A METHODOLOGY TO MATCH DISTRIBUTIONS OF BOTH HOUSEHOLD AND PERSON ATTRIBUTES IN THE GENERATION OF SYNTHETIC POPULATIONS

Xin Ye
Department of Civil and Environmental Engineering
Arizona State University, Room ECG252
Tempe, AZ 85287-5306
Tel: (480) 965-2262; Fax: (480) 965-0557
Email: xin.ye@asu.edu

Karthik Konduri
Department of Civil and Environmental Engineering
Arizona State University, Room ECG252
Tempe, AZ 85287-5306
Tel: (480) 965-3589; Fax: (480) 965-0557
Email: karthik.konduri@asu.edu

Ram M. Pendyala
Department of Civil and Environmental Engineering
Arizona State University, Room ECG252
Tempe, AZ 85287-5306
Tel: (480) 727-9164; Fax: (480) 965-0557
Email: ram.pendyala@asu.edu

Bhargava Sana
Department of Civil and Environmental Engineering
Arizona State University, Room ECG252
Tempe, AZ 85287-5306
Tel: (480) 965-3589; Fax: (480) 965-0557
Email: bsana@asu.edu

Paul Waddell
Public Affairs and Urban Design and Planning
University of Washington
Seattle, WA 98195-3055
Tel: (206) 221-4161; Fax: (206) 543-1096
Email: pwaddell@u.washington.edu

Submitted for Presentation Only to the
88th Annual Meeting of the Transportation Research Board
January 11-15, 2009

Word Count: 9600 (text) + 9 (tables/figures) × 250 = 11850 equivalent words

ABSTRACT

The advent of microsimulation approaches in travel demand modeling, wherein activity-travel patterns of individual travelers are simulated in time and space, has motivated the development of synthetic population generators. These generators typically use census-based marginal distributions on household attributes to generate joint distributions on variables of interest using standard iterative proportional fitting (IPF) procedures. Households are then randomly drawn from an available sample in accordance with the joint distribution such that household-level attributes are matched perfectly. However, these traditional procedures do not control for person-level attributes and joint distributions of personal characteristics. In this paper, a heuristic approach, called the Iterative Proportional Updating (IPU) algorithm, is presented to generate synthetic populations whereby both household-level and person-level characteristics of interest can be matched in a computationally efficient manner. The algorithm involves iteratively adjusting and reallocating weights among households of a certain type (cell in the joint distribution) until both household and person-level attributes are matched. The algorithm is illustrated with a small example, and then demonstrated in the context of a real-world application using small geographies (blockgroups) in the Maricopa County of Arizona in the United States. The algorithm is found to perform very well, both from the standpoint of matching household and person-level distributions and computation time. It appears that the proposed algorithm holds promise to serve as a practical population synthesis procedure in the context of activity-based microsimulation modeling.

Keywords: synthetic population, population generator, heuristic algorithm, iterative proportional fitting, iterative proportional updating, travel demand forecasting, activity based modeling, microsimulation

INTRODUCTION

The emergence of activity-based microsimulation model systems has ushered in a new era in travel demand forecasting. Activity-based approaches to travel demand explicitly recognize that travel demand is a derived demand, where individuals travel to undertake activities that are distributed in time and space. The behavioral unit considered in activity-based approaches is the individual traveler, thus leading to microsimulation model systems that are capable of simulating activity-travel patterns of individual persons over the course of a day. Recent examples of full-fledged activity-based microsimulation model systems include AMOS (Kitamura and Fujii, 1998), FAMOS (Pendyala et al., 2005), CEMDAP (Bhat et al., 2004), ALBATROSS (Arentze and Timmermans, 2004), and TASHA (Miller and Roorda, 2003). In addition to these model systems, various tour-based model systems that also simulate activity-travel patterns at the level of the individual traveler are being implemented in several urban areas around the United States, including (but not limited to) San Francisco, New York, Columbus, Denver, Atlanta, Tampa Bay, and Puget Sound (Vovsha et al., 2005). Even if the underlying behavioral paradigms differ across model systems, the common element is that they all simulate activity-travel patterns at the level of the individual traveler while attempting to explicitly recognize the role of spatio-temporal constraints, time use allocation, and agent interactions (e.g., interactions among household members, interactions among activities and trips). For these reasons, activity-based and tour-based model systems are considered to provide robust behavioral frameworks for analyzing travel demand under a wide variety of modal and policy contexts.

As activity-based microsimulation model systems operate at the level of the individual traveler, one needs household and person attribute information for the entire population in a region to calibrate, validate, and apply (in a forecasting mode) such model systems. However, such information is virtually never available at the disaggregate level for an entire region. In the base year, one may have disaggregate household and person information for a random sample of households. Such information may be available from a census (for example, in the United States, such data is available through the Public Use Microdata Sample or PUMS) or from a traditional activity-travel survey that may have been conducted by the planning agency in a region. The challenge in activity-based modeling, then, is to generate a complete synthetic population with comprehensive data on attributes of interest. The activity-based model system can then be applied to the synthetic population to forecast travel demand at the level of the individual traveler.

Synthetic populations can be formed from the random samples by choosing or selecting households and persons from the random samples such that the joint distribution of the critical attributes of interest in the synthetic population match known aggregate distributions of household and person attributes available through a Census. For example, in the United States, marginal distributions of population characteristics are readily available from Census Summary Files (SF) for any region. For some combinations of critical variables, the Census Summary Files may also directly provide joint distributions against which synthetic population joint distributions can be matched. However, more often than not, such joint distributions of critical attributes of interest are not directly available and the analyst must generate these joint distributions from the known marginal distributions of interest.

The joint distributions among a set of control variables can be estimated using the well-known Iterative Proportional Fitting (IPF) procedure. The iterative procedure was first presented by Deming and Stephan (1941) in the context of adjusting sample frequency tables to match known marginal distributions and further refined by Fienberg (1970) and Ireland and Kullback

(1968). Wong (1992) showed that the IPF procedure yields the maximum entropy estimates of the joint distribution under the constraints of the given marginal distributions. Beckman et al. (1996) used the iterative procedure to estimate joint distributions of household attributes. Sample frequency tables used in the study were generated from Public Use Microdata Sample (PUMS) data using critical household attributes, for which marginal distributions were available from the Census Summary Files. Synthetic households were then generated by randomly drawing households from the PUMS according to the estimated joint distributions. The synthetic population then consisted of all persons from the selected households. While this procedure ensured that household attributes in the synthetic population closed matched the iteratively determined joint distributions, it did not have any mechanism to ensure that such consistency existed for the person attributes of interest (e.g., age, race, gender, ethnicity, employment status). As a result, distributions of person attributes fail to match the known distributions of person characteristics from the Census Summary Files. In general, this is the approach that has been adopted in tour-based and activity-based model development efforts around the world.

This is not to say that this problem has gone unnoticed. Guo and Bhat (2007) propose an algorithm that can be used to generate synthetic populations where the household-level joint distributions are close to those estimated using the conventional IPF procedure, while simultaneously improving the fit of person-level distributions. Arentze et al. (2007) propose a method using relation matrices to convert distributions of individuals to distributions of households such that marginal distributions can be controlled at the person-level as well. This paper intends to further build on their work by presenting a practical heuristic approach for generating synthetic populations while simultaneously controlling for both household and person attributes of interest.

There are two primary factors motivating this paper. First, it is desirable to have an approach where both household-level and person-level distributions can be matched as closely as possible in a synthetic population generator. Second, it is desirable to have an approach that is practical from an implementation and computational standpoint. Synthetic population generators need to generate populations for small geographies, at the level of the traffic analysis zone (TAZ), census tract, blockgroup, or block. As there are several thousand small geographies in any region, the algorithm must be computationally efficient to be practical and have reasonable run times. Although computational burden may not be a consideration from a theoretical standpoint in this era of parallel computing and ever-faster machines, it remains a practical consideration for many agencies that want the ease and flexibility of running a population synthesizer on a single desktop computer.

The next section of the paper describes the algorithm in detail. The third section offers a discussion of key issues encountered in the generation of synthetic populations. The fourth section presents real-world examples where the proposed approach has been applied to two small geographies (blockgroups) in the Maricopa County region of Arizona. Results of the case study examples are furnished in this section. The fifth and final section presents concluding thoughts.

THE PROPOSED ALGORITHM

This section presents a detailed description of the proposed algorithm. First, a simple example is presented to illustrate how the algorithm works. Then, a geometric interpretation of the algorithm is presented. Finally, a step-by-step procedure for implementing the general iterative proportional updating algorithm is offered in this section.

An Example to Illustrate the Algorithm

The traditional IPF procedure that lies at the heart of most synthetic population generators involves the estimation of household and person level joint distributions that match the given household and person level marginal frequency distributions. This procedure will naturally result in two different sets of weights, one set for matching household distributions and one set for matching person-level distributions. Except under extremely unrealistic conditions, household weights will never match person weights. As a result, a synthetic population that is generated based on the application of household weights will yield joint distributions of person attributes that do not match the given person-level marginal distributions. This is because the traditional procedure involves simply selecting all persons in the chosen households according to the household weights. In other words, the person weights are forced to be equal to the corresponding household weights, when in fact they are different. The desire to generate a synthetic population whereby both household and person-level attribute distributions are matched against known marginal distributions is one of the primary motivating factors for this paper.

The inconsistency in person-level distributions can be reduced if the household weights are adjusted based on the person weights obtained from the IPF procedure. The process by which this can be accomplished is best illustrated with the help of a small numerical example. The algorithm begins by creating a frequency matrix *D* (Table 1). A row in the matrix corresponds to a single household record and provides data describing the composition of the household. For example, the first household is of type 1 and has one individual each of person types 1, 2, and 3. There are two household types and three person types considered in this example. The household types may be defined by such variables as household size or income. The person types may be described by such variables as age, race, and gender. In this example, there are eight households with 23 individuals. All initial household weights are set to unity as shown in the Table. The row titled "weighted sum" represents the sum of each column weighted by the "weights" column. The "constraints" row provides the frequency distribution of the household and person types that must be matched. The rows titled δ_a and δ_b provide the absolute value of the relative difference between the weighted sum and the given constraints so that the "goodness-of-fit" of the algorithm can be assessed at each stage of the algorithm and convergence criteria can be set. The data structure shown in the table can be used to formulate a mathematical optimization problem in which one desires to calibrate weights such that the weighted sum equals or nearly equals the given frequency distribution. The mathematical optimization problem takes the following form depending on the form of the objective function that is adopted:

$$\text{Minimize } \sum_j \left[\left(\sum_i d_{i,j} w_i - c_j \right) / c_j \right]^2 \text{ or } \sum_j \left[\left(\sum_i d_{i,j} w_i - c_j \right)^2 / c_j \right] \text{ or } \sum_j \left[\left| \left(\sum_i d_{i,j} w_i - c_j \right) \right| / c_j \right] \quad (1)$$

subject to $w_i \geq 0$,

where i denotes a household ($i = 1, 2, ..., 8$)

j denotes the constraint or population characteristic of interest ($j = 1, 2, ..., 5$)

$d_{i,j}$ represents the frequency of the population characteristic (household/person type) j in household i

w_i is the weight attributed to the i^{th} household

c_j is the value of the population characteristic j.

The objective functions to be minimized represent different measures of inconsistency between the weighted frequency of the household/person type and the given frequency distribution constraints that need to be met. It is straightforward to solve this optimization problem for a small number of households. The constrained optimization problem can be converted into an unconstrained optimization problem by parameterizing w_i as $\exp(\lambda_i)$ or λ_i^2 and using well-established gradient search methods to minimize the objective function. However, in real-world applications, there are usually thousands of households in the sample and using gradient search methods would involve solving for thousands of λ_i to satisfy the first order conditions for minimizing the objective function. This makes the solution of this optimization problem using traditional optimization methods computationally intractable.

In this paper, a heuristic iterative procedure that is termed the Iterative Proportional Updating (IPU) algorithm is proposed as an approach to solve the problem described above. As optimal solutions cannot be strictly guaranteed in the proposed algorithm, it is considered to be a heuristic algorithm wherein the analyst must monitor performance or goodness-of-fit to determine the point at which the algorithm should be terminated. The idea behind the proposed algorithm is very intuitive and the algorithm itself is highly practical in terms of computational performance and goodness-of-fit. In this section, the algorithm will be described using the simple example illustrated in Table 1. The complete generalized procedure is presented in the next section of this paper.

The IPU algorithm starts by assuming equal weights for all households in the sample. The algorithm then proceeds by adjusting weights for each household/person constraint in an iterative fashion until the constraints are matched as closely as possible for both household and person attributes. For example, the weights for the first household level constraint are adjusted by dividing the number of households in that category (i.e., the constraint value) by the weighted sum of the first household type column. That ratio is 35/3 = 11.67. The weights for all households of household type 1 are multiplied by this ratio to satisfy the constraint. The weights for all households of household type 1 become equal to 11.67, and the weighted sum for household type 1 will be equal to the corresponding constraint, as shown in the row titled "weighted sum 1". Similarly, the weights for households of household type 2 are adjusted by an amount equal to 65/5 = 13.00. The updated weights are shown in the "weights 2" column of Table 1, and one notes that the household level constraints are perfectly satisfied at this point (see the row titled "weighted sum 2").

The weights are next updated to satisfy person constraints. For the first person-level constraint, the adjustment is calculated as the ratio of the constraint for person type 1 to weight sum of the person type 1 column after the completion of household-level adjustments. This ratio is equal to 91/111.67 = 0.81. This value is used to update the weights of all households that have individuals of person type 1. As the fifth household (household ID 5) does not have any persons of type 1, the weight for this particular household remains unchanged. The resulting adjusted weights are shown in Table 1 in the column titled "weights 3". The constraint corresponding to person type 1 is now perfectly matched. The process is repeated for the other two person type constraints and the corresponding updated weights are shown in the columns titled "weights 4" and "weights 5" in Table 1. The corresponding weighted sums are shown in the various rows of Table 1 titled "weighted sum".

The completion of all adjustments to weights for one full set of constraints is defined as one iteration. It can be seen from Table 1 that the difference between the weighted sums and the corresponding constraints for the household/person types of interest has been considerably reduced after one complete iteration. The absolute value of the relative difference between the weighted sum and the corresponding constraint may be used as a goodness-of-fit measure and is defined as:

$$\delta_j = \frac{\left| d_{i,j} w_i - c_j \right|}{c_j} \tag{2}$$

where all symbols are as denoted earlier in the context of equation (1). The average value of this measure across all constraints is denoted by δ and serves as an overall goodness-of-fit measure after each complete iteration. Prior to any adjustments being made to the weights, the value of δ, denoted δ_b, is found to be 0.9127. After the completion of one full iteration, the value of δ, denoted δ_a, is found to be 0.0954, representing a substantial improvement in the matching of the weighted sample against known population numbers. The gain in fit between two consecutive iterations can be calculated as:

$$\Delta = | \delta_a - \delta_b | \tag{3}$$

In this particular example, the gain in fit after one iteration is 0.8173. The entire process is continued until the gain in fit is negligible or below a preset tolerance level. This tolerance level serves as the convergence criterion at which the algorithm is terminated. The weights are thus adjusted iteratively until the value of Δ is less than a small value, ε (say, 1 x 10^{-7}). Convergence criteria can be set as a reasonable compromise between desired goodness-of-fit and computation time. Figure 1 shows the reduction in δ value as the number of iterations increases. The plot shows values of δ on the Y-axis on a logarithmic scale and the number of iterations along the X-axis. It can be seen that, after about 80 iterations, the curve flattens out to a value very close to zero. After just 80 iterations, the average absolute relative difference across all constraints, δ, has reduced to 0.01, and after about 250 iterations, the δ value is 0.001.

Final weights, obtained after completion of 638 iterations, for the households in the small example are shown in Table 1. The corresponding δ value is very small (8.51 x 10^{-6}), showing that the weighted sums almost perfectly match the household type and person type constraints. It can be seen that the household weights for households belonging to a particular household type are no longer identical. Essentially, the household weights have been reallocated so that both weighted household and person sums match (in this case, perfectly) the given constraints (see the row titled "final weighted sum").

A Geometric Interpretation of the Algorithm

This subsection of the paper is devoted to explaining the logic underlying the IPU algorithm. Suppose there are two households belonging to the same household type (defined by a set of control variables of interest such as household size and income). Let the first household (household 1) not have any persons of a particular person type and let the second household (household 2) have one member belonging to that person type category. If the given constraints for the household type and the person type are respectively 4 and 3, the weights for satisfying the person and household constraints can be obtained by solving the following simultaneous linear equations:

$$\begin{aligned} w_1 + w_2 &= 4 \\ w_2 &= 3 \end{aligned} \tag{4}$$

where w_1 and w_2 represent the weights for household 1 and 2. Straight lines representing the two linear equations in equation (4) can be plotted as shown in Figure 2a with w_1 plotted on the vertical axis and w_2 plotted on the horizontal axis. The point of intersection, *I*, denotes the solution to the simultaneous equations in equation (4) and the coordinates of the point serve as the weights satisfying the given household and person constraints.

The proposed IPU algorithm provides a mechanism by which one can iteratively reach the point of intersection (solution) by starting anywhere in the quadrant defined by $w_1 > 0$, $w_2 > 0$. Consider an arbitrary starting point, *S*, as shown in the figure. A line is drawn connecting the starting point (S) and the origin such that it intersects the line $w_1 + w_2 = 4$ at point *B* and intersects the line $w_2 = 3$ at point *A*. When the weights are proportionally updated with respect to the first constraint, the starting point is moved to point *B* as the coordinates of the starting point will be scaled by the same ratio in order to satisfy that constraint. Then, the proportional update according to the second constraint will move the weight coordinates from point *B* to point *C*, because the update only changes the value of w_2 (in this example), but does not change the value of w_1. It is to be noted that point *C* can be obtained by drawing a line originating at point *B* perpendicular to the line $w_2 = 3$. In the second iteration, the weight coordinates move from point *C* to point *D*, which is the intersection of line *OC* with the line $w_1 + w_2 = 4$. The subsequent update (with respect to the second constraint) will move the weight coordinates from point *D* to point *E*, which can be obtained by drawing a line from point D perpendicular to the line $w_2 = 3$. It can be seen that, after every iteration, the weight coordinates are moving closer to the solution represented by the point of intersection *I*. The process is repeated until the weight coordinates move as close as possible to point *I*.

In practice, one is dealing with many constraints and the solution is unlikely to neatly fall into the first quadrant (perfect solution that matches both household and person constraints exactly). For example, the solution may be in the fourth quadrant as shown in Figure 2b. The figure shows a situation where the second constraint is changed to $w_2 = 5$. If such a situation is encountered, the algorithm will move the weight coordinates closer to the solution point, *I*, but will never be able to reach the point *I*. Eventually, the algorithm will move the weight coordinates back and forth (from one iteration to the next) between the points I_1 and I_2 where the two constraint equations intersect with the horizontal axis. In this instance, one can adopt a corner solution, usually corresponding to that which perfectly matches household constraints of interest. Alternatively, one could adopt a solution in between I_1 and I_2 which would represent a compromise between matching household and person distributions. Given the historical precedence given to matching household attributes of interest (Beckman et al., 1996), the algorithm currently adopts a corner solution corresponding to this requirement. However, it is to be noted that, even in this case, the algorithm provides considerable improvement in the match of person-level attributes over algorithms that do not adjust weights iteratively for both household and person level attributes.

Another theoretical situation where the IPU algorithm will fail is that where a group of households belonging to the same household type account for all of the individuals of a particular person type. That is, if one or more person types fall or appear exclusively in households of a single household type, then the algorithm will not be able to converge to a solution. This happens because one obtains two straight lines representing the household and person constraints, neither of which is perpendicular to the coordinate axes. The IPU algorithm will only be able to move the weight coordinates back and forth between two points on the lines, but cannot move them closer to the point of intersection. This situation is virtually never encountered in practice and is of little consequence for most practical applications of the algorithm. However, it is good practice to check for this potential problem when selecting the control variables of interest against which synthetic population joint distributions must match observed distributions. In the rare situation where this problem exists, the dimensions of interest can be changed or categories with very few households can be consolidated to overcome the problem.

The geometric example described in this subsection corresponds to a situation with only two dimensions (w_1, w_2). This two-dimensional example can be easily extended to three or more dimensions. For example, if there are three households belonging to the same household type, and one household constraint and one person constraint need to be satisfied, then the constraints can be represented by planes in a three-dimensional space. The potential solution should be located on the line formed by the intersection of the two planes. It is possible to have an infinite number of solutions on that line or no solution at all in the instance when the line formed by the intersection of the planes lies in a space characterized by negative coordinates. In most practical contexts, the IPU algorithm will reach a solution as long as solution(s) exist and all households of a single category do not account for all of the persons of a particular type. The latter condition ensures that there is a plane perpendicular to one of the coordinate axes, which is critical to the progress of the IPU algorithm.

In summary, the IPU algorithm provides a flexible mechanism for generating a synthetic population where both household and person-level attribute distributions can be matched very closely. The IPU algorithm works with joint distributions of household and person attributes derived using the IPF method, and then iteratively adjusts and reallocates weights across households such that both household and person-level attribute distributions are matched as closely as possible. The algorithm is flexible in that it can accommodate a multitude of household and person-level variables of interest and meets dual household- and person-level constraints with reasonable computational time. These are some of the noteworthy features of the algorithm that distinguish it from previous synthetic population generation algorithms.

The General Iterative Proportional Updating (IPU) Algorithm

The IPU algorithm illustrated using the small examples in the previous subsections can be easily extended to accommodate multiple household and person type constraints in the estimation of suitable weights. This subsection presents the general formulation of the algorithm in a step-by-step manner. The steps are as follows:

1. Generate a frequency matrix *D* showing the household type and the frequency of different person types within each household for the sample. The dimension of the matrix generated will be $N \times m$, where *N* is the number of households in the sample and *m* is the number of population characteristic (household type and person type) constraints. An element in the matrix $d_{i,j}$ represents the contribution of household *i* to the frequency of population characteristic (household type/person type) *j*.
2. Obtain joint distributions of household type and person type constraints using the standard IPF procedure and store the resulting estimates into a column vector *C* where c_j represents the value of the population characteristic *j* and $j = 1, 2, \ldots, m$.
3. Initialize the weights vector represented by the column vector, *W*, such that $w_i = 1$ where $i = 1, 2, \ldots, N$.

   Also, initialize a scalar, $\delta = \dfrac{\sum_j \left[\left| \left(\sum_i d_{i,j} w_i - c_j \right) \right| / c_j \right]}{m}$ and set the value of the scalar, $\delta_{\min} = \delta$.

4. Initialize a scalar, $r = 1$, representing the iteration number.
5. For each column j ($j = 1, 2, \ldots, m$), record the indices (i.e., the row number or, in the context of the simple example, the household ID) into a column vector S_j, including only those that actually belong to household or person type j. Let an entry in such a column vector be denoted by s_{qj} where q is an index corresponding to non-zero elements in the jth column. For instance, in the simple example considered in Table 1, S_1 would include elements (households) 1, 2, and 3; S_2 would include elements 4, 5, 6, 7, and 8; and so on.
6. Initialize a scalar $k = 1$ to serve as a constraint counter.
7. Retrieve the indices s_{qk} of all the non-zero elements in the kth column stored in S_k of Step 5 where q is the index corresponding to non-zero elements in the kth column.
8. Calculate the adjustment ρ for the kth constraint, $\rho = \dfrac{c_k}{\sum\limits_q d_{s_{qk},k} \times w_{s_{qk}}}$
9. Update the weights with respect to the kth constraint as $w_{s_{qk}} = \rho \, w_{s_{qk}}$. Recall that all initial weight values are set to one.
10. Update $k = k + 1$.
11. If $k \leq m$, i.e., the weight have not been adjusted with respect to all population characteristic constraints, then go to Step 7; otherwise, proceed to Step 12.
12. Set the value of a scalar, $\delta_{prev} = \delta$.
13. Calculate the new value of δ corresponding to the current iteration,
$$\delta = \frac{\sum_j \left[\left| \left(\sum_i d_{i,j} w_i - c_j \right) \right| / c_j \right]}{m}.$$
14. Calculate the improvement in goodness-of-fit, $\Delta = | \delta - \delta_{prev} |$.
15. If $\delta < \delta_{min}$, update $\delta_{min} = \delta$, and store the corresponding weights in a column vector SW with elements $sw_i = w_i$ for $i = 1, 2, \ldots, N$. Otherwise, proceed to Step 16.
16. Update the iteration number, $r = r + 1$.
17. If $\Delta > \varepsilon$ (a small positive number, e.g., 1×10^{-4}), go back to step 6. Otherwise, convergence has been achieved and a solution is obtained. The selected weights are stored in the column vector SW corresponding to the smallest absolute relative difference δ_{min}.

The updated household weights are recorded in the column vector SW. It should be noted that Step 15 in the algorithm is critical because the δ value is not always strictly decreasing. As a result, it is necessary to ensure that weights corresponding to the minimum value of δ are retained at each iteration of the process.

At the conclusion of the process outlined above, a perfect solution is obtained if it falls within the feasible range (positive quadrant in Figures 2a and 2b). If, however, the solution does not fall within a feasible range, then additional steps may be warranted to choose the appropriate corner solution. Given the emphasis on matching household-level constraints in current practice, the additional steps in the procedure proceed to the corner solution to ensure that household constraints are met perfectly. The steps are:

18. Initialize a scalar $h = 1$, where $h = 1, 2, \ldots m_h$, where m_h is the number of household constraints that need to be satisfied.

19. Retrieve the indices s_{qh} of all the non-zero elements in the h^{th} column stored in column vector S_h of Step 5.
20. Calculate the adjustment ρ for the h^{th} constraint, $\rho = \dfrac{c_h}{\sum_q d_{s_{qh},h} \times w_{s_{qh}}}$
21. Update the weights with respect to the h^{th} constraint as $sw_{s_{qh}} = \rho\, sw_{s_{qh}}$.
22. Update $h = h + 1$.
23. If $h \leq m_h$, go back to Step 18; otherwise, a corner solution has been reached and the algorithm is terminated.

POPULATION SYNTHESIS FOR SMALL GEOGRAPHIC AREAS

Population synthesis should be conducted at the smallest level of geography for which data is available. This allows for retaining the location information and for maintaining the heterogeneity in attributes across the region when allocating the households to the street network. This is not possible if the population synthesis is conducted and households are allocated to the roadway network for a larger geography. This aspect of population synthesis has been previously recognized and implemented in various studies including that by Beckman et al. (1996) who synthesized populations at the census tract level and Guo and Bhat (2006) who synthesized populations at the blockgroup level. In this study, the algorithm was used to synthesize a population for the entire Maricopa County region in Arizona. This region has a population of more than three million persons in a little over 2000 blockgroups. In the next section of this paper, results of population synthesis for two sample blockgroups are presented to demonstrate the real-world applicability of the IPU algorithm presented in this paper. However, prior to doing that, it was considered beneficial to discuss solutions to two particular problems commonly encountered in dealing with small geographies. This section presents a discussion of the two issues and the approaches adopted in this study to tackle these problems.

Zero-cell Problem

Beckman et al. (1996) suggested using joint distributions of PUMS households belonging to the Public Use Micro Area (PUMA) as prior information for estimating a blockgroup's joint distribution with the IPF procedure. However, this practice may lead to the zero-cell problem where a few cells may end up with zero frequencies for small geographies. This may happen because the PUMA sample is a subsample of the PUMS, and some demographic groups may not appear in the joint tabulations although they are present in the marginal distributions of the population in the small geographic areas under consideration. Guo and Bhat (2006) also note that this problem exists for small geographies. The IPF procedure cannot converge to a solution when the zero-cell problem exists. Beckman et al. (1996) recommended adding an arbitrarily small value (e.g., 0.01) to zero cells in order to make the IPF procedure converge to a solution. However, Guo and Bhat (2006) note that this treatment may introduce an arbitrary bias.

In this paper, an alternative method to account for the zero-cell problem is proposed. The idea underlying the approach is to borrow the prior information for the zero cells from PUMS data for the entire region (where zero cells are not likely to exist as long as the control variables of interest and their categories are defined appropriately). It may be reasonable to use probabilities estimated from the PUMS representing the entire region as substitutes for the zero-cells in the small geography where this problem exists. However, caution should be exercised when adopting such an approach because there is a risk of over-representing the demographic group, which was rare in the small geography in the first place (as evidenced by the zero-cell value). For example, suppose a PUMA contains 5000 households and let the probability associated with a zero-cell demographic group be 0.001 in the PUMS as a whole. If one were to borrow this probability directly, then the frequency for that demographic group in the PUMA is expected to be 5. However, the fact that the PUMA sample does not contain the demographic group at all suggests that the probability of occurrence of this group in the PUMA is likely to be less than the borrowed 0.001. To overcome this potential problem, the approach in this study implements an upper-bound or threshold approach for borrowing probabilities. If the threshold frequency for the zero cells is assumed to be unity, the upper bound of the borrowed probability may be considered to be 1 ÷ the total number of households in the PUMA. In this case, that would be 1/5000 = 0.0002. If the borrowed probability (from the PUMS) is less than this upper bound, it can be used to replace the zero cells. Otherwise, the upper bound itself is used to replace the zero cells. This procedure ensures that the estimated frequency for the zero-cells (infrequent demographic groups) is not over-estimated in the PUMA.

It is to be noted that this approach (where zero cells are replaced with borrowed or threshold probabilities) will result in all probabilities adding up to a value greater than unity. To correct this inconsistency, all of the non-zero cell probabilities are scaled down by a ratio, $y = 1 - u$, where u is the sum of the borrowed probabilities. After this adjustment, all of the probabilities will add up to unity and the ratio of probability values between each pair of original non-zero cells will remain unchanged. The procedure outlined in this section can be used to replace zero cells for certain rare demographic groups using borrowed probability values from the PUMS subject to an upper limit. The IPF procedure can now be executed on the modified PUMA priors to estimate household and person level joint distributions for populations in virtually any small geography.

The Zero-Marginal Problem

The zero-marginal problem is encountered in the context of the IPU algorithm proposed in this paper. In small geographical areas, it is reasonable to expect the marginal frequency distribution to take a value of zero for certain attributes. For example, it is possible to have absolutely no low-income households residing in a particular blockgroup. If so, all of the cells in the joint distribution corresponding to the low income category will take zero values as a result of the execution of the IPF procedure. Then, when weights are computed using the proposed IPU algorithm (to match the zero constraints), all of the households in the PUMS contributing to that particular zero-marginal household type will take zero weights. However, it is entirely possible that at least some of these households need to take non-zero weights to satisfy other non-zero constraints. The iterative algorithm allows weights to be updated as one proceeds from one constraint to the next, but the algorithm cannot do so when certain households have zero weights. As the denominator will take a zero value in the calculation of the adjustment to the weights (as shown in Step 8 of the generalized procedure), the algorithm fails to proceed. To overcome this problem, a small positive value (e.g., 0.01) is assigned to all zero-marginal categories. The IPF procedure will distribute or allocate this small value to all of the relevant cells in the joint distribution and the cells in the resulting joint distribution will differ by a small amount compared to the situation where no adjustment is made to account for zero-marginals. The effect of this adjustment on final IPF results is virtually negligible; however, this adjustment allows the

IPU algorithm to move forward with the adjustment of weights across all households to meet both household and person constraints.

CASE STUDY

This section presents results of the application of the proposed IPU algorithm for two random, but illustrative, blockgroups in the Maricopa County region of Arizona using year 2000 census data files. According to the 2000 census, the Maricopa County region had a population of 3,071,219 persons residing in 1,133,048 households in 2,088 blockgroups (25 blockgroups had zero households). Marginal distributions on household and person variables were obtained from the Summary Files while the five-percent PUMS served as the sample from which to draw households for generating the synthetic population. The PUMS included 254,205 persons residing in 95,066 households.

Calibration of Weights

For both blockgroups, the IPU algorithm was applied to satisfy household and person joint distributions generated using the standard IPF procedure with the adjustments noted in the previous section for zero-cell and zero-marginal problems. The household joint distributions were generated using household type, household size, and household income as control variables while the person-level joint distributions were generated using gender, age, and ethnicity as control variables. Table 2 presents a summary of the household and person-level variables and categories used in this case study. The categorization scheme adopted in Table 2 resulted in 280 cells for household level joint distribution and 140 cells for the person level joint distribution. As explained earlier, the IPF procedure was applied to generate joint distributions based on marginal distributions obtained from Census Summary Files. Prior probability information on the joint distribution was obtained from the subsample of the PUMS corresponding to the PUMA to which the blockgroup belonged.

The proposed IPU algorithm was applied to calibrate household weights to match both household and person level joint distributions. Figure 3a illustrates the reduction in average absolute relative difference across all constraints (δ). For the first blockgroup, the δ value reduced from 2.471 to 0.041 in 20 iterations. The δ value for this blockgroup indicates that the algorithm reached a corner solution and the calibrated household weights result in a perfect match of household-level distributions, while simultaneously resulting in substantial reductions in inconsistencies with respect to person-level distributions (in comparison to traditional population synthesis procedures that generally match household attribute distributions). For the second blockgroup, the value of δ reduces from 0.8151 to 0.00064 in 500 iterations as shown in Figure 3b. Although 500 iterations were run for purposes of demonstration, about 100 iterations are sufficient to provide satisfactory weights in practice. It can be seen that the average absolute relative difference (δ) is very close to zero, suggesting that a near-perfect solution was reached to match both household and person-level joint distributions of attributes.

Drawing of Households

After the calibration of weights using the IPU algorithm, households may be randomly drawn from the PUMS to generate the synthetic population. The approach adopted in this study is similar to that adopted by Beckman et al. (1996), except that the probability with which a household is drawn is dependent on its assigned weight (from the IPU algorithm). In the Beckman et al. (1996) procedure, the household level joint distributions obtained from the IPF procedure are rounded off to the nearest integer (because the IPF procedure can result in many cells with decimal values). Then, for each household type, households are drawn randomly from the set of PUMS households that belong to that particular category. The number of households randomly drawn is equal to the frequency of that household type in the rounded joint distribution table. This ensures that errors in the household level distribution between that of the synthetic population and that of the original PUMS will be no more than 0.5.

The Beckman et al. (1996) procedure is appropriate in the context of matching household-level joint distributions. However, as the objective of the IPU algorithm is to match both household and person-level attribute distributions, different weights are assigned to different households that fall within the same household type category (because they presumably have different types of persons). Hence the approach for drawing households to constitute the synthetic population is slightly different in this approach than that in Beckman et al. (1996). In the IPU approach, the probability of a household being chosen is equal to its weight divided by the sum of weights of all households belonging to a particular household type. As the IPU-calibrated weights not only control for household-level attributes, but also person-level attributes, this weight-based procedure for probabilistically drawing households results in the generation of a synthetic population that closely matches household and person-level distributions.

As noted previously, the household level joint distributions are rounded off to the nearest integer. As a result of this rounding, the total number of households in the synthetic population will not be equal to the total number of households from the Census Summary Files. As quite a few cells in the joint distribution take on a value less than one, rounded household totals will, more likely than not, be less than original household totals in the Summary Files. For the two blockgroups considered here, total households in the synthetic population and original population were respectively 615 and 627 in the first blockgroup and 456 and 462 in the second blockgroup. This difference in household totals was resolved in the following manner. The cell frequencies in the joint distributions of household attributes were compared – between that of the synthetic population and that generated by the IPF procedure using the PUMS subsample for the PUMA to which the blockgroup belonged. Differences in cell values were computed and arranged in descending order; then, the top-ranked 12 cells in the first blockgroup and the top-ranked six cells in the second blockgroup each received an additional household to resolve the difference. The cells were essentially ordered in descending order of the differences between the synthetic population cell frequencies and the observed IPF-generated cell frequencies. Thus, one household was added to those cells where the discrepancies were largest.

Performance Measures

The IPU algorithm generates household-specific weights such that household-level distributions are matched perfectly and person-level distributions are matched as closely as possible. As per the discussion in the previous subsection, households are drawn probabilistically using a Monte Carlo procedure in accordance with the weight that is allocated by the algorithm. As the Monte Carlo procedure constitutes a probabilistic mechanism for drawing households, it is recommended that multiple synthetic populations be drawn and the synthetic population that best matches the person-level attribution distributions be chosen (note that the household-level distributions are always matched perfectly).

It is necessary to establish a criterion by which the performance of the synthetic population generator (in matching joint distributions) can be assessed. The average absolute relative difference (δ value) used in the course of the algorithm itself is useful, but has a shortcoming particularly in the context of small geographies. The δ value cannot be used as an appropriate measure of fit because it masks the differences in magnitude between the estimated and desired joint distributions. For example, the absolute relative error for a cell in the synthetic population that takes a value of 0, when the desired value is 0.2, is 100 percent. Another cell that takes a value of one, when the desired value is 0.5, also has an absolute relative error of 100 percent. And yet another cell that takes a value of 200, when the desired value is 100, also has an absolute relative error of 100 percent. While the differences in magnitude between the estimated and desired values may be acceptable in the first two scenarios, it is clearly not acceptable in the third scenario. However, all three scenarios offer the same absolute relative error/difference values. From that standpoint, the δ value may not be a good measure for comparing the estimated and desired joint distributions and assessing the fit of the algorithm.

An alternative measure of fit is the chi-square (χ^2) statistic which is often used to statistically compare two distributions of interest. The χ^2 statistic, which serves as an appropriate measure of fit in the context of this study, may be calculated as follows:

$$\tau = \sum_j \left[\frac{\left(\sum_i d_{i,j} w_i - c_j \right)^2}{c_j} \right] \tag{5}$$

where i denotes household type ($i = 1, 2, ..., N$)

j denotes the constraint or population characteristic of interest ($j = 1, 2, ..., m$)

$d_{i,j}$ represents the frequency of the population characteristic (household/person type) j in household i

w_i is the weight attributed to the i^{th} household

c_j is the value of the population characteristic j.

It should be noted that those cells which have zero values in the observed joint distribution (obtained by applying the standard IPF procedure on PUMS subsamples) are not included in the calculation of this statistic. The test statistic, τ, follows a χ^2 distribution with (J-1) degrees of freedom. Using the τ value, one may test the null hypothesis that the estimated frequency distribution matches the observed frequency distribution. The expression $[1 - \chi^2_{\text{J-1}}(\tau)]$ provides the probability of incorrectly rejecting the null hypothesis when, in fact, it is true, where $\chi^2_{\text{J-1}}(.)$ represents the cumulative distribution function of the χ^2 distribution with (J-1) degrees of freedom. Also, the value of $\chi^2_{\text{J-1}}(\tau)$ represents the level of confidence at which the estimated frequency distribution is considered to match the observed frequency distribution.

The value of $\chi^2_{\text{J-1}}(\tau)$ serves as an appropriate measure of fit with respect to matching person-level distributions (note again that household level distributions are matched perfectly). From 100 randomly drawn synthetic populations for the first blockgroup in this case study, it was found that about 27 populations exhibited a confidence level of less than 0.l, while 21 populations exhibited a confidence level greater than 0.9, when the synthetic person-level joint distribution and the IPF-generated person-level joint distribution were compared. Based on these numbers, it is possible to approximate the number of draws required to obtain a synthetic population with the desired person-level distribution. It is estimated that one should draw a synthetic population at least 13 times to ensure a high degree of performance wherein the probability of obtaining at least one synthetic population with a confidence level better than 0.9

is more than 0.95. As the probability can be calculated as $[1 - (1 - 0.21)^n]$, where n represents the number of trials, n must be greater than 13 to obtain a probability level greater than 0.95. However, in view of computational performance considerations, it is desirable to limit the number of trials. For the two blockgroups considered in this case study, 20 synthetic populations were created using the described probabilistic procedure and the one with the best τ was chosen.

The best synthetic population (lowest τ value) for the first blockgroup provided a χ^2 value of 74.77 with 119 degrees of freedom. The corresponding p-value is 0.999, indicating a high level of confidence that the estimated joint distribution (of person attributes) matches the observed distribution. Similarly, the best synthetic population for the second blockgroup provided a χ^2 value of 52.01 with 99 degrees of freedom (recall that zero cells are omitted). The corresponding p-value is 1.000, once again indicating a high level of confidence that the estimated and observed joint distributions match each other.

The similarity between the synthetic population person-level distribution and the IPF-generated person-level distribution is visually illustrated in Figures 4a and 4b. A point in each scatter plot represents one cell in the person-level joint distribution of interest. The Y-coordinate represents the cell frequency in the synthetic population, while the X-coordinate represents the cell frequency in the observed IPF-generated joint distribution. If the corresponding cell frequencies match perfectly, then the points should fall on a 45^o straight line. Two sets of points are plotted for comparison purposes. One set of points is that obtained using traditional IPF procedures for generating synthetic populations. In this procedure, household attribute distributions are matched, and then entire households are randomly drawn without consideration for matching person-level distributions. The other set of points is that obtained using the IPU algorithm proposed in this paper, wherein both household and person-level distributions are matched as closely as possible. It can be seen from the plots that, for both blockgroups, the proposed IPU algorithm better replicates the observed or desired IPF-generated joint distribution. The cell frequency points corresponding to the comparison between the IPU and IPF-generated person-level joint distributions fall more closely along the 45^o line than the set of points obtained using standard traditional IPF procedures for generating a synthetic population. It is also noteworthy that the χ^2 test statistic assessing the fit of the person-level joint distribution obtained using the traditional IPF procedure strongly rejects the null hypothesis that the synthetic population distribution matches the observed joint distribution. All of these findings suggest that the adjustments in weights made by the IPU algorithm offer considerable benefits in matching person-level distributions while ensuring that household-level attributes are matched perfectly.

The proposed IPU algorithm was applied to generate a synthetic population for the entire Maricopa County region of Arizona. Synthetic populations were created for all 2088 blockgroups that had a positive number of households (zero household blockgroups were not included in the population generation procedure). A Dell Precision Workstation with Quad Core Intel Xeon processor was used to run the entire algorithm. The algorithm was coded in Python – a dynamic open-source object-oriented programming language and the data was stored and accessed using MySQL – a commonly used open source database solution. The code was parallelized to take advantage of the multiple cores in the processor. The total processing time in a single-core configuration was approximately 16 hours, an average of 27 seconds per blockgroup. As the code was parallelized to take advantage of the Quad Core processor, the computational time to generate an entire synthetic population was reduced to approximately four hours, an average of seven seconds per blockgroup. A dual–core processor would have resulted in a computation time of eight hours, an average of 14 seconds per blockgroup. The presence of multiple cores does not necessarily reduce total processing time, but does reduce the computation time taken to probabilistically generate synthetic populations because of the parallel processing.

According to the 2000 Census Summary Files, a total of 3,071,219 people resided in 1,133,048 households within the 2088 blockgroups. The synthetic population generated using the IPU algorithm resulted in a virtual match in terms of the number of households, with the number of synthetic households generated equal to 1,133,039. The number of synthetic persons generated using the IPU algorithm was equal to 2,929,679, a 4.6 percent difference relative to the actual number of persons in the population.

CONCLUSIONS

This paper presents a heuristic iterative approach, dubbed the Iterative Proportational Updating (IPU) algorithm, for generating a synthetic population while simultaneously matching both household-level and person-level joint distributions of control variables of interest. In traditional population synthesis procedures, only household-level joint distributions are matched using the standard iterative proportional fitting (IPF) procedure, and entire households are randomly drawn according to the the IPF-generated household-level joint distribution. Little regard is given to matching person-level distributions (all of the persons in the drawn households become part of the synthetic population). This results in a synthetic population wherein the household distributions are matched, but the person-level distributions are not likely to be matched well simply because all households in a certain cell of the joint distribution receive the same weight. In contrast to that approach, the IPU algorithm iteratively adjusts and reallocates weights among households so that person-level distributions are matched as closely as possible without compromising on the fit to household-level distributions. The paper presents the algorithm in detail, illustrates the algorithm using a small example, and then offers real-world case study results for small geographies (blockgroups) in the Maricopa County region of Arizona.

It is to be noted that the total number of household or person types (i.e., the number of cells in the joint distributions) plays an important role in determining the degree to which the person attributes will be matched by the proposed IPU algorithm. As the level of disaggregation increases, the frequency in each cell will drop. When this happens, one has less flexibility in reallocating weights across households of a certain type (cell) and this results in a poorer ability to replicate or fit to person-level joint distributions. In particular, cells that have only one household or one individual hinder the ability to match person-level distributions. In applying the IPU algorithm, one should examine cell frequencies in joint distributions carefully and consolidate sparse categories so that the performance of the IPU algorithm can be enhanced. The IPU algorithm will function even in the presence of sparse cells; however, the fit with respect to person-level distributions will not be as good as what might have been achieved in the absence of the sparse cells. In the extreme case where all persons of certain types completely fall into a single household type, the algorithm fails. Although this situation rarely, if ever, occurs in practice, it is still worthwhile to examine joint distributions and ensure that this problem is avoided by consolidating appropriate household or person categories where cell frequencies are very small. In ongoing and future research, the authors are analyzing the sensitivity of the algorithm performance with respect to the number of cells that have small values and the frequency of observations in these cells. This ongoing research will provide a basis to determine cell threshold values at which the performance of the algorithm is seriously compromised.

The proposed algorithm has the potential to be useful in other applications as well. For example, the algorithm may be applied to calibrate survey weights for travel survey samples.

Household travel surveys involve collecting data from samples of respondents at the household, person, vehicle, and trip level. Often, one encounters data sets where there are three or four sets of weights corresponding to each of these behavioral units to match survey distributions against known population distributions. Using the IPU algorithm, it may be possible to generate a unique set of weights that can simultaneously satisfy household, person, vehicle, and trip-level distributions of interest, and reduce the need to generate and maintain multiple weights at each level of analysis. In general, any application where one is interested in weighting and expanding a sample to simultaneously match or fit to multiple distributions (constraints) may benefit from the use of the IPU algorithm proposed in this paper. Future research efforts will involve the incorporation of model components that evolve the synthetic population over time (Goulias and Kitamura, 1996).

ACKNOWLEDGEMENTS

This research effort was partially funded by the Federal Highway Administration of the US Department of Transportation under contract DTFH61-08-C-00010 of the Exploratory Advanced Research Program and by the US Environmental Protection Agency under Grant RD-83183701 of the Science To Achieve Results (STAR) program. The authors benefitted from discussions with Dr. Hillel Bar-Gera of Ben-Gurion University, Israel, who was a visiting professor at Arizona State University when some of this work was undertaken. All errors and omissions are the sole responsibility of the authors.

REFERENCES

Arentze, T.A., and H.J.P. Timmermans. A Learning-Based Transportation Oriented Simulation System, *Transportation Research Part B: Methodological*, **38(7)**, 2004, pp. 613-633.

Arentze T., H.J.P. Timmermans, and F. Hofman. Creating Synthetic Household Populations: Problem and Approach. *Transportation Research Record: Journal of the Transportation Research Board*, **2014**, 2007, pp. 85-91.

Beckman, R.J., K.A. Baggerly, and M.D. McKay. Creating Synthetic Baseline Populations. *Transportation Research Part A: Policy and Practice*, **30(6)**, 1996, pp. 415-429.

Bhat, C.R., J.Y. Guo, S. Srinivasan, and A. Sivakumar. Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns. *Transportation Research Record*, **1894**, 2004, pp. 57-66.

Deming, W.E., and F.F. Stephan. On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals Are Known. *Annals of Mathematical Statistics*, **11**, 1940, pp. 427-444.

Fienberg, S.E. An Iterative Procedure for Estimation in Contingency Tables. *Annals of Mathematical Statistics*, **41**, 1970, pp. 907-917.

Goulias, K.G., and R. Kitamura. A Dynamic Model System for Regional Travel Demand

Forecasting. In Golob, T., Kitamura R. (eds.) Panels for *Transportation Planning: Methods and Applications*, Kluwer Academic Publishers., 1996, pp. 321-348.

Guo, J. Y., and C.R. Bhat. Population Synthesis for Microsimulating Travel Behavior. *Transportation Research Record: Journal of the Transportation Research Board*, **2014**, 2007, pp. 92-101.

Ireland, C.T., and S. Kullback. Contingency Tables with Given Marginals. *Biometrika*, **55(1)**, 1968, pp. 179-188.

Kitamura, R., and S. Fujii. Two Computational Process Models of Activity-Travel Behavior. In T. Garling, T. Laitila, and K. Westin (eds.) *Theoretical Foundations in Travel Choice Modeling*, Pergamon, Elsevier, 1998, pp. 251-279.

Miller, E. J., and J. Roorda. A Prototype Model of Household Activity/ Travel Scheduling. *Transportation Research Record: Journal of Transportation Research Board*, **1831**, 2003, pp. 114-121.

Pendyala, R.M., R. Kitamura, A. Kikuchi, T. Yamamoto, and S. Fujii. Florida Activity Mobility Simulator: Overview and Preliminary Validation Results. *Transportation Research Record: Journal of Transportation Research Board*, **1921**, 2005, pp. 123 – 130.

Vovsha, P., M. Bradley, and J. Bowman. Activity-Based Travel Forecasting Models in the United States: Progress Since 1995 and Prospects for the Future. In H.J.P. Timmermans (ed.) *Progress in Activity-Based Analysis*, Elsevier Science, Oxford, 2005, pp. 389-414.

Wong, D.W.S. The Reliability of Using the Iterative Proportional Fitting Procedure. *Professional Geographer*, **44(3)**, 1992, pp. 340-348.

Table 1. An Example of the Iterative Proportional Updating (IPU) Algorithm

Household ID	Weights	Household Type 1	Household Type 2	Person Type 1	Person Type 2	Person Type 3	Weights 1	Weights 2	Weights 3	Weights 4	Weights 5	Final Weights
1	1	1	0	1	1	1	11.67	11.67	9.51	8.05	12.37	1.36
2	1	1	0	1	0	1	11.67	11.67	9.51	9.51	14.61	25.66
3	1	1	0	2	1	0	11.67	11.67	9.51	8.05	8.05	7.98
4	1	0	1	1	0	2	1.00	13.00	10.59	10.59	16.28	27.79
5	1	0	1	0	2	1	1.00	13.00	13.00	11.00	16.91	18.45
6	1	0	1	1	1	0	1.00	13.00	10.59	8.97	8.97	8.64
7	1	0	1	2	1	2	1.00	13.00	10.59	8.97	13.78	1.47
8	1	0	1	1	1	0	1.00	13.00	10.59	8.97	8.97	8.64
Weighted Sum		3.00	5.00	9.00	7.00	7.00						
Constraints		35.00	65.00	91.00	65.00	104.00						
δ_b		0.9143	0.9231	0.9011	0.8923	0.9327						
Weighted Sum 1		***35.00***	5.00	51.67	28.33	28.33						
Weighted Sum 2		35.00	***65.00***	111.67	88.33	88.33						
Weighted Sum 3		28.52	55.38	***91.00***	76.80	74.39						
Weighted Sum 4		25.60	48.50	80.11	***65.00***	67.68						
Weighted Sum 5		35.02	64.90	104.84	85.94	***104.00***						
δ_a		0.0006	0.0015	0.1521	0.3222	0.0000						
Final Weighted Sum		35.00	65.00	91.00	65.00	104.00						

Table 2. Household and Person Level Attributes Used for Case Study

Household Attributes	Description	Value
Household Type	Family: Married Couple	1
	Family: Male Householder, No Wife	2
	Family: Female Householder, No Husband	3
	Non-family: Householder Alone	4
	Non-family: Householder Not Alone	5
Household Size	1 Person	1
	2 Persons	2
	3 Persons	3
	4 Persons	4
	5 Persons	5
	6 Persons	6
	7 or more Persons	7
Household Income	$0 - $14,999	1
	$15,000 - $24,999	2
	$25,000 - $34,999	3
	$35,000 - $44,999	4
	$45,000 - $59,999	5
	$60,000 - $99,999	6
	$100,000 - $149,999	7
	Over $150,000	8
Person attributes		
Gender	Male	1
	Female	2
Age	Under 5 years	1
	5 to 14 years	2
	15 to 24 years	3
	25 to 34 years	4
	35 to 44 years	5
	45 to 54 years	6
	55 to 64 years	7
	65 to 74 years	8
	75 to 84 years	9
	85 and more	10
Ethnicity	White alone	1
	Black or African American alone	2
	American Indian and Alask Native alone	3
	Asian alone	4
	Native Hawaiian and Other Pacific Islander alone	5
	Some other race alone	6
	Two or more races	7

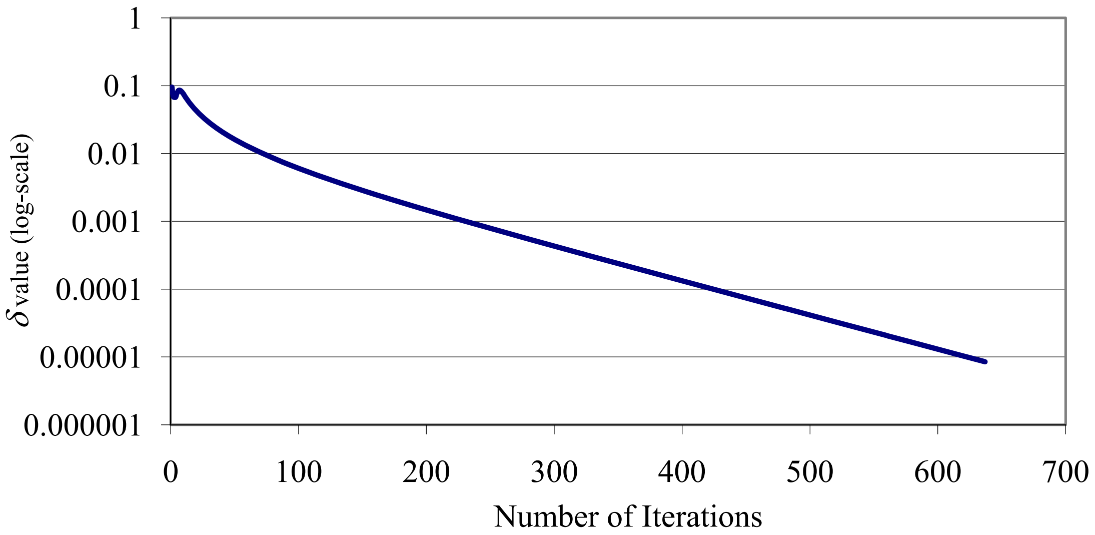

Figure 1. Reduction in Average Absolute Relative Difference for the Illustrative Example

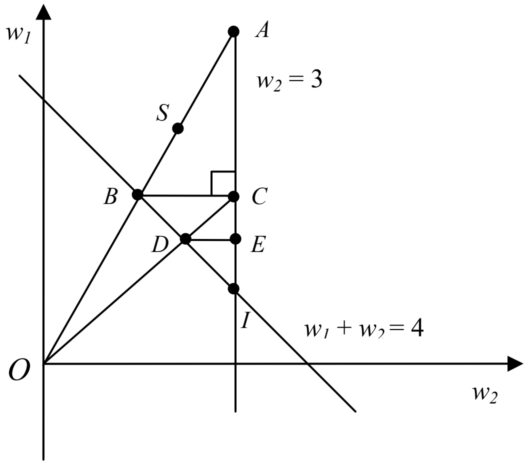

Figure 2a. Feasible Solution Case

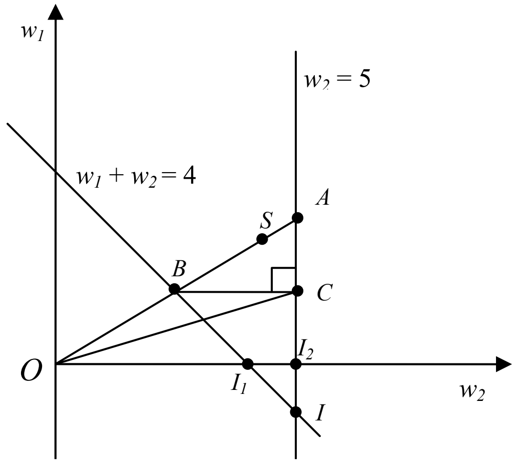

Figure 2b. Infeasible Solution Case

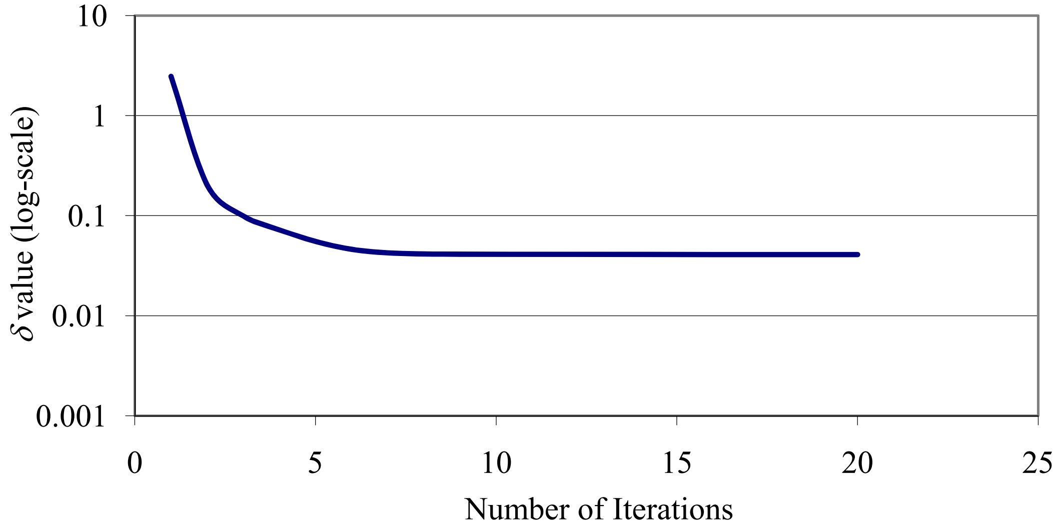

Figure 3a. Reduction in Average Absolute Relative Difference for the First Blockgroup

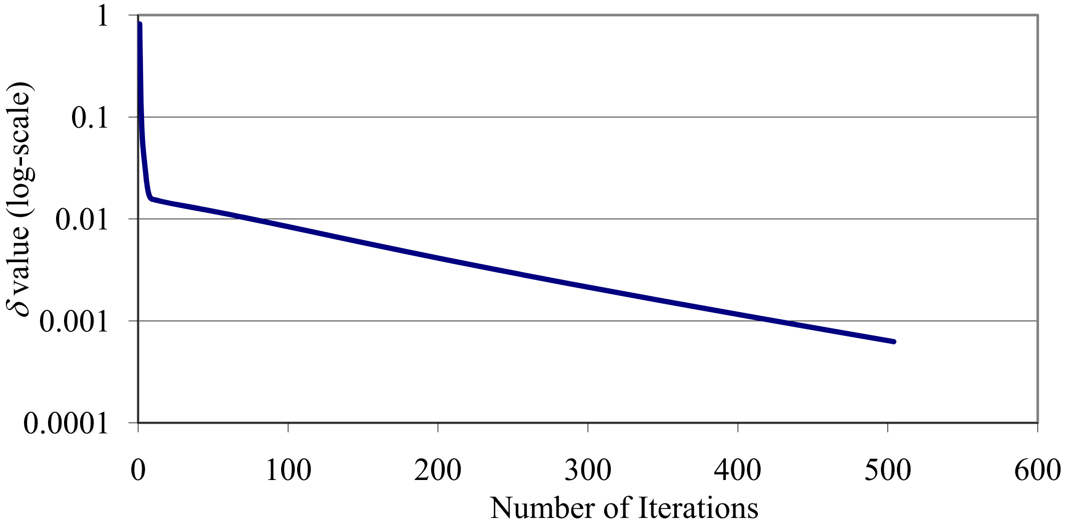

Figure 3b. Reduction in Average Absolute Relative Difference for the Second Blockgroup

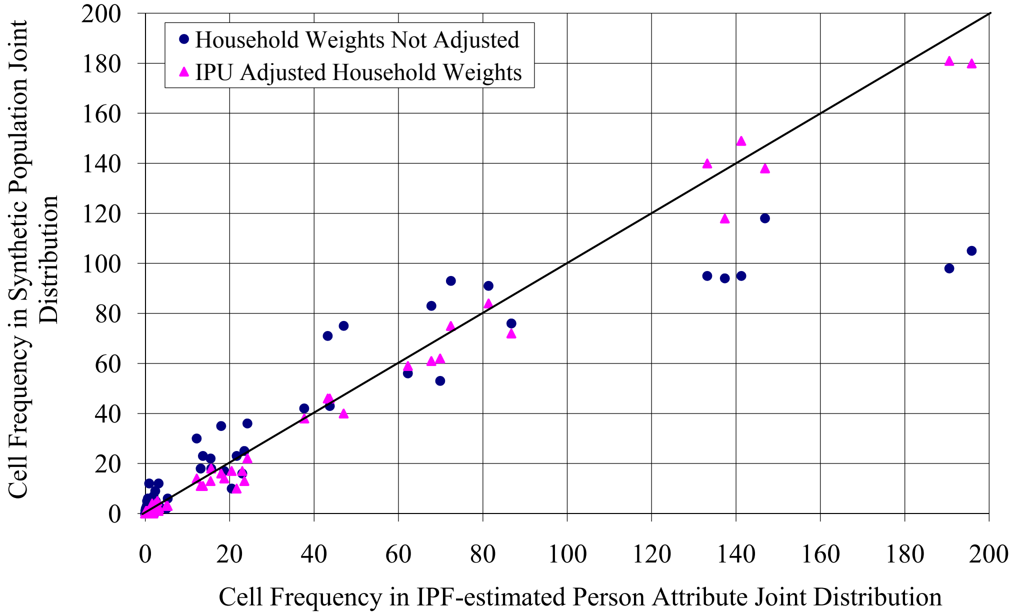

Figure 4a. Plot of IPF-estimated and IPU-Generated Person Joint Distribution Cell Frequencies for the First Blockgroup (Number of Cells = 140)

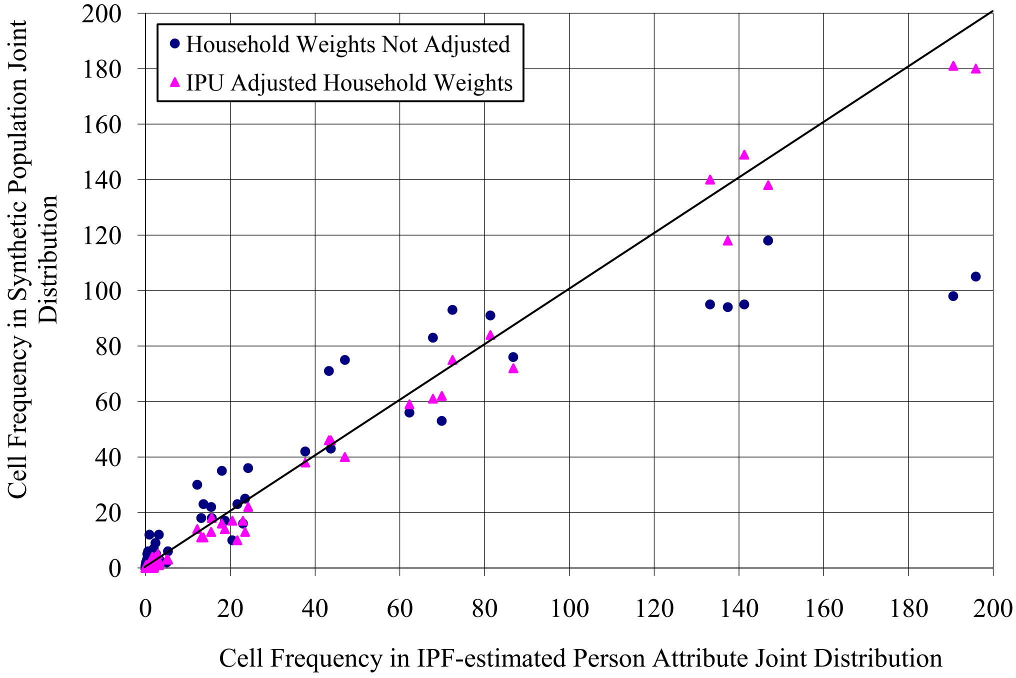

Figure 4b. Plot of IPF-estimated and IPU-Generated Person Joint Distribution Cell Frequencies for the Second Blockgroup (Number of Cells = 140)