories data, then the quadratic program solution forced the Sales value to equal zero. Reported zero inventories with non-zero sales can be legitimate (e.g., merchant agents). To avoid this erroneous correction, the Sales/Beginning inventories constraint was dropped when Beginning Inventories were zero.

satisfactory results in a production environment. Most of the difficulty lay in the set-up, e.g., developing parameters on the ratio tests, finding appropriate item-weights (α_i), collecting the entire set of constraints. For example, developing ratio edit parameters can be time-consuming when sufficient historical data are available. When historical data are not available for parameter development, then the edit-implementors must evaluate the effect(s) of the ratio edit constraints on the solutions very carefully.

We designed our item-weights to control the adjustment solution. In doing this, we sacrifice some of the optimal statistical properties that we would gain from a pure least squares estimation application. For the Manufacturing application described in Section 3, this is not a problem: the objective was to preserve reported data. In the Wholesale application, it could be. Census data are available that could be used to develop item-weights that reflect the proportion of measurement error represented by each data item *i* (i.e., we could develop $\alpha_i \approx \ / \ _i$). It would be worthwhile to pursue this for future applications so that the quadratic program solutions would truly be least squares estimators (retaining the associated optimal statistical properties).

Quadratic programming is a very common type of non-linear problem, and there are a variety of software packages available for finding (non-integer) solutions. For our census applications, we needed to incorporate the quadratic programs into our production edit, so we designed customized FORTRAN programs. In other settings, this customization may not be necessary. The quadratic programming approach that we used for these two examples could easily be applied to other Economic Census data sets. The challenge is to make it more accessible, via generalized programs, parameter development tools, and user training. The process that we used to develop the quadratic program applications described in Sections 3 and 4 were fairly ad hoc. If we generalize this approach, then we need to combine it with reliable diagnostic tools for evaluating the individual quadratic program components.

REFERENCES

Cox, L. and Ernst, L. (1982). Controlled Rounding. *INFOR*, **20**, pp. 423-432.

Deming (1943). Statistical Adjustment of Data. New York: Wiley.

Flannery, B.R., Taukolsky, S.A., and Vetterling, W.T. (1992). Numerical Recipes – the Art of Scientific Programming (FORTRAN Version). Boston, MA: Cambridge University Press.

Wolfe, P. (1959). The Simplex Method for Quadratic Programming. *Econometrica*, **27**, pp. 382 – 398.

Appendix

Consider the quadratic programming problem in *n* variables, with m_1 equality (balance/additivity) constraints and m_2 inequality (ratio) constraints:

$$\min \tfrac{1}{2}\,[\underline{x}^T Q \underline{x} + \underline{p}^T \underline{x}]$$
$$\text{s.t.}\quad A\underline{x} = \underline{b}$$
$$C\underline{x} \le \underline{d}$$
$$\underline{x} \ge 0$$

where $\underline{x}$ is an $n \times 1$ vector of output values, Q is an $n \times n$ matrix = $2^*\text{diag}(\alpha_i)$, and $\underline{p}$ is an $n \times 1$ vector of items equal to $-2^*\alpha_i{}^*x_{i(\text{original})}$.

If Q is real symmetric and positive semi-definite, this problem is convex. In this case, if the problem is feasible, there exists a global solution at some $\underline{x}^*$. Furthermore if Q is positive definite that solution $\underline{x}^*$ is unique.

The necessary and sufficient conditions for $\underline{x}$ to be a global minimum of the above quadratic programming problem are that there exist $\underline{x} \in R^n$, $\underline{y} \in R^{m2}$, $\underline{v} \in R^n$, $\underline{w} \in R^{m2}$, and $\underline{u} \in R^{m1}$ such that

$$A\underline{x} = \underline{b} \qquad (1)$$
$$C\underline{x} + \underline{y} = \underline{d} \qquad (2)$$
$$Q\underline{x} - \underline{v} + C^T\underline{w} + A^T\underline{u} = -\underline{p} \qquad (3)$$
$$\underline{v}^T\underline{x} = 0 \qquad (4)$$
$$\underline{w}^T\underline{y} = 0 \qquad (5)$$
$$\underline{x}, \underline{y}, \underline{w}, \underline{v} \ge 0 \qquad (6)$$

Note that $\underline{u}$, $\underline{v}$, and $\underline{w}$ (the dual variables) are Lagrange multipliers. Solving this system amounts to finding a basic feasible (phase 1) solution to the constraint set { (1), (2), (3), (6)} with the following basis restrictions:

- If x_j is in the basis, v_j cannot enter the basis; conversely, if v_j is in the basis, x_j cannot enter the basis
- If y_j is in the basis, w_j cannot enter the basis; conversely, if w_j is in the basis, y_j cannot enter the basis

This methodology is described in Wolfe (1959). To solve the quadratic programming problem, we modified the simplex linear programming code found in Flannery et al (1992) to incorporate the above complimentary slackness conditions. The result is a fractional solution that satisfies the ratio and balance constraints.