Population synthesis for microsimulation: State of the art

Kirill Müller

Kay W. Axhausen

STRC 2010 **August 2010**

IVT Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Population synthesis for microsimulation: State of the art

Kirill Müller
IVT
ETH Zürich
8093 Zürich
phone: +41-44-633 33 17
fax: +41-44-633 10 57
kirill.mueller@ivt.baug.ethz.ch

Kay W. Axhausen
IVT
ETH Zürich
8093 Zürich
phone: +41-44-633 39 43
fax: +41-44-633 10 57
axhausen@ivt.baug.ethz.ch

August 2010

Abstract

In agent-based microsimulation models for land use (e.g., UrbanSim (2010)) or transportation planning (e.g., MATSim-T (2010)), agents' decisions are simulated over time in order to predict future states of the system. The initial step is the definition of agents – usually, persons and households. If a snapshot of the entire population of the study area, taken at the simulation's base year, were on hand, one could use this as an initial placement. Unfortunately, such data is often not available due to privacy and cost constraints. To tackle this issue, one can combine different data sources to derive a disaggregate representation of the agents, matching given criteria like correlation structure and marginal sums. This process is referred to as *population synthesis*.

We summarize recent efforts to population synthesis for microsimulation (Auld *et al.*, 2010; Pritchard and Miller, 2009; Ye *et al.*, 2009; Srinivasan and Ma, 2009; Arentze *et al.*, 2007; Guo and Bhat, 2007). All of the aforementioned works share two tasks: (a) adjustment of an initial population, taken from a past census or other survey data, to current constraints, and (b) selecting households and optionally assigning them to geographic areas. We describe the above tasks, and analyze and evaluate the characteristics of the particular approaches. This information will hopefully be helpful for the implementation of future population synthesis routines.

Keywords

Population synthesis, Microsimulation, Households, Disaggregation, IPF, Iterative Proportional Fitting

1 Introduction

Agent-based microsimulation models for land use (e.g., UrbanSim (2010)) or transportation planning (e.g., MATSim-T (2010)) become more and more widespread. These models simulate agents' decisions over time in order to predict future states of the system. They allow for more detailed and accurate simulation and prediction of, e.g., land pricing and travel demand than traditional aggregate models. However, they also require disaggregate input data.

When implementing such a model, the initial step is the definition of *agents* and their relationships. Most frequently, in this context, the agents of the microsimulation represent the individual people living in the study area, grouped by households. Other kinds of agents and relationships might be of interest as well, such as employees/firms, vehicles/households, tenants/buildings (Auld *et al.*, 2010).

One feasible source for such disaggregate data is the national census that is collected for many countries on a regular basis. However, some issues impede the utilization of untreated census data as input for microsimulation. First, the complete census is rarely available: In many countries, only a small subsample, the so-called *public-use sample*, can be accessed. Information in the sample may be randomly rounded, aggregated, or removed altogether. Second, the census is collected rather infrequently, as much as 10 years can pass between two consecutive surveys. This restricts the choice of the base year for the microsimulation model.

The objective of *population synthesis* is to compensate for the difficulties named above. The main idea, as pioneered by Beckman *et al.* (1996), is to combine the census data with readily available up-to-date aggregate data. This results in a set of agents where, on the one hand, the distribution and correlation of the agents' attributes are similar to those in the census, and, on the other hand, the number of agents with a specific property matches the aggregate data.

Several case studies successfully employed *synthetic reconstruction* based on the original work by Beckman *et al.* (1996). Bowman (2004) presents a good overview of the techniques available in 2004. The TRANSIMS population synthesizer is described in Hobeika (2005). Frick and Axhausen (2004) present a synthesizer for the Swiss population.

Another approach to population synthesis is to employ *combinatorial optimization* techniques, as shown by Voas and Williamson (2000). This approach is compared to synthetic reconstruction in Ryan *et al.* (2009); Huang and Williamson (2001).

Recently, issues like the control for agent relationships, categorization detail, and memory requirements have been scrutinized in the literature (Auld *et al.*, 2010; Pritchard and Miller, 2009; Ye *et al.*, 2009; Srinivasan and Ma, 2009; Arentze *et al.*, 2007; Guo and Bhat, 2007). The purpose of this paper is to analyze above selection of state of the art population synthesizers, to describe what they have in common and to analyze and evaluate their individual characteristics. We hope that our review will be helpful for the implementation of future population synthesis routines.

This paper reviews synthesizer procedures that have been published since 2007 that the authors are aware of. We focus on synthetic reconstruction procedures for person/household populations only; approaches using combinatorial optimization or synthesizing other kinds of relationships are not reviewed. The accuracy of the synthesized population is outside the scope of this paper: Every synthesis routine works with different kinds of input data, and different measures of goodness-of-fit are used for validation of the particular synthesizers.

The remainder of this paper is organized as follows. First, we introduce the reviewed synthesizers. In the two subsequent sections we describe in detail the two primary steps each synthesis procedure consists of, highlighting similarities and differences. We conclude with a summary.

2 Scope of the review

This section introduces the various population synthesis procedures reviewed. For each synthesizer we briefly summarize the focus of the corresponding publications. Each synthesis procedure is given a name, typeset in bold, that we will use subsequently. For the two standalone synthesizers, this is the name of the synthesizer itself; for the others, the name denotes the model that the synthesizer has been used for primarily.

PopSynWin: Auld *et al.* (2008) present a population synthesizer called PopSynWin, further improvements are described in Auld *et al.* (2010). It has been used for synthesizing the population of Chicago, Illinois, but can be applied to other study areas and input data as well. The publications focus on automatic adjustment of categorization detail, and on adjusting household selection probabilities to control for person-level constraints.

ILUTE: The ILUTE model (Salvini and Miller, 2005) uses a population synthesizer developed by Pritchard (2008), with improvements shown in Pritchard and Miller (2009). The simulation area of ILUTE is Toronto, Canada. The authors employ a new approach for handling large attributes spaces during execution of the IPF procedure: The contingency table is represented as a sparse list structure with one entry per unique combination of attributes. In addition, the authors propose a solution for generating person/household relationships when such are not defined by the input data.

PopGen: This synthesizer, presented by Ye *et al.* (2009), is another standalone software package. Its primary application was Maricopa County, Arizona, but it can be used for any other area in the United States out of the box. A novel technique is used to simultaneously fit for household and person marginals.

FSUMTS: For the FSUMTS model (Srinivasan and Ma, 2009; Srinivasan *et al.*, 2008), the population of Tampa Bay and South-East Florida is synthesized. Instead of using a probabilistic procedure for choosing desired households, the authors propose a ranking procedure that uniquely determines the household to be selected next. Also, the authors have verified their results very thoroughly using a back-casting approach.

CEMDAP: The work by Guo and Bhat (2007) presents a synthesizer implemented on the basis of Beckman *et al.* (1996) for the CEMDAP model (Pinjari *et al.*, 2006) of the Dallas–Fort Worth area, Texas. The authors introduce a method for combining multiway contingency tables in a generic fashion.

ALBATROSS: The populaton synthesizer for the Albatross model, presented by Arentze *et al.* (2007), is an example of a synthesizer for a European region. The household-level distribution is computed from the person-level distribution in a preprocessing step.

3 Fitting

Bowman (2009) presents a succinct description of population synthesis, according to which all population synthesis procedures have two basic stages in common. We call these stages *fitting* and *allocation*. The fitting stage computes an aggregate representation of the target population for the base year; disaggregation is performed in the allocation stage. In all synthesizers reviewed, both stages are present. The following two sections describe these stages. For each stage, we provide a general description and show approach-specific peculiarities.

3.1 Description

The purpose of the fitting stage is to fit a disaggregate sample of agents (called *reference sample*) to aggregated constraints (referred to as *marginal sums*, *marginals*, or *control totals*). For person/household synthesis, the reference sample contains demographic data for a representative subsample of the population, potentially having spatial information removed, and/or taken at a point in time different to the base year of the simulation. The control totals are given for a selection of attributes, the *controlled variables*, present in the reference sample. For each attribute, the desired number (or proportion) of agents per category is given. The joint distribution of controlled variables in the sample is referred to as *seed*.

Typically, the reference sample is created from a census, a microcensus, or from public-use micro sample data. We assume that the demographic distribution is the same in the sample and in our target population. The marginal sums are obtained from readily available aggregate data for the base year.

Marginals can be single- or multi-dimensional. An example of a three-dimensional marginal is the joint distribution of age, sex, and education in the population. Note that, in general, a two-dimensional marginal for variables from the domains M and N can be treated as a single-dimensional marginal of a variable from the Cartesian product $M \times N$. Higher-dimensional marginals can be converted analogously. Hence, in the remainder of this paper, we consider single-dimensional marginals only.

The *Iterative Proportional Fitting* procedure (IPF) estimates a distribution of control variables with the following two properties: (a) the number of agents in a given category matches the corresponding marginal sum, and (b) the correlation structure of the seed is retained. A multi-dimensional *contingency table* or *cross tabulation* is initialized with the seed and the marginals. Then, all control variables are iterated in a round-robin fashion: For each category of the current control variable, the corresponding slice of the contingency table is scaled proportionally so that the total number of agents matches the control total. Each such iteration is referred to as *IPF step*. The loop is terminated, as soon as the relative error of the distribution vs. the marginal sums reaches a user-specified threshold.

IPF has been first described by Deming and Stephan (1940) and is also known as *matrix raking*, *RAS method*, or *matrix scaling*. The theory behind IPF is well understood, cf. Csiszár (1975); Fienberg (1970); Mosteller (1968); Ireland and Kullback (1968); Stephan (1942). Important features of IPF are minimization of relative entropy and preservation of cross-product ratios; in other words, among all contingency tables that satisfy the marginal constraints, the resulting table is the most similar one to the initial table.

Only recently, Pukelsheim and Simeone (2009) presented a proof of convergence that allows to determine in advance whether a given contingency table converges under IPF. In practice, for the application of population synthesis, convergence problems only occur if entire rows or columns are zero, and the corresponding marginal is nonzero. This is discussed in Section 3.4.

In its basic formulation, IPF can estimate only one level of aggregation, i.e., it can control either for agent-level or for group-level attributes but not for both simultaneously. Sometimes it suffices to convert all agent-level attributes into group-level attributes; in this case, IPF can be used on the group-level distribution (Arentze *et al.*, 2007). For simultaneous fitting of more than one level of aggregation, one would have to resort to another algorithm for fitting.

With the exception of PopGen, all of the synthesizers reviewed use IPF for the fitting stage. We review the structure of the marginals and the spatial resolution for the different synthesis procedures. Subsequently, we describe various refinements found in the literature. There are quite a few propositions on how to handle the so-called “zero-cell problem” that may occur with real-world data. Also, two modifications to the original formulation of IPF deserve attention: The sparse list data structure introduced by Pritchard and Miller (2009), and the automatic category reduction described by Auld *et al.* (2008). Finally, we show two approaches to simultaneously control for both agent- and group-level attributes: an algorithm similar to IPF developed by Ye *et al.* (2009), and a formulation as an IPF problem on a special structure as shown by Arentze *et al.* (2007).

3.2 Control dimensions

As described before, multi-dimensional control totals can be easily converted into single-dimensional control totals with more categories. Most of the reviewed population synthesizers use multi-dimensional marginals. The exceptions are PopGen and ALBATROSS: The special data structures used here do not provide straightforward support for attributes with a great many categories.

3.3 Zoning

Most synthesizers work with a spatial hierarchy: A *region* contains several *zones*, and one or many regions form the study area. Marginals are usually provided at zone level, but the reference sample is given region-wise. Consequently, IPF is first run at regional level using marginals aggregated for the region, and the result becomes the seed for further IPF runs that computes the disaggregate zone-wise population.

The ILUTE synthesizer implements two approaches to zone-level synthesis. The most straightforward way is to run many smaller IPF runs, one per zone; this is referred to as the *zone-by-zone* approach. In contrast, the *multizone* approach synthesizes all zones simultaneously by extending each control total with a zone dimension. It turns out that, with the multizone approach, the fit improves slightly (Pritchard and Miller, 2009).

However, multizone synthesis also requires more memory. The ILUTE synthesizer requires one additional floating-point value per agent per zone. This is mainly due the sparse-list structure described in a forthcoming subsection; with classical IPF, storage requirements multiply by the number of zones. – All other synthesizers use the simpler zone-by-zone approach.

3.4 Zero-cell problem

As already noted by Beckman *et al.* (1996), especially when dealing with small geographies, it is possible to have a non-zero marginal for a category that has no representative in the reference sample. In this case, eventually a division by zero occurs during the execution of IPF, and the outcome of the algorithm is undefined. This is referred to as the *zero-cell problem* in the literature. The simplest solution to the problem is to initialize the false zero cells with an arbitrarily small value. This assures convergence, however, a bias may be introduced. For this reason, other solutions were sought after.

PopSynWin reduces the occurrence of zero cells using a category reduction routine; we further describe this approach in Section 3.6. Guo and Bhat (2007) also suggest to perform a category reduction as a preprocessing step; however, this has not been automated in the CEMDAP synthesizer.

PopGen replaces false zero cells in the zone-level seed with an estimate computed from the region-level seed and the number of agents in that zone. Then, it performs a simple linear fit to account for the discrepancy introduced by incrementing cells' values. For details, we refer the reader to Ye *et al.* (2009). Similarly, the FSUMTS synthesizer "borrows" from another area to fill false zero cells.

3.5 Sparse list

A major problem of many previous synthesizers were the memory requirements for the contingency table: With every controlled variable, another dimension is added to the table. The contingency table grows exponentially with the number of attributes; its size equals the product of the category counts of all attributes. A large contingency table is inherently sparse: The number of nonzero values is at most as large as the size of the reference sample. This calls for a more efficient storage scheme of the contingency table.

Williamson *et al.* (1998) recommend a list-based representation in the context of population synthesis. As the reference sample is usually given as a list of attributes, it can be used without further treatment. The ILUTE synthesizer implements a variant of IPF that operates directly on the list of attributes by attaching a real-valued *expansion factor* to each item. Every operation of classical IPF can be translated into a change of the expansion factor, allowing the algorithm to work entirely on the list-based representation. A detailed description can be found in (Pritchard, 2008, section 4.2.1).

For large attribute spaces and detailed categorization schemes, memory consumption is greatly reduced. The memory required by the list-based representation is only proportional to the size of the reference sample and the number of attributes and does not depend on categorization detail. For the categorization scheme used by the ILUTE model, the sparse list representation cuts down memory requirements to 0.2 % for the zone-by-zone and to 0.07 % for the multizone approach (Pritchard and Miller, 2009).

The following two convenient properties of list-based IPF are noteworthy as well. First, all attributes present in the reference sample are preserved, not only the ones that are controlled for. Second, the method natively supports simultaneous fitting for marginals with different categorization detail.

3.6 Category reduction

Among the reviewed population synthesizers, only the one used by the ILUTE model supports arbitrarily detailed categorization of attributes. This is attributed to the list-based IPF. All other procedures implement classical IPF, and hence need to reduce categorization detail and/or number of control variables in order to keep memory consumption at a reasonable level.

PopSynWin has an option to automatically aggregate categories for interval-scale attributes. For this, the user specifies a percentage threshold that is applied to the marginals: All categories whose marginal does not exceed this threshold are merged with a neighboring category. Apart from decreasing the number of categories, this procedure also reduces the occurrence of false zero cells in the seed (cf. Section 3.4).

As PopSynWin runs IPF on a region-by-region basis, different categorization schemes are applied for each region: For example, the income attribute is recategorized differently in wealthier and poorer regions. – According to Auld *et al.* (2008), category reduction potentially worsens the quality of the synthesized population. While it is a feasible remedy against the zero-cell problem, using a sparse list as shown in the previous subsection seems to be a better solution to the memory consumption problem.

3.7 Iterative Proportional Updating

Of all the synthesis procedures reviewed, only PopGen uses a fitting procedure different to IPF. The new approach, named *Iterative Proportional Updating* (IPU), simultaneously controls for multiple hierarchy levels (agents and agent groups) during the fitting procedure. The proposed algorithm has many parallels to the sparse list variant of IPF. The core data structure of IPU is a tabular list of agent groups. In this list, a count column exists for each category of each group-level or agent-level attribute. Columns for agent-level categories contain the number of agents (persons) in a group that belong to the corresponding category. The value of a column for a group-level category equals one if and only if the given group belongs to this category, and zero otherwise. An additional weight column is initialized with ones (or with group weights taken from the reference sample). After that, each category is repeatedly considered in a round-robin fashion. For each category, the groups with nonzero counts in the corresponding count columns are reweighted in a fashion quite similar to sparse-list IPF. The procedure is continued until convergence is reached. As a result, the weights match both agent-level and group-level constraints.

According to Ye *et al.* (2009), IPU performs well in practice. The authors also provide a geometric explanation of the algorithm, however, a theoretical proof of convergence is missing.

Fitting against multidimensional marginals would require one column per combination of controlled categories. The memory requirements increase exponentially with the dimensionality of the marginal. Similarly to the sparse list approach for classical IPF, a list of lists can be used to solve this problem. Each element of the main list of groups would then contain a list of agents with their full attribute set.

3.8 Relation matrix

ALBATROSS uses the concept of a *relation matrix* to estimate a composition of households that perfectly matches person-level constraints. The relation matrix is a specially formed contingency table, consisting of two rows and two columns in the simplest case. In what follows, we provide an example for a 2x2 relation matrix that allows to estimate the distribution of single and male-female households for a human population. The row marginals define the total number of females in the first column and the total number of males that live independently in the second column. Conversely, the column marginals control for the amount of males and single females. The interior cells of the table contain household counts: The top left cell represents the number of couples, while the bottom left and top right cells denote the number of single-person male or female households, respectively. The bottom-right cell is a zero by definition: A household with an independently living male and an independently living female is a contradiction in itself.

Our simple example can be extended by splitting the rows and columns. Arentze *et al.* (2007) add another row and another column, each of which controls for persons like children that live in another household and are not head of that household. Two distinct relation matrices are created by disaggregating the first row and first column by age and work status, respectively. As with classical IPF, the matrices are initialized with data from the reference sample. After performing IPF on these relation matrices, the resulting distributions are used as marginals for a classical IPF run controlling for household attributes.

While the relation matrices allow to compute a household-level distribution from a person-level one, it remains unclear how to apply this method for other hierarchies like employees/firms or tenants/buildings.

4 Allocation

As shown in the previous section, the fitting stage computes an aggregated representation of the target population. Disaggregation is performed in the allocation stage by solving the following tasks (Bowman, 2009):

- The joint distribution estimated by IPF is adjusted to integers.
- Households are selected from the reference sample according to the fitted distribution. The full set of variables required by the model system is retained.
- Sometimes, the geographic placement of the households is refined.

In this work, we focus on the selection task.

To the best of our knowledge, no theoretical results have been reported for the allocation stage. Especially the integerization and the selection tasks may introduce a bias in the synthesized population. As as consequence, a synthetic population should be validated carefully by statistically comparing the estimated joint distribution and the seed (Voas and Williamson, 2001).

4.1 Description

In the allocation stage, a disaggregate set of agents and agent groups with attributes required by the microsimulation model is computed. The result of the fitting procedure is a real-valued *group weight* for each feasible combination of group categories; all of the synthesizers reviewed use these weights to select concrete groups into the synthetic population.

Repeated probabilistic selection with replacement is the most common strategy: Groups are drawn with a probability proportional to the group weight. If fitting does not control for agent-level attributes, the allocation procedure can also prefer groups that best fit the agent-level marginals. After the allocation procedure, each member of the synthetic population has clearly defined attributes, the full set of agents match the predefined control totals, and the interactions present in the reference sample are, to a great extent, preserved for the synthetic agents.

In this section, we review alterations to the selection procedure proposed by Auld *et al.* (2010) and Pritchard and Miller (2009), and a greedy deterministic selection procedure presented by Srinivasan *et al.* (2008).

4.2 Altered selection probability

PopSynWin uses a sophisticated formula for computing group selection probabilities. This formula favors groups with agents of categories still being underrepresented in the population so far. As a result, after completion of the selection procedure, agent-level marginals are matched approximately. Experiments confirm absolute differences of 2 % on average and at most 7 % for rare categories for a person/household population (Auld *et al.*, 2010).

A drawback of the improved selection probability is that it needs to be recomputed for every group after each selection. Performing the selection naïvely would result in run times depending at least quadratically on the number of groups. Owing to that, PopSynWin implements a heuristics that iterates over a random shuffle of the groups and recomputes the selection probability only for the current group (Auld *et al.*, 2010).

4.3 Conditional Monte Carlo

The reference sample available for the ILUTE model is unique in the sense that it does not contain links between households and persons. Household- and person-level distributions are estimated independently; they are subsequently fitted against each other to ensure consistency. It is only in the allocation stage that persons are assigned to households.

Households are repeatedly selected according to their selection probabilities. Each member of the current household is then drawn from the subset of eligible persons according to the persons' selection probabilities. This ensures consistency with both household structure and person-level marginals.

The synthesis of relationships can be carried out as described above even if the agent/group relationships are present in the input data. According to Pritchard and Miller (2009), one should consider whether variation in group composition outweighs the drawback of purely synthetic agent/group relationships. The authors report only a slight worsening in goodness-of-fit when synthesizing links.

4.4 Deterministic selection

The FSUMTS synthesizer is the only one among the reviewed procedures that does not rely on probabilistic selection. Instead, a per-household *fitness value* is used as deterministic choice criterion. The fitness value is a measure for the adherence to both household- and person-level constraints, given the already selected households. After computing fitness values for all households, the household with the largest fitness value is selected. This process is repeated until all fitness values fall below zero.

While this approach guarantees repeatability and control for both agent and group levels (Srinivasan *et al.*, 2008), the fitness values potentially change after each selection. This results in run times depending quadratically on the number of synthesized groups if implemented naïvely. However, since the fitness values are nonincreasing with the execution of the algorithm, it should be possible to drastically reduce the amount of computation required for each iteration by maintaining a list of groups sorted by fitness value. After selection, the recomputation of the fitness value for the top group in the list may result in moving this group away from the top. By repeating this until the top group stays on top, it is guaranteed that the group with the largest fitness value is found. This group is selected, and fitness values are readjusted again, etc. – Unfortunately, Srinivasan *et al.* (2008) do not provide running times or implementation details.

5 Summary

We have reviewed six population synthesis procedures used for various microsimulation models. Each synthesizer is used for a different region and requires different input data. While each synthesizer has its own advantages, a superior synthesizer that incorporates all favorable characteristics of recent approaches and allows to compare and validate them for the same input data, is yet to be developed. Given the difficulties that routinely arise when trying to properly create a synthetic population, it seems worthwhile to invest time to develop a generic software solution. The software should be applicable to different kinds of input data – concerning both geographic contexts and agent types – without code level changes. Due to the diversity of the input and output data, however, it is likely that a single standalone program will not be able to provide a solution for each possible application. Instead, an extendable open-source software framework that offers routines for tasks that frequently arise in population synthesis applications could be the method of choice.

6 Acknowledgements

The authors would like to thank Nadine Schüssler for careful revision and invaluable comments and suggestions.

References

Arentze, T. A., H. J. P. Timmermans and F. Hofman (2007) Population synthesis for microsimulating travel behavior, *Transportation Research Record*, **2014** (11) 85–91.

Auld, J., A. K. Mohammadian and K. Wies (2008) Population synthesis with control category optimization, paper presented at the *the 10th International Conference on Application of Advanced Technologies in Transportation*, Athens, Greece, May 2008.

Auld, J., A. K. Mohammadian and K. Wies (2010) An efficient methodology for generating synthetic populations with multiple control levels, paper presented at the *the 89th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2010.

Beckman, R. J., K. A. Baggerly and M. D. McKay (1996) Creating synthetic baseline populations, *Transportation Research Part A: Policy and Practice*, **30** (6) 415–429.

Bowman, J. L. (2004) A comparison of population synthesizers used in microsimulation models of activity and travel demand, `http://jbowman.net/papers/2004.Bowman.Comparison_of_PopSyns.pdf`, accessed on 29/07/2010.

Bowman, J. L. (2009) Population synthesizers, *Traffic Engineering and Control*, **49** (9) 342.

Csiszár, I. (1975) I-divergence geometry of probability distributions and minimization problems, *Annals of Probability*, **3**, 146–158.

Deming, W. E. and F. F. Stephan (1940) On the least squares adjustment of a sampled frequency table when the expected marginal totals are known, *Annals of Mathemtical Statistics*, **11** (4) 427–444.

Fienberg, S. (1970) An iterative procedure for estimation in contingency tables, *Annals of Mathemtical Statistics*, **41**, 907–917.

Frick, M. and K. W. Axhausen (2004) Generating synthetic populations using IPF and Monte Carlo techniques: Some new results, paper presented at the *4th Swiss Transport Research Conference*, Ascona, March 2004.

Guo, J. Y. and C. R. Bhat (2007) Population synthesis for microsimulating travel behavior, *Transportation Research Record*, **2014** (12) 92–101.

Hobeika, A. (2005) Transims fundamentals: Chapter 3: Population synthesizer, *Technical Report*, U.S. Department of Transportation, Washington, D.C. `http://tmip.fhwa.dot.gov/resources/clearinghouse/docs/transims_fundamentals/ch3.pdf`. Accessed on 29/07/2010.

Huang, Z. and P. Williamson (2001) Comparison of synthetic reconstruction and combinatorial optimisation approaches to the creation of small-area microdata, *Working Paper*, **2001/2**, Department of Geography, University of Liverpool, UK.

Ireland, C. T. and S. Kullback (1968) Contingency tables with given marginals, *Biometrika*, **55**, 179–188.

MATSim-T (2010) Multi Agent Transportation Simulation Toolkit, webpage, `http://www.matsim.org`. Accessed on 29/07/2010.

Mosteller, F. (1968) Association and estimation in contingency tables, *Journal of the American Statistical Association*, **63**, 1–28.

Pinjari, A. R., N. Eluru, R. B. Copperman, I. N. Sener, J. Y. Guo, S. Srinivasan and C. R. Bhat (2006) Activity-based travel-demand analysis for metropolitan areas in Texas: CEMDAP models, framework, software architecture and application results, *Research Report*, **4080–8**, Texas Department of Transportation, Department of Civil, Architectural and Environmental Engineering, University of Texas Austin, Austin, October 2006.

Pritchard, D. R. (2008) Synthesizing agents and relationships for land use / transportation modelling, Master Thesis, University of Toronto, Toronto.

Pritchard, D. R. and E. J. Miller (2009) Advances in agent population synthesis and application in an integrated land use and transportation model, paper presented at the *the 88th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2009.

Pukelsheim, F. and B. Simeone (2009) On the iterative proportional fitting procedure: Structure of accumulation points and L1-error analysis, `http://opus.bibliothek.uni-augsburg.de/volltexte/2009/1368/pdf/mpreprint_09_005.pdf`, accessed on 29/07/2010.

Ryan, J., H. Maoh and P. Kanaroglou (2009) Population synthesis: Comparing the major techniques using a small, complete population of firms, *Geographical Analysis*, **41** (2) 181–203.

Salvini, P. A. and E. J. Miller (2005) ILUTE: An operational prototype of a comprehensive microsimulation model of urban systems, *Networks and Spatial Economics*, **5** (2) 217–234.

Srinivasan, S. and L. Ma (2009) Synthetic population generation: A heuristic data-fitting approach and validations, paper presented at the *the 12th International Conference on Travel Behaviour Research (IATBR)*, Jaipur, December 2009.

Srinivasan, S., L. Ma and K. Yathindra (2008) Procedure for forecasting household characteristics for input to travel-demand models, *Final Report*, **TRC-FDOT-64011-2008**, Transportation Research Center, University of Florida. `http://www.fsutmsonline.`

net/images/uploads/reports/FDOT_BD545_79_rpt.pdf. Accessed on 29/07/2010.

Stephan, F. (1942) An iterative method of adjusting sample frequency tables when expected marginal totals are known, *Annals of Mathemtical Statistics*, **13**, 166–178.

UrbanSim (2010) Open Platform for Urban Simulation, webpage, http://www.urbansim.org. Accessed on 29/07/2010.

Voas, D. and P. Williamson (2000) An evaluation of the combinatorial optimisation approach to the creation of synthetic microdata, *International Journal of Population Geography*, **6**, 349–366.

Voas, D. and P. Williamson (2001) Evaluating Goodness-of-Fit Measures for Synthetic Microdata, *Geographical and Environmental Modelling*, **5** (2) 177–200.

Williamson, P., M. Birkin and P. H. Rees (1998) The estimation of population microdata by using data from small area statistics and samples of anonymised records, *Environment and Planning A*, **30** (5) 785–816.

Ye, X., K. Konduri, R. M. Pendyala, B. Sana and P. Waddell (2009) A methodology to match distributions of both household and person attributes in the generation of synthetic populations, paper presented at the *the 88th Annual Meeting of the Transportation Research Board*, Washington, D.C., January 2009.