MULTI-LEVEL POPULATION SYNTHESIS USING ENTROPY MAXIMIZATION BASED SIMULTANEOUS LIST BALANCING

Binny Mathew Paul, Corresponding Author
RSG, Inc.
1515 SW 5th Avenue, Suite 1030, Portland, OR 97201
Tel: 503-200-6603 Email: binny.mathewpaul@rsginc.com

Jeff Doyle
RSG, Inc.
55 Railroad Row, White River Junction, VT 05001
Tel: 802-359-6465 Email: jeff.doyle@rsginc.com

Ben Stabler
RSG, Inc.
1515 SW 5th Avenue, Suite 1030, Portland, OR 97201
Tel: 503-200-6601 Email: ben.stabler@rsginc.com

Joel Freedman
RSG, Inc.
1515 SW 5th Avenue, Suite 1030, Portland, OR 97201
Tel: 503-200-6602 Email: joel.freedman@rsginc.com

Alex Bettinardi
ODOT
555 13th Street NE, Suite 2, Salem, OR 97301
Tel: 503-986-4104 Email: alexander.o.bettinardi@odot.state.or.us

Paper size: 5,264 words + 5 tables (×250) + 3 figures (×250) = 7,264 words

Submitted for presentation at the 97th Annual Meeting of the Transportation Research Board and Publication in the Transportation Research Records

November 15th, 2017

ABSTRACT

Synthetic population generation is the first step to an Activity Based Model (ABM). Most population synthesizers are limited when it comes to multi-level data, modifying an existing synthetic population and avoiding algorithmic errors. Monte Carlo variance resulting from drawing discrete households and persons from a probability distribution is a common source of error. In algorithms where zones are processed sequentially, errors can propagate through the list of zones resulting in large errors for the last zone processed. To the extent that these errors are higher for smaller population segments, they can adversely impact the accuracy of forecasts dependent upon these markets; for example, transit ridership estimates may be inaccurate due to errors in the location of university students.

This paper presents an entropy maximization based population synthesizer (PopulationSim) which handles multiple geographies and avoids algorithmic errors. It is implemented as part of Oregon Department of Transportation's (ODOT) effort to develop an open source population synthesis platform. PopulationSim has been implemented in the Python-based ActivitySim framework, an open-source collaborative framework for model development. PopulationSim uses a simultaneous list balancer and a Linear Programming based simultaneous integerizer to eliminate error due to the sequential processing of zones.

A working version of PopulationSim was implemented for a test case and compared to a widely-used population synthesizer. PopulationSim eliminates the errors due to sequential processing of zones and results in a reasonable match to controls. Besides these major algorithmic enhancements, PopulationSim includes provision to specify flexible number of geographies and options to modify an existing synthetic population.

Keywords: Activity-based model, population synthesizer, population synthesis, list balancing, entropy maximization

INTRODUCTION & MOTIVATION

In the past few years, travel demand forecasting has witnessed tremendous growth in the development of activity-based models (ABM). There are many fully-functional ABMs in practice, some examples are CT-RAMP (1), DaySim (2), CEMDAP (3) and FAMOS (4). ABMs predict activity and travel choices of persons and households considering space and time constraints as well as individual characteristics. ABMs operate in a micro-simulation framework, wherein the choices of each decision-making agents are predicted by applying Monte Carlo methods to behavioral models. This requires person and household level attributes of the entire population in the modeling region. Moreover, designing a forecasting scenario requires a process to synthesize a population fitting the scenario's demographic assumptions. Disaggregate population samples can be obtained from sources like American Community Survey (ACS) Public Use Microdata Sample (PUMS) or a household travel survey. Marginal distributions of person and household-level attributes of interest are also available from Census. The challenge is to generate a synthetic population using the population sample and marginal distributions by applying a data fusion technique. This population sample is commonly referred to as the *seed or reference sample* and the marginal distributions are referred to as *controls or targets*. The process of expanding the seed sample to match the marginal distribution is termed *population synthesis*. With the advancement of ABMs in recent years, synthetic population generation has also received research attention. Most of the population synthesis methods in literature involve two steps – first, a fitting step and then a generation step. At the end of the fitting step, an expansion factor is assigned to each record in the seed sample. The generation step involves expanding the sample using Monte Carlo drawing, bucket rounding or an optimization-based method.

The most common fitting technique used by various population synthesizers is the Iterative Proportional Fitting (IPF) procedure (5). Beckman et al. (6) used the IPF procedure to obtain joint distributions of demographic variables and random sampling from PUMS to generate baseline synthetic population. One of the limitations of the IPF method is that it does not incorporate person level attributes while generating the joint distributions. Many studies have refined this method to incorporate both household and person level attributes (7, 8, 9, 10, 11). Ye et al. (12) proposed a heuristic algorithm called the Iterative Proportional Updating Algorithm (IPU) to incorporate both person and household-level variables in the fitting procedure. Besides IPF, entropy maximization algorithms have been used as a fitting technique (13, 14, 15). In most of the entropy based methods, the relative entropy is used as the objective function. The relative entropy based optimization ensures that the least amount of new information is introduced in finding a feasible solution. The base entropy is defined by the initial weights in the seed sample. The weights generated by the entropy maximization algorithm preserves the distribution of initial weights while matching the marginal controls. This is an advantage of the entropy maximization based procedures over the IPF based procedures. The other major group is the combinatorial optimization based techniques (16, 17, 18, 19). Besides these, optimization and simulation based methods have been developed (20, 21, 22). While much research has been done towards finding innovative solutions to incorporate controls at multiple levels, little has been done to include controls at multiple geographic levels. Vovsha et al. (15) presented an entropy maximization-based algorithm (PopSynIII, developed for Maricopa Association of Governments) which permits specification of controls at multiple geographic levels. Konduri et al. (23) presented an extension of the IPU algorithm which allows for specification of controls at multiple geographies.

PopSynIII operates across three geographic levels and uses a Linear Programming (LP) based generation procedure. However, PopSynIII traverses sequentially through geographies while allocating population from an upper to lower geography (e.g., PUMS to Traffic Analysis Zones). This sequential processing at each geographic level results in a big error for the last zone to be processed. These errors can be significant for minority population segments like university students, low income households, and so on, whose travel behavior can be very different from the general population. For example, university students tend to be a small segment of the overall regional population but are the majority users of transit services (24, 25, 26, 27). Therefore, it is critical for a population synthesizer to accurately predict the residential location of specific travel markets in a modeling region.

Another potential application of the population synthesis tool is to generate a synthetic population for traffic impact studies. In this type of application, a synthetic population must be generated for small geographies (only one or a few zones) without perturbing the synthetic population for the rest of the region. Population synthesizers with sequential zone processing makes it difficult to generate a synthetic population for such applications. Finally, all population synthesizers that the authors are aware of prescribe a certain number of geographies (two in the case of PopGen2 (23) and three in the case of PopSynIII (15)). For many applications (statewide population synthesis, future forecasts, etc.), analysts may wish to specify control data at fewer or more geographic levels.

This paper describes a new population synthesizer, PopulationSim. It has been implemented as part of Oregon Department of Transportation's (ODOT) on-going effort to develop a standardized population synthesis tool in the ActivitySim (28) framework, an open-source collaborative framework for model development. PopulationSim is based upon the maximum-entropy list-based approach developed as part of PopSynIII. However, PopulationSim eliminates sequential processing of zones, allows for specification of controls at any number of geographies, and generates synthetic populations for small geographies.

The next section presents the PopSynIII algorithm, PopulationSim enhancements and PopulationSim implementation. After this, the test environment and results are described and finally, conclusions and directions for future work are listed.

POPSYNIII ALGORITHM

PopSynIII is built around a list balancing procedure which is applied at three geographic levels. The balancing procedure maintains correspondence between different geographic levels by following an aggregation approach for the controls where in the controls from the lower geographies are aggregated to the upper geographies. The list balancing implementation follows an allocation approach where in the results from an upper geography are distributed or allocated to a lower geography within the upper geography. Two basic components of PopSynIII algorithm can be grouped as *list balancing* and *integerizing*.

List Balancing

Input to this procedure is a list of household records with an initial weight attached to each record, typically obtained from Census Public Use Microdata Sample (PUMS). The objective is to find weights that match the given marginal control distributions. Table 1 presents a tabular representation of the list balancing problem. Where, W_n are the initial weights of each household record and X_n are the final weights which satisfy the marginal controls (A_i).

TABLE 1 List Balancing Example

HH ID	HH size				Person age				Initial weights	weights	Example Final weights
	1	2	3	4+	0-15	16-35	36-64	65+			
	i=1	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	W_n	X_n	
n=1	1							1	20	x_1	250
n=2		1			1	1			20	x_2	250
n=3			1			1	2		20	x_3	250
n=4				1		2	2		20	x_4	150
n=5				1	1	3	2		20	x_5	150
Control(A_i)	250	250	250	300	400	1250	1100	250			

There can be multiple solutions (x_n) to this problem. However, an ideal solution would match the distribution of the initial weights as closely as possible. This is achieved by formulating the list balancing procedure as a convex entropy maximization problem. The mathematical formulation of list balancing as an entropy maximization problem is presented below:

$$\min_{\{x_n\}} \sum_n x_n \ln \frac{x_n}{w_n},$$

Subject to constraints:

$$\sum_n a_{in} \times x_n = A_i, (\alpha_i),$$
$$x_n \geq 0 \ ,$$

Where, α_i represents dual variables that give rise to the balancing factors. a_{in} are the incidence table value relating each record to controls. The solution for the above problem is as follows:

$$x_n = k \times w_n \times exp(\sum_i a_{in}\, \alpha_i) = w_n \times \prod_i [exp(\alpha_i)]^{a_{in}} = w_n \times \prod_i (\hat{\alpha}_i)^{a_{in}},$$

where $\hat{\alpha}_i$ represents balancing factors that must be calculated iteratively.

This basic formulation is modified to allow for control relaxations, setting relative importance of controls and upper and lower bounds on weight variables. These modifications do not change the basic form of the optimization problem and this convex optimization with linear constraints is solved iteratively using the Newton-Raphson method (15).

Integerizing

The final output from the list balancing procedure are fractional weights corresponding to each household record in the seed sample. Simple rounding techniques or Monte Carlo drawing can introduce significant errors, especially when summed across large number of geographies. PopSynIII uses an LP based procedure to covert fractional weights to integers. First, the integer part of the fractional weights is separated and the residual controls are computed as:

$$\underline{A}_i = A_i - \sum_n a_{in} \times int\ (x_n)$$

The residual weights range from 0 to 1 and can be assumed to be binary (0 or 1). With these, a maximum entropy problem can be formulated for binary weights (y_n) as follows:

$$\min \sum_n y_n \times \begin{bmatrix} \ln\left(\frac{y_n}{x_n}\right), & if \quad y_n = 1 \\ 0 & if \quad y_n = 0 \end{bmatrix} \Rightarrow \max \sum_n y_n \times \ln x_n ,$$

Subject to constraints:

$$\sum_n a_{in} \times y_n = \underline{A}_i ,$$

$$y_n = 0,1$$

Here too, slack variables or relaxation factors can be introduced to handle cases when no solution exists (15).

Algorithm

PopSynIII algorithm starts at the "seed" level (usually Public Use Microdata Areas) and implements list balancing for each seed geography independently. A simple methodology is used to apply regional, or meta, controls to each seed geography, wherein the final seed weights from the initial seed balancing are used to compute the seed-level values of meta controls as suggested by the initial seed weights. A final seed balancing is implemented adding the factored meta controls. The final step involves allocation of households from the seed geography to mid and low-level geographies. The allocation is implemented sequentially from smallest to biggest lower geography based on number of households in each zone. The last zone is not processed whilst the remaining upper geography weights are allocated to the last zone.

POPULATIONSIM ENHANCEMENTS

While PopSynIII algorithm is very efficient and incorporates various innovative features, there are some issues which can lead to errors in the generated synthetic population. The list balancer needs to be applied once for each zone when multiple zones are nested below an upper level zone. Allocation is implemented for one sub-geography at a time without replacement. Household records available for allocation to each subsequent zone are updated by subtracting the allocated records to the current zone from the list of records at the upper geography. Thus, the last zone to be processed is allocated all the remaining household records in the upper geography without any optimization. Since households are not allocated to the last zone via list balancing, it is probable that the controls are not matched for this zone. For a small zone, the resulting error can be significant. To avoid this, mid and low zones are sorted in an increasing order of total number of households and processed in the same order. This assumes that bigger zones can easily absorb the resulting error in control matching. This however may not always work as desired, especially for controls relating to a minority segment of the population (e.g., university students), where small errors accumulate over zones, resulting in a large error in the last zone processed. This problem in the context of this paper will be referred to as the "large-zone-error-problem".

Traditionally, travel demand models have ignored such minority segments like university students, but their activity and trip making behavior can be quite different from the general

population (24, 25, 26, 27). Even though these university students represent a minority segment of the population, they are one of the majority users of transit services. Therefore, it is critical to accurately predict such minority groups in the synthetic population for successfully evaluating various policies aimed at improving transit ridership, etc. It is also important to allocate the generated minority groups to the right zone. For example, university students tend to live close to the transit corridors for easy access to transit lines. Therefore, it is very important that a population synthesizer not just generate accurate number of minority individuals but also allocate them to the right location in the modeling region.

The key to solving the large-zone-error-problem is to replace sequential list balancing with a simultaneous list balancer whose objective function is to calculate weights for each household record which match the marginal controls for each zone simultaneously. A tabular representation of this problem is presented in Table 2. Where, W_n are the final weights assigned to the household records at the upper geography. W_{1n}, W_{2n} and W_{3n} are the weight variables for each household record corresponding to each zone. A_{1i}, A_{2i}, A_{3i} are the marginal control totals for the three zones in this example.

TABLE 2 Simultaneous List Balancing

HH ID	HH size				Person age				Final weight	Zone 1 weight	Zone 2 weight	Zone 3 weight
	1	2	3	4+	0-15	16-35	36-64	65+				
	i=1	*i=2*	*i=3*	*i=4*	*i=5*	*i=6*	*i=7*	*i=8*	W_n	W_{1n}	W_{2n}	W_{3n}
n=1	1							1	250	w_{11}	w_{21}	w_{31}
n=2		1			1	1			250	w_{12}	w_{22}	w_{32}
n=3			1			1	2		250	w_{13}	w_{23}	w_{33}
n=4				1		2	2		150	w_{14}	w_{24}	w_{34}
n=5				1	1	3	2		150	w_{15}	w_{25}	w_{35}
Control(A_i)	250	250	250	300	400	1250	1100	250				
Control(A_{1i})	50	50	50	60	80	250	220	50				
Control(A_{2i})	75	75	75	90	120	375	330	75				
Control(A_{3i})	125	125	125	150	200	625	550	125				

The simultaneous list balancer formulates a convex entropy maximization problem for each zone which is solved using the iterative Newton-Raphson method. The upper limit on W_{1n}, W_{2n} and W_{3n} is set to W_n, the final assigned weights at the upper geography. The Newton-Raphson iterations are completed for all zones simultaneously. This allows for shifting weights across zones to achieve a system optimal solution as the iterations progresses. Having a weight variable for each household record-zone combination results in much higher degrees of freedom for this optimization problem. This higher flexibility gives every zone a fair access to some hard to find household records, for e.g., university students.

The simultaneous list balancer does not attempt to resolve the inconsistencies in the control data but tries to minimize the error per zone. The higher flexibility in simultaneous balancing allows for errors resulting from inconsistent controls to be distributed across all zones instead of being concentrated among few zones.

In PopSynIII, the upper limit on weights is strictly enforced during sequential list balancing. This ensures that the final weights at the lower geographies would not violate the matching of controls at the upper geography. This is not applicable in case of simultaneous list balancing and the sum of final weights for a given household record across all zones may exceed or fall below the final weight assigned to that record at the upper geography. Therefore, to enforce this constraint, the weights are scaled at the end of each iteration to match the weights at the upper geography. Mathematically, this scaling can be expressed as:

$$W_n = W_{1n} + W_{2n} + W_{3n}$$

The final output from the simultaneous list balancing procedure is a list of weights (x_{zn}) for each household record (n) corresponding to each zone (z). The next step is to discretize these floating-point weights. Discretization is implemented at the end of the simultaneous balancing based allocation and can be sequential or simultaneous. Under sequential discretization, an LP problem is formulated for each zone as described earlier. In case of simultaneous discretization, a single LP problem is formulated to convert decimal portions of the final weights for all household records across all zones in each geographic level. First, residual controls are computed for each zone in a geographical level as:

$$\underline{A}_{zi} = A_{zi} - \sum_n a_{in} \times int\,(x_{zn})$$

The residual weights can again be assumed binary and must be computed for each household record (n) and zone (z) combination. With these, a maximum entropy problem can be formulated for binary weights (y_{zn}) as follows:

$$\min \sum_z \sum_n y_{zn} \times \begin{bmatrix} \ln\left(\frac{y_{zn}}{x_{zn}}\right), & if \quad y_{zn} = 1 \\ 0 & if \quad y_{zn} = 0 \end{bmatrix} \Rightarrow \max \sum_z \sum_n y_{zn} \times \ln x_{zn} ,$$

Subject to constraints:

$$\sum_n a_{in} \times y_{zn} = \underline{A}_{zi},$$

$$y_{zn} = 0,1$$

These constraints will match the zonal controls but might perturb the upper geography control distribution. This is handled by imputing the upper geography controls for current geographic level using the balanced weights. The imputed upper geography controls (j) are expressed as:

$$\sum_n a_{jn} \times y_{zn} = \underline{A}_{zj}$$

Where, $\underline{A}_{zj} = \sum_n a_{jn} \times (x_{zn}) - \sum_n a_{jn} \times int\,(x_{zn})$

This LP problem is solved using a standard LP solver. The current version of PopulationSim uses CVXPY (28) for simultaneous integerization and can use CVXPY or ORTOOLS (29) for sequential integerization.

POPULATIONSIM IMPLEMENTATION

PopulationSim has been implemented in the open-source ActivitySim (30) framework, an ABM software platform sponsored by a consortium of transportation planning agencies. As illustrated below, the framework is quite flexible and is being used by the Federal Highway Administration (FHWA) and the Oregon Metro benefit cost analysis toolkit (31). The framework is implemented in Python and makes heavy use of the Pandas and Numpy libraries (32,33), which allow for vectorization of operations to reduce overall runtime.

Controls for a PopulationSim run are specified via a Comma Separated Values (CSV) expression file of Python expressions. User can specify Pandas and Numpy expressions in the expressions file to operate on the input data tables. This makes the control specification very intuitive and flexible. Table 3 shows a portion of an expressions file. The seed table can be households or persons and if persons, then the expressions calculator counts the persons fitting the expression to households.

TABLE 3 Example Control Expression File

Description	Geography	Seed Table	Importance	Control Field	Expression
HH Size 1	MAZ	households	5000	HHSIZE1	NP == 1
HH Size 2	MAZ	households	5000	HHSIZE2	NP == 2
HH Size 3+	MAZ	households	5000	HHSIZE3	NP >= 3
OSU students in family	TAZ	Persons	10000	OSUfam	OSUTag==1
OSU students in non-family	TAZ	Persons	10000	OSUnfam	OSUTag==0
Workers in occupation category 1	County	Persons	100	OCCP1	occp==1

Models implemented in the framework are run by a data pipeliner that reads the list of model steps and executes the steps in order. All tables are stored in an intermediate Hierarchical Data Format (HDF5) binary file that is used for data input and output throughout the model. This allows for restarting of a PopulationSim run from any point. The final output from a PopulationSim run include the final synthetic population (expanded household and persons file), HDF5 data pipeline, a household-level weights summary from each step in the algorithm and a final control vs synthesized totals comparison.

PopulationSim framework was developed keeping in mind some of the desired use-cases in travel demand modeling. Traffic impact study is one such use-case where the user would want to update or add to the synthetic population of a subset of zones. Another desired feature is flexible number of geographies for control specification. The current working version of PopulationSim includes both features.

RESULTS & VALIDATION

To evaluate PopulationSim's performance, PopSynIII has been used as the base case. To make a fair comparison, both PopSynIII and PopulationSim were run on the same input dataset. The

following sub-sections describe the test environment, the used validation procedures and finally, the test results.

Test Environment

The Corvallis-Albany-Lebanon Modeling (CALM) region in Oregon, USA was selected as a test region to validate PopulationSim against PopSynIII. The CALM region consists of a single PUMA, which becomes both the seed and the meta geography for this region. The 35 Census Tracts form the mid geography and 930 TAZs form the lower geography, resulting in three geographies for specifying controls. The CALM modeling region represents 62,041 households and 156,452 persons. The CALM modeling region also houses the Oregon State University (OSU). Total university student population for the CALM region is about 17,510 students. The 5% ACS PUMS data from 2007-2011 has been used as the seed sample. Table 4 presents the common set of controls across the two population synthesizers. Both PopulationSim and PopSynIII were run on the same machine with two 3.00 GHz processors and 160 Giga Bytes of installed Random Access Memory (RAM). PopSynIII runs in about 12 minutes and 30 seconds while PopulationSim takes about 16 minutes and 30 seconds to run with sequential integerization and around 40 minutes with simultaneous integerization. It should be noted that the current version of PopulationSim used for this test has not been optimized to use parallel processing to bring down the run time.

TABLE 4 Control Specification

Control	Values	Importance	Geography
Total number of households		Very high	TAZ
Household size	1, 2, 3, 4+	Med	TAZ
Age of householder	15-24, 25-54, 55-64, 65+	Low	TAZ
Household income quartiles		Med	TAZ
OSU students by housing type	Family/non-family	High	TAZ
Number of workers	0, 1, 2, 3+	Med	Tract
Housing type	SF, MF, MH, Duplex	Med	Tract
Number of workers by occupation	8 occupation groups	Med	Region

Validation Statistics

For each control, percentage difference between synthesized totals and control totals are computed at the geography at which the control was specified. A validation chart is created for both software which is a visualization of the disaggregate summary statistics – mean percentage difference and standard deviation (STDEV) of percentage differences. A form of dot and whisker plot is generated for each control where the dots are the mean percentage differences and horizontal bars are twice the STDEV centered around zero. Since the test environment has only one meta geography, STDEV is not computed for meta controls. For all other controls (TAZ and Tract level), means and STDEV of percentage differences are computed across the geographies at which the control was specified.

Test Results

PopulationSim was run with both sequential and simultaneous integerization. Figure 1 presents the validation summary from the sequential integerization run. The TAZ-level total households control is matched perfectly. Most of the other TAZ-level controls are matched within reasonable ranges except for the OSU students control, which has a slightly higher relative variance. Tract level housing type controls is matched perfectly with almost no variance. There is a discrepancy between the number of workers specified at the regional level and the number of workers implied by the distribution of workers by workers per household specified at the Tract level. The meta level occupation controls were scaled down for the PopulationSim test runs.

The sequential integerization procedure can match the Tract level controls while essentially over-riding the meta level controls matched by the simultaneous list balancer. This is avoided by imputing the upper geography controls using the balanced household weights and use those as additional constraints in discretization. The sequential discretization may still perform inefficiently for minority population segment. This was demonstrated by poor performance of the regional occupation type 7 (Military) control which applies to a minority population segment. Simultaneous integerization performs better for controls on minority population segments and leads to closer match of controls at lower geographies. Figure 2 presents the results from the PopulationSim run with simultaneous integerization.

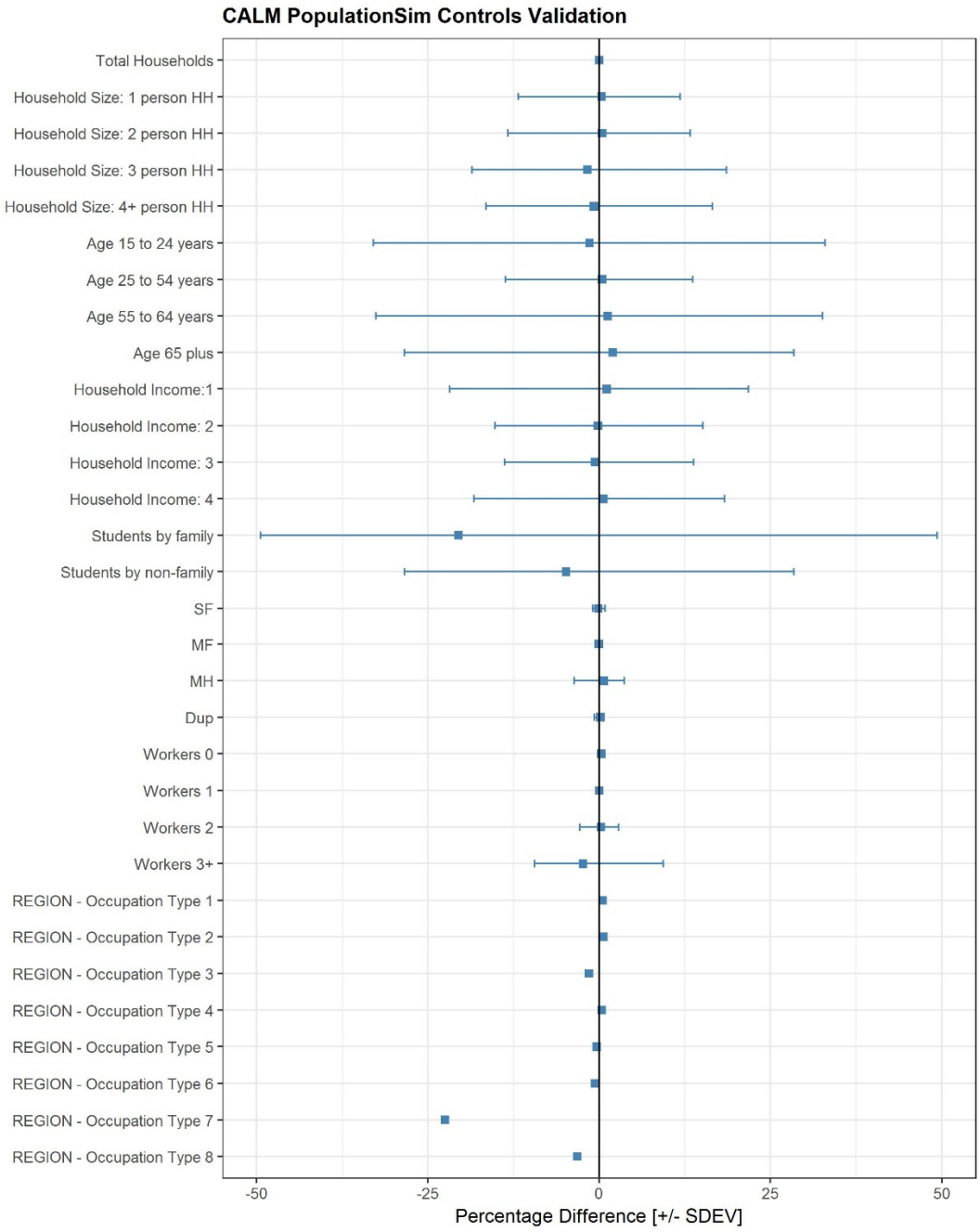


FIGURE 1 PopulationSim Validation – Sequential Integerization

It can be observed that simultaneous integerization leads to an improvement in the match to meta-level occupation type 7 control. Simultaneous integerization improves the performance across most controls. Figure 3 presents the comparison of PopulationSim and PopSynIII validation. The PopulationSim run is with scaled meta controls. Some controls perform better for PopulationSim especially the ones which performed poorly for PopSynIII (population aged 15 to 24 years). PopulationSim has higher variability on students control. This might be a result of simultaneous balancing spreading the error across multiple zones instead of concentrating it over few zones.

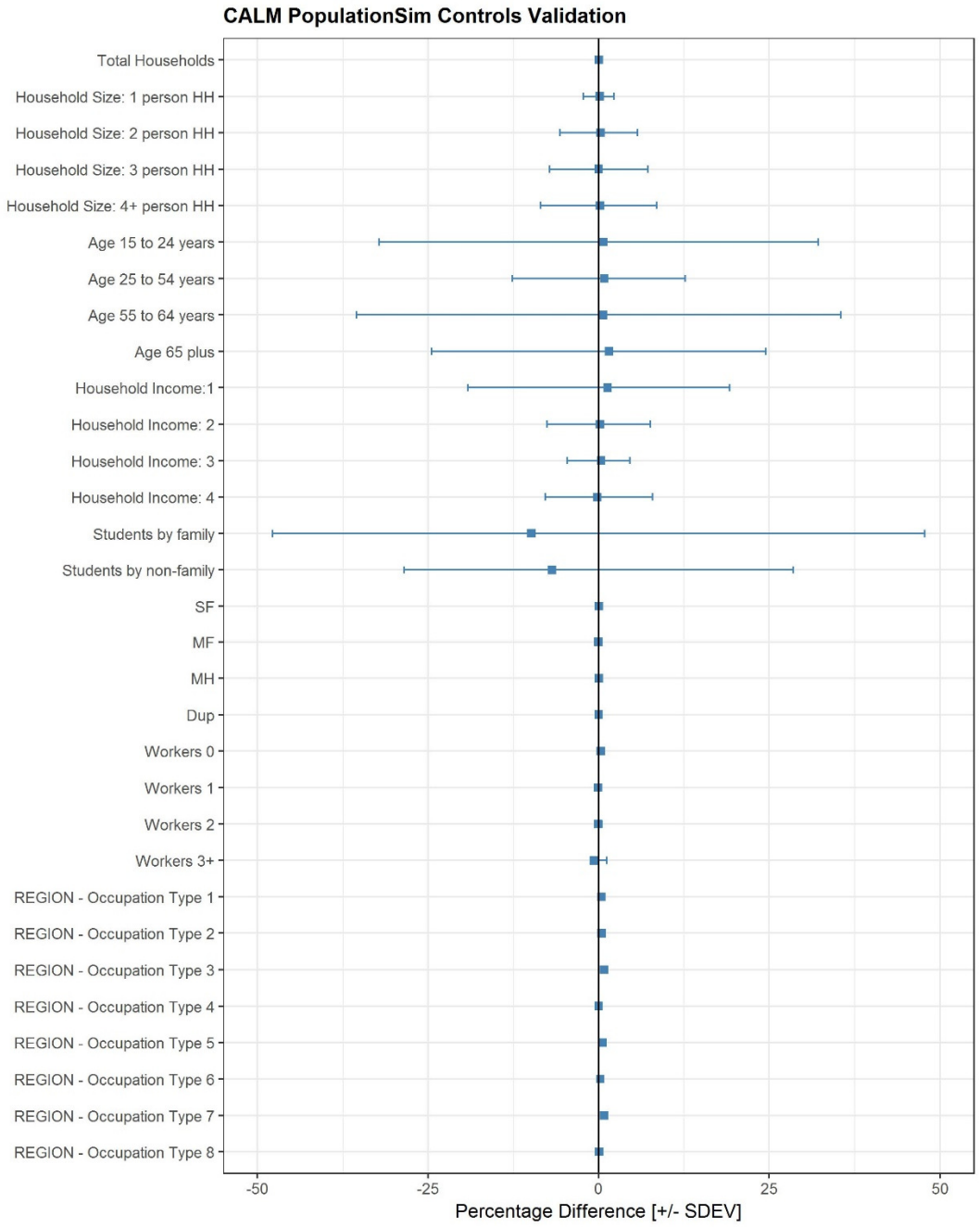
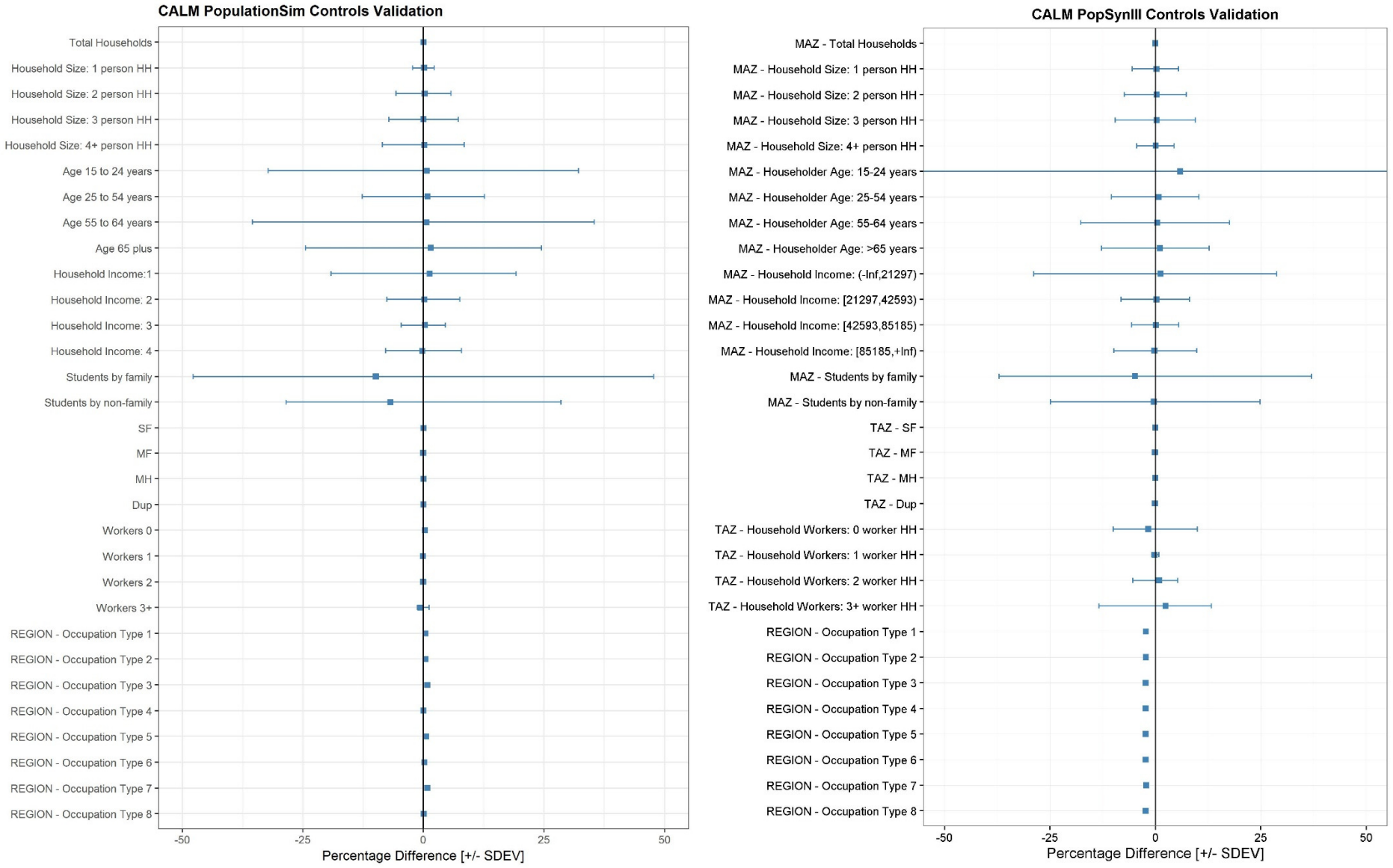


FIGURE 2 PopulationSim Validation – Simultaneous Integerization



2 **FIGURE 3 PopulationSim (Simultaneous Integerization) vs. PopsynIII**

Fixing the Large Zone Error Problem

One of the major objectives of the PopulationSim algorithm is to solve the large zone error problem resulting from sequential list balancing, which is more evident with respect to minority population segments. One such control in the current test environment is Oregon State University (OSU) students by family type control. Table 5 shows the performance of this control for the biggest TAZs in each Tract in the CALM region for PopulationSim and PopSynIII (the TAZs that are processed last in the PopSynIII algorithm). It can be observed that the PopulationSim algorithm resolves the large zone error problem for all the zones shown.

TABLE 5 Large Zone Error [OSU Student Control]

Zone Number (TAZ)	Control Value	(Control – Predicted)	
		PopSynIII	PopulationSim
OSU Students in Family Households			
302	69	11	0
356	74	9	-1
654	4	6	0
563	13	6	-2
241	18	6	1
OSU Students in Non-Family Households			
121	250	-13	2
722	236	13	4
337	0	12	0
822	219	11	-1
302	312	-11	-1

CONCLUSION & FUTURE WORK

Generating a synthetic population is the first step in microsimulation-based disaggregated ABMs. The accuracy of a synthetic population in representing the true population can have a significant impact on the quality of forecasts from ABM and the resulting policy analysis. This becomes even more critical when the analysis is aimed at studying a smaller market in the modeling region (e.g., university students, low-income households, etc.). Most existing population synthesizers are limited when it comes to accurately predicting smaller markets within a general population. Algorithmic errors can add up across geographies resulting in inaccurate prediction of smaller markets of interest. Monte Carlo variance resulting from drawing discrete households and persons from a probability distribution is a common source of error. In algorithms, where zones are processed sequentially, errors can propagate through the list of zones resulting in large errors for the last zone processed.

This work presented an entropy maximization based population synthesizer called PopulationSim. PopulationSim has been implemented in the Python-based ActivitySim framework and employs a simultaneous list balancer. A simultaneous list balancer avoids propagation of errors to the last zone which is the case with sequential list balancing of zones. A

Linear Programming based simultaneous integerizer in PopulationSim averts random noise from Monte Carlo drawing. PopulationSim was tested for the CALM region and the results were compared to PopSynIII implementation for the same region. Results show that PopulationSim eliminates the large zone error observed in PopSynIII and performs reasonably well in terms of match to controls at various geographies. Besides these major algorithmic enhancements, PopulationSim includes various desirable features. Once such feature is provision for specifying a flexible number of geographies. Certain applications like traffic impact studies require synthetic population to be generated for a subset of zones. PopulationSim allows the user to update the synthetic population for a subset of zones.

Some other desirable features under development include an inputs pre-processor which exposes all the input tables to the user via expression files. These expressions files follow the ActivitySim framework format and allow the user to write Python expressions to create additional required data fields without changing the source code. The inputs pre-processor will perform standard consistency checks for all the inputs at the start of the run. A tracer is also under development which will enable the user to specify one zone at each desire geographic level to output trace / debug results. Another feature under development is the ability to share seed sample across geographies. The motivation for this feature is that as regions grow and change, PUMAs from other regions may be more appropriate sources of households than the PUMS sample from the last Census for the region. The next release version would also output aggregate multiway distributions which will be an input to aggregate demand models.

REFERENCES

1. Vovsha, P., J. Freedman, V. Livshits, and W. Sun. Design Features of Activity-Based Models in Practice: Coordinated Travel-Regional Activity Modeling Platform. *Transportation Research Record: Journal of the Transportation Research Board*, *No. 2254*, 2011, pp. 19-27.
2. Bradley, M., J.L. Bowman, and B. Griesenbeck. SACSIM: An Applied Activity-Based Model System with Fine Level Spatial and Temporal Resolution. *Journal of Choice Modeling*, Vol. 3, No. 1, 2010, pp. 5-31
3. Bhat, C.R., J. Y. Guo, S. Srinivasan, and A. Sivakumar. Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns. *Transportation Research Record: Journal of the Transportation Research Board*, *No. 1894*, 2004, pp. 57-66.
4. 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 the Transportation Research Board*, *No. 1921*, 2005, pp. 123-130.
5. 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*, Vol. 11, 1940, pp. 427-444.
6. Beckman, R.J., K.A. Baggerly, and M.D. McKay. Creating Synthetic Baseline Populations. *Transportation Research Part A: Policy and Practice*, Vol. 30, No. 6, 1996, pp. 415-429.
7. Guo, J. Y., and C.R. Bhat. Population Synthesis for Microsimulating Travel Behavior. *Transportation Research Record: Journal of the Transportation Research Board*, *No. 2014*, 2007, pp. 92-101.
8. 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*, *No. 2014*, 2007, pp. 85-91.
9. Pritchard, D. R., and E. J. Miller. Advances in Agent Population Synthesis and Application in an Integrated Land Use and Transportation Model. Presented at 88th Annual Meeting of Transportation Research Board, Washington, D.C., 2009.
10. Auld, J., and A. Mohammadian. Efficient Methodology for Generating Synthetic Populations with Multiple Control Levels. *Transportation Research Record: Journal of the Transportation Research Board*, *No. 2175*, 2010, pp. 138-147.
11. Müller, K., and K. W. Axhausen. Hierarchical IPF: Generating a Synthetic Population for Switzerland. Eidgenössische Technische Hochschule Zürich, IVT, 2011.
12. Ye, X., K. Konduri, R. M. Pendyala, B. Sana, and P. Waddell. A Methodology to Match Distributions of Both Household and Person Attributes in the Generation of Synthetic Populations. Presented at 88th Annual Meeting of the Transportation Research Board, Washington, D.C., 2009
13. Bar-Gera, H., K. Konduri, B. Sana, X. Ye, and R. M. Pendyala. Estimating Survey Weights with Multiple Constraints Using Entropy Optimization Methods. Presented at 88th Annual Meeting of the Transportation Research Board, Washington, D.C., 2009.
14. Lee, D. H., and Y. Fu. Cross-Entropy Optimization Model for Population Synthesis in Activity-based Microsimulation Models. *Transportation Research Record: Journal of the Transportation Research Board*, *No. 2255*, 2011, pp. 20-27.

15. Vovsha, P., J. E. Hicks, B. M. Paul, V. Livshits, P. Maneva, and, K. Jeon. New Features of Population Synthesis. Presented at 94th Annual Meeting of the Transportation Research Board, Washington, D.C., 2015.
16. Williamson, P., M. Birkin and P. H. Rees. The estimation of population microdata by using data from small area statistics and samples of anonymised records. *Environment and Planning A*, *30 (5),* 1998, pp. 785–816.
17. Voas, D. and P. Williamson. An evaluation of the combinatorial optimization approach to the creation of synthetic microdata. *International Journal of Population Geography, 6*, 2000, pp. 349–366.
18. Ryan, J., H. Maoh and P. S. Kanaroglou. Population synthesis: Comparing the major techniques using a Small, Kenneth A., complete population of firms, *Geographical Analysis,* Vol 42, No. 2, 2009, pp. 181–203.
19. Abraham, J. E., K. J. Stefan, and J. D. Hunt. Population Synthesis Using Combinatorial Optimization at Multiple Levels. Presented at the 91st Annual Meeting of Transportation Research Board, Washington, D.C., 2012.
20. Barthelemy, J. and P. L. Toint. Synthetic Population Generation Without a Sample, *Transportation Science,* Vol 47, No. 2, 2013, pp. 266-279.
21. Ma, L., and S. Srinivasan. Synthetic Population Generation with Multilevel Controls: A Fitness-Based Synthesis Approach and Validations. *Computer-Aided Civil and Infrastructure Engineering*, Vol. 30, No. 2, 2015, pp. 135-150.
22. Farooq, B., K. Müller, M. Bierlaire and K. W. Axhausen (2015) Methodologies for synthesizing populations, in R. Hurtubia, M. Bierlaire, P. A. Waddell and A. de Palma (eds.) *Integrated transport and land use modeling for sustainable cities*, 77–94, EPFL Press, Lausanne.
23. Konduri, K.C., D. You, V. M. Garikapati, and R. M. Pendyala. Enhanced Synthetic Population Generator that Accommodates Control Variables at Multiple Geographic Resolutions. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2563, 2016, pp. 40-50. https://doi.org/10.3141/2563-08
24. Eom, J. K., Stone, J. R., and Ghosh S. K. Daily Activity Patterns of University Students. *Journal of Urban Planning and Development,* Vol. 135, No. 4, 2009.
25. Khattak, X. Wang, and S. Son. Travel by University Students in Virginia: Is this Travel Different from Travel by the General Population? *Transportation Research Record: Journal of the Transportation Research Board,* Vol. 2255, 2011, pp. 137–145.
26. Limanond, T., Butsingkorn, T., and Chermkhunthod, C. Travel Behavior of University Students Who Live on Campus: A case Study of a Rural University in Asia. *Transport Policy,* Vol. 18, No. 1, 2011, pp. 163–171. https://doi.org/10.1016/j.tranpol.2010.07.006.
27. Paul, B. M., P. Vovsha, J. Hicks, V. Livshits, R. Pendyala. Extension of Activity-Based Modeling Approach to Incorporate Supply Side of Activities: Examples for Major Universities and Special Events. *Transportation Research Record: Journal of the Transportation Research Board,* Vol. 2429, 2014, pp. 138–147.
28. *ORTOOLS*, https://github.com/google/or-tools. Accessed November 14, 2017.
29. *CVXPY,* http://www.cvxpy.org/en/latest/ Accessed November 14, 2017.
30. *ActivitySim*, https://github.com/udst/activitysim. Accessed July 28, 2017.
31. Benefits calculator, https://github.com/RSGInc/bca4abm. Accessed July 28, 2017.
32. Numpy, http://www.numpy.org. Accessed July 28, 2017.
33. Pandas, http://pandas.pydata.org. Accessed July 28, 2017.