On Iterative Proportional Updating: Limitations and Improvements for General Population Synthesis

Peijun Ye, *Member, IEEE*, Bin Tian, Yisheng Lv, *Senior Member, IEEE*, Qijie Li, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—Population synthesis is the foundation of the agent-based social simulation. Current approaches mostly consider basic population and households, rather than other social organizations. This article starts with a theoretical analysis of the iterative proportional updating (IPU) algorithm, a representative method in this field, and then gives an extension to consider more social organization types. The IPU method, for the first time, proves to be unable to converge to an optimal population distribution that simultaneously satisfies the constraints from individual and household levels. It is further improved to a bilevel optimization, which can solve such a problem and include more than one type of social organization. Numerical simulations, as well as population synthesis using actual Chinese nationwide census data, support our theoretical conclusions and indicate that our proposed bilevel optimization can both synthesize more social organization types and get more accurate results.**

***Index Terms*—Agent-based simulation, bilevel optimization, iterative proportional updating (IPU), population synthesis.**

I. Introduction

Multiagent system (MAS) is a significant paradigm to model distributed systems. This technique is usually applied to analyze the engineering systems where disaggregate components are coordinately controlled [1], [2], and the social/ecological systems where the systemic dynamics have emerged through a bottom-up way [3]. For the latter field, two stages are typically required to reconstruct or predict the system's evolution. First, a synthetic population that represents the real individuals within the studied region is generated [4]. Classic methods involve synthetic reconstruction [5], combinatory optimization [6], Markov chain Monte Carlo (MCMC) simulation-based method [7], etc. Second, computational agent-based models are established on the synthetic population to investigate the micro/macro dynamics. Popular research topics in recent years involve group decision making [8], [9]; consensus achieving [10]–[13]; social cognition [14], [15]; and cooperative or noncooperative gaming [16]–[18]; to name a few. Obviously, the synthetic population provides a reasonable and reliable initial state to keep a plausible systemic evolution. Thus, it is fundamental to multiagent social computing.

In population synthesis, households and individuals are usually both considered. This is partly because the existing population synthesis methods usually use census data as their main input. A part from individual attributes, most census statistics also provide partial household information, which facilitates the synthesis of individual and household as a whole. For example, Zhao *et al.* [19] generated a synthetic population for disaggregate traffic demand analysis. Gargiulo *et al.* [20] presented an iterative method to generate statistically realistic population of households. Auld and Mohammadian [21] developed another technique to determine how both household- and person-level characteristics can jointly be used as controls when synthesizing populations. They introduced the Bayesian method to assemble the constraints from two levels dynamically. Barthelemy and Toint [22] proposed a sample-free household selection method based on entropy maximization and tabu search, and they adopted it to Belgian and Australian synthetic population generation. Huet *et al.* [23] used this assignment approach to create households in French municipalities, and to study dynamics of labor status and job changes. Ma and Srinivasan [24] proposed a fitness-based approach in 2015. They reweight each household sample record by measuring the fitness of the total household constraints. Most recently, Huynh *et al.* [25] used population assignment to generate the synthetic population of New South Wales. They assigned the basic population according to the household-individual mapping relations. Anderson and Farooq [26] modeled the population assignment as a k partite-graph problem. After compared

Manuscript received January 4, 2020; accepted April 16, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61603381, Grant 61503380, Grant 61533019, and Grant U1811463, in part by the Center of National Railway Intelligent Transportation System Engineering and Technology under Contract RITS2019KF03, in part by the China Academy of Railway Sciences, in part by the Chinese Guangdong's S&T Project under Grant 2019B1515120030, and in part by the Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles ("ICRI-IACV"). This article was recommended by Associate Editor J. Liu. *(Corresponding author: Yisheng Lv.)*

Peijun Ye, Bin Tian, and Yisheng Lv are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yisheng.lv@ia.ac.cn).

Qijie Li is with the with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence and Automation, Beijing University of Technology, Beijing 100124, China.

Fei-Yue Wang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the Parallel Intelligence Research Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.2991427

TABLE I
EXAMPLE OF THE IPU ALGORITHM

HH ID	Wei	HHT 1	HHT 2	IndT 1	IndT 2	IndT 3	Wei 1	Wei 2	Wei 3	Wei 4	Wei 5	FWei
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						
FWei Sum		35.00	65.00	91.00	65.00	104.00						

[1] HH: HouseHold; HHT: HouseHold Type; Ind: Individual; IndT: Individual Type; Wei: Weight; FWei: Final Weight.

the mainstream methods, Ye *et al.* [27]–[29] proposed joint distribution inference and copula-based approaches to synthesize the population. Sun *et al.* [30] gave a hierarchical mixture model for household-individual population synthesis in their Singapore application.

Basically, the related methods can be categorized into two types: 1) population assignment and 2) distribution fitting. Population assignment starts by generating the household and individual entity pools, respectively, and then assembles them according to a heuristic search. It provides an intuitive model of how a family is formed in reality. However, since the two entity pools are usually inconsistent, the assignment may probably terminate due to one of them exhausted while the other is still not empty. In contrast, the distribution fitting fits the joint distribution of both individual and household attributes. This, in essence, deals with the inconsistency during the fitting phase and achieves a joint distribution that contains matched households and individuals. Its primary merit is that the final accuracy can be minimized before entities generated. Currently, only a few methods belong to this type where iterative proportional updating (IPU) is a representative one [31]. The authors of the IPU paper provide a 2-D case that illustrates how the algorithm works. But unfortunately, it lacks a solid validation, which elicits the further analysis of this article. The main contributions of this article are two-fold: 1) we theoretically prove that the IPU algorithm does not converge to an optimal population in general. Such a theoretical analysis, to the best of our knowledge, is given for the first time and 2) the problem that IPU solves is extended to a more general case in which not only households but also other social organizations are taken into account. A bilevel optimization is proposed to compute such a general population. To support our conclusions, the methods are tested and evaluated using both stochastic numerical simulation and real Chinese national census data. Results indicate that our proposed method can not only include more social organization types but also achieve better population distributions.

The remainder of this article is organized as follows. Section II presents a brief review of the IPU method with an example provided by the original paper. Section III investigates the general mathematical model that IPU solves. We will give two propositions in this section to prove this algorithm does not converge to a claimed optimal solution. Section IV extends the problem to a bilevel optimization and introduces gradient descent to solve the problem. To test and support our theoretical conclusions, Section V conducts computational experiments using both pure numerical random inputs and real Chinese census data. Finally, in Section VI, this article concludes with some additional discussions.

II. PROBLEM STATEMENT AND ALGORITHM REVIEW

As a sample-based method, the IPU algorithm was developed to fit the weights of different types of households, so that households and individuals can simultaneously satisfy the statistical marginals in both levels [31]. To make this article self-contained, the simple example from the original literature is cited here to illustrate its main thought. Consider a target population that contains two household types and three individual types. Here, a household/individual type is defined by a particular combination of its attributes. For instance, (*Number of Members* = 3, *Type of Residence* = urban) and (*Number of Members* = 3, *Type of Residence* = rural) are two different household types. Table I gives the compositions of each household. Household type 1 contains three possible cases indicated by the top three rows. The first row means that in this composition, the household consists of one person of each individual types 1–3. The compositions of each household shown in the table actually depict the mapping relations from household to persons. They are determined from the input sample. In this example, there are eight households with 23 individuals. All initial household weights are set to be one arbitrarily. The "weighted sum" row represents the sum of each column weighted by the "weights" column. The "constraints" row provides the marginal distributions of household and individual types that must be matched. These marginal distributions come from statistical results published officially, each of which provides an entity or individual number with a part but not all of the attributes. δ_a and δ_b rows calculate 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. During the computation, the IPU algorithm adjusts weights for each household/person constraint in an iterative fashion. As in the table, 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. It is $35/3 = 11.67$. The weights for all households of household type 1 are multiplied by this ratio to satisfy the constraint. Thus, the weights for all households of household-type 1 become equal to 11.67, and the weighted sum for this type will be equal to the corresponding constraint. Similarly, the weights for households of household type 2 are adjusted by an amount equal to $65/5 = 13.00$. Note that after this update, the weights match both two household-level constraints. When fitting the first individual-level constraint, the adjustment is calculated as the ratio of the constraint for individual type 1 to the weight sum of the individual type 1 column. This ratio is equal to $91/111.67 = 0.81$. This value is used to update the weights of all households that have persons of individual 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 the "weights 3" column. The constraint corresponding to individual type 1 is now perfectly matched. The process is repeated for the remaining two types of constraints and the corresponding updated weights are shown in the columns titled "weights 4" and "weights 5." The corresponding weighted sums are shown in the various rows titled "weighted sum." Ye *et al.* [31] argued in his paper that the iterative updates would minimize the following objective functions:

$$\sum_j \left[\frac{\sum_i d(i,j)\cdot\omega_i - c_j}{c_j}\right]^2 \text{ or } \sum_j \frac{\left[\sum_i d(i,j)\cdot\omega_i - c_j\right]^2}{c_j} \text{ or } \sum_j \frac{\left|\sum_i d(i,j)\cdot\omega_i - c_j\right|}{c_j} \tag{1}$$

where i stands for the household ($i = 1,\ldots,8$); j stands for the constraint ($j = 1,\ldots,5$); $d(i,j)$ is the frequency of household or person of type j in household i; w_i is the weight of the ith household; and c_j is the value of constraint j.

III. Theoretical Analysis of the IPU Algorithm

Clearly, the IPU algorithm runs two iterative proportional fitting (IPF) procedures to fit household and individual constraints, respectively [32]. In a general case, however, it may probably encounter fatal problems: the algorithm cannot converge to an optimal solution. Before giving a proof, we first construct the mathematical model that IPU operates. Table II shows the general mathematical model of IPU. Let A_1 and A_2 be the frequency matrices of household and individual calculated by the disaggregate samples ($R_m = R$)

$$A_1 = \begin{bmatrix} d(1,1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d(R_m, m) \end{bmatrix}$$

$$A_2 = \begin{bmatrix} d(1,m+1) & \cdots & d(1,m+n) \\ \vdots & \ddots & \vdots \\ d(R,m+1) & \cdots & d(R,m+n) \end{bmatrix}.$$

Let $H(x_i)(i = 1,\ldots,m)$ and $P(y_j)(j = m+1,\ldots,m+n)$ be the constraints of household and individual, and $h = (h_1, h_2, \ldots, h_R)^T$ be the relevant weights. The subscription stands for the cell of h. During the t-th ($t \geq 1$) iteration, the weights are updated through the household fitting and individual fitting

$$h_i^{(1)}(t) = \frac{h_i^{(m+n)}(t-1)}{\sigma_i^{(1)}(t)},\quad h_i^{(2)}(t) = \frac{h_i^{(1)}(t)}{\sigma_i^{(2)}(t)},\ldots$$
$$h_i^{(m+n)}(t) = \frac{h_i^{(m+n-1)}(t)}{\sigma_i^{(m+n)}(t)} \tag{2}$$

where $h_i^{(j)}(t)$ ($j = 1,\ldots,(m+n)$) represents the weights after the jth constraint fitted. Note that only $h_i^{(1)}(t)$ uses the h value from the $(t-1)$th iteration. The other $h_i^{(j)}(t)$ ($j > 1$) use the previous h value in the t-th iteration. In (2) $\forall i \in [1,R]$, the update factors are

$$\sigma_i^{(j)}(t) = \begin{cases} \frac{\sum_r d(r,j)\cdot h_r^{(j-1)}(t)}{H(x_j)}, & d(i,j) > 0 \\ 1, & d(i,j) = 0 \end{cases} \quad j \in [1,m]$$
$$\sigma_i^{(j)}(t) = \begin{cases} \frac{\sum_r d(r,j)\cdot h_r^{(j-1)}(t)}{P(y_j)}, & d(i,j) > 0 \\ 1, & d(i,j) = 0 \end{cases} \quad j \in [m+1,m+n] \tag{3}$$

where $h_i^{(0)}(t) = h_i^{(m+n)}(t-1)$. The subscript r is a subscript of summation, meaning that the summary operation is for all rows. Here, we use this notation to avoid the confusion with the subscript i in $\sigma_i^{(j)}(t)$.

In order to investigate the convergence, we need to consider the objective functions in (1). Here, we choose the third indicator

$$L = \sum_j \frac{\left|\sum_i d(i,j)\cdot h_i - c_j\right|}{c_j} \tag{4}$$

to compute. But please note the related conclusions can be applied to the other two analogically. Our discussions start from two different cases according to the feature of A_2.

Case 1: A_2 does not contain zero elements. It means that for a given i, $\sigma_i^{(m+1)}(t),\ldots,\sigma_i^{(m+n)}(t)$ are all determined by the weights. The following proposition reveals the trend of the objective function.

Proposition 1: Suppose the IPU algorithm starts from any feasible initial solution and updates the weights according to (2). $\forall d(i,j) > 0$ in A_2. Then

$$\prod_{k=1}^{m+n} \sigma_i^{(k)}(t) = 1$$

holds for $t > 1$ $\forall i \in 1,2,\ldots,R$.

TABLE II
GENERAL CASE OF IPU

HH Type 1	HH Type 2	$\cdots$	HH Type m	Ind Type $(m+1)$	$\cdots$	Ind Type $(m+n)$	Weights
$d(1,1)=1$	0		0	$d(1,m+1)$		$d(1,m+n)$	h_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$
$d(R_1,1)=1$	0		$\vdots$	$\vdots$		$\vdots$	$\vdots$
0	$d(R_1+1,2)=1$		$\vdots$	$\vdots$		$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
$\vdots$	$d(R_2,2)=1$		$\vdots$	$d(i,m+1)$		$d(i,m+n)$	h_i
$\vdots$	0		0	$\vdots$		$\vdots$	$\vdots$
$\vdots$	$\vdots$		$d(R_{m-1}+1,m)=1$	$\vdots$		$\vdots$	$\vdots$
$\vdots$	$\vdots$			$\vdots$		$\vdots$	$\vdots$
0	0		$d(R_m,m)=1$	$d(R,m+1)$		$d(R,m+n)$	h_R
$H(x_1)$	$H(x_2)$		$H(x_m)$	$P(y_{m+1})$		$P(y_{m+n})$	

Proof: From (3), there is

$$\sigma_i^{(m+2)}(t) = \frac{\sum_r d(r,m+2)\cdot h_r^{(m+1)}(t)}{P(y_{m+2})} = \frac{1}{P(y_{m+2})}\sum_r d(r,m+2)\cdot\frac{h_r^{(m)}(t)}{\sigma_r^{(m+1)}(t)}.$$

Since $\forall d(i,j) > 0$

$$\sigma_i^{(m+1)}(t) = \frac{\sum_r d(r,m+1)\cdot h_r^{(m)}(t)}{P(y_{m+1})}$$

holds $\forall i$. Thus, $\sigma_i^{(m+1)}(t)$ is a constant that does not depend on i and

$$\begin{aligned}\sigma_i^{(m+2)}(t) &= \frac{1}{P(y_{m+2})}\sum_r d(r,m+2)\cdot\frac{h_r^{(m)}(t)}{\sigma_r^{(m+1)}(t)}\\ &= \frac{1}{P(y_{m+2})}\cdot\frac{1}{\sigma_i^{(m+1)}(t)}\\ &\quad\times\sum_r d(r,m+2)h_r^{(m)}(t)\\ \Rightarrow \sigma_i^{(m+1)}(t)\sigma_i^{(m+2)}(t) &= \frac{1}{P(y_{m+2})}\sum_r d(r,m+2)h_r^{(m)}(t).\end{aligned}$$

Inductively, we have

$$\prod_{k=m+1}^{m+n}\sigma_i^{(k)}(t) = \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)\cdot h_r^{(m)}(t). \tag{5}$$

For a given $i \in 1,\ldots,m$, the ith row in A_1 only has one nonzero element. Suppose it emerges in the $k_{i,0}$th column and $d(i,k_{i,0}) = 1$, $(1 \le k_{i,0} \le m)$. Therefore

$$\sigma_i^{(k)}(t) = 1 \quad \forall k \in [1,m], \quad k \ne k_{i,0}.$$

Note that during household fitting, the weight cells are actually partitioned into m disjoint subsets and each update only involves one of them. This elicits

$$\begin{aligned}\sigma_i^{(k_{i,0})}(t) &= \frac{\sum_r d\big(r,k_{i,0}\big)\cdot h_r^{(k_{i,0}-1)}(t)}{H\big(x_{k_{i,0}}\big)}\\ &= \frac{\sum_r d\big(r,k_{i,0}\big)\cdot h_r^{(m+n)}(t-1)}{H\big(x_{k_{i,0}}\big)}\end{aligned}$$

which is also a constant and does not rely on i. Thus, (5) can be written as

$$\begin{aligned}\prod_{k=m+1}^{m+n}\sigma_i^{(k)}(t) &= \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)\cdot h_r^{(m)}(t)\\ &= \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)\cdot\frac{h_r^{(m+n)}(t-1)}{\prod_{k=1}^{m}\sigma_i^{(k)}(t)}\\ &= \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)\cdot\frac{h_r^{(m+n)}(t-1)}{\sigma_i^{(k_{i,0})}(t)}\\ &= \frac{\sum_r d(r,m+n)\cdot h_r^{(m+n)}(t-1)}{\sigma_i^{(k_{i,0})}(t)\cdot P(y_{m+n})}.\end{aligned}$$

So, we have

$$\begin{aligned}\prod_{k=1}^{m+n}\sigma_i^{(k)}(t) &= \prod_{k=1}^{m}\sigma_i^{(k)}(t)\prod_{k=m+1}^{m+n}\sigma_i^{(k)}(t)\\ &= \sigma_i^{(k_{i,0})}(t)\cdot\prod_{k=m+1}^{m+n}\sigma_i^{(k)}(t)\\ &= \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)h_r^{(m+n)}(t-1)\\ &= 1.\end{aligned}$$

The last equation sign holds because

$$\sum_r d(r,m+n)\cdot h_r^{(m+n)}(t-1) = P(y_{m+n})$$

after the last constraint fitted when $t > 1$. ■

Proposition 1 indicates after the first iteration, the weights will remain unchanged in subsequent computation. Specifically, consider the first iteration where $t = 1$

$$\prod_{k=1}^{m+n}\sigma_i^{(k)}(1) = \frac{1}{P(y_{m+n})}\sum_r d(r,m+n)h_r(0). \tag{6}$$

If the IPU converges to an optimal solution, it will be completed in the first round, which elicits

$$\sum_r d(r,j)\cdot h_r^{(m+n)}(1) = H\big(x_j\big) \quad \forall j \in [1,m]$$

$$\sum_{r} d(r,j) \cdot h_r^{(m+n)}(1) = P(y_j) \quad \forall j \in [m+1, m+n] \quad (7)$$

when there exist positive solutions. Equations (6) and (7) lead to

$$\sum_{i} \left[d(i,j) - \frac{H(x_j)}{P(y_{m+n})} \cdot d(i,m+n) \right] \cdot h_i(0) = 0$$
$$\sum_{i} \left[d(i,j) - \frac{P(y_j)}{P(y_{m+n})} \cdot d(i,m+n) \right] \cdot h_i(0) = 0 \quad (8)$$

where $\forall j \in [1,m]$ and $[m+1, m+n]$, respectively. Note (8) holds automatically for $j = m+n$. Therefore, (8) is usually an underdetermined system with R variables and $(m+n-1)$ sequations. The above analysis manifests that if the optimization problem shown in Table II has feasible positive weights, only when the initial solution $h(0)$ satisfies (8) that can the IPU find one. This type of $h(0)$, however, cannot always be guaranteed by the input sample.

Case 2: A_2 has some zero elements. In contrast with the previous one, this case is more complicated. If the ith row of A_2 contains zero(s), then $\sigma_i^{(m+1)}(t), \ldots, \sigma_i^{(m+n)}(t)$ are no longer constants and we cannot receive (5) now. In the following, we will prove by contradiction.

Assume that IPU converges to a positive solution of Table II for the first time after the t-th iteration, then we have error

$$L^{(m+n)}(t-1) = \sum_{1 \le j \le m} \frac{\left| \sum_i d(i,j) h_i^{(m+n)}(t-1) - H(x_j) \right|}{H(x_j)} + \sum_{m+1 \le j \le m+n} \frac{\left| \sum_i d(i,j) h_i^{(m+n)}(t-1) - P(y_j) \right|}{P(y_j)} > 0 \quad (9)$$

and

$$L^{(m+n)}(t) = \sum_{1 \le j \le m} \frac{\left| \sum_i d(i,j) h_i^{(m+n)}(t) - H(x_j) \right|}{H(x_j)} + \sum_{m+1 \le j \le m+n} \frac{\left| \sum_i d(i,j) h_i^{(m+n)}(t) - P(y_j) \right|}{P(y_j)} = 0. \quad (10)$$

The superscript $(m+n)$ means after fitting the $(m+n)$th constraint. For convenience, let $L^{(0)}(t) = L^{(m+n)}(t-1)$. Then consider the error sequence

$$\left\{ L^{(0)}(t), L^{(1)}(t), \ldots, L^{(m+n)}(t) \right\}.$$

Let $L^{(c)}(t)$ be the last positive error ($1 \le c \le m+n-1$). Thus

$$L^{(c)}(t) = \sum_{1 \le j \le m} \frac{\left| \sum_i d(i,j) h_i^{(c)}(t) - H(x_j) \right|}{H(x_j)} + \sum_{m+1 \le j \le m+n} \frac{\left| \sum_i d(i,j) h_i^{(c)}(t) - P(y_j) \right|}{P(y_j)} > 0 \quad (11)$$

and

$$L^{(c+1)}(t) = \sum_{1 \le j \le m} \frac{1}{H(x_j)} \left| \sum_i d(i,j) h_i^{(c+1)}(t) - H(x_j) \right| + \sum_{m+1 \le j \le m+n} \frac{\left| \sum_i d(i,j) h_i^{(c+1)}(t) - P(y_j) \right|}{P(y_j)} = 0. \quad (12)$$

Note that (11) and (12) are both the sum of $(m+n)$ items with absolute value signs. Therefore, each item in (12) equals zero. And there is at least one positive item in (11) (if not, the greater sign cannot hold). Again, consider the last positive item, represented as the kth one.

1°: If the last positive item in (11) comes from individual constraints, we first investigate the situation that $m+1 \le k < m+n$, which means

$$\frac{1}{P(y_k)} \left| \sum_i d(i,k) h_i^{(c)}(t) - P(y_k) \right| > 0$$
$$\Rightarrow \sum_i d(i,k) h_i^{(c)}(t) \ne P(y_k). \quad (13)$$

Since the kth item is the last positive one, there is

$$\frac{1}{P(y_{k+1})} \left| \sum_i d(i,k+1) h_i^{(c)}(t) - P(y_{k+1}) \right| = 0$$
$$\Rightarrow \sum_i d(i,k+1) h_i^{(c)}(t) = P(y_{k+1}). \quad (14)$$

From (12), we have

$$\frac{1}{P(y_k)} \left| \sum_i d(i,k) h_i^{(c+1)}(t) - P(y_k) \right| = 0$$
$$\Rightarrow \sum_i d(i,k) h_i^{(c+1)}(t) = P(y_k) \quad (15)$$

$$\frac{1}{P(y_{k+1})} \left| \sum_i d(i,k+1) h_i^{(c+1)}(t) - P(y_{k+1}) \right| = 0$$
$$\Rightarrow \sum_i d(i,k+1) h_i^{(c+1)}(t) = P(y_{k+1}). \quad (16)$$

According to (15), we know

$$\sum_i d(i,k) h_i^{(c+1)}(t) = \sum_i d(i,k) \frac{h_i^{(c)}(t)}{\sigma_i^{(c+1)}(t)} = P(y_k).$$

This indicates that at least one $\sigma_i^{(c+1)}(t)$ does not equal 1 [if not, (13) will turn into an equation], denoted as $\sigma_r^{(c+1)}(t) \ne 1$. On the other hand, from (14) and (16), we have

$$\sum_i d(i,k+1) \left[h_i^{(c)}(t) - h_i^{(c+1)}(t) \right] = 0$$
$$\Rightarrow \sum_i d(i,k+1) h_i^{(c)}(t) \left[1 - 1/\sigma_i^{(c+1)}(t) \right] = 0. \quad (17)$$

Recall that

$$\sigma_i^{(c+1)}(t) = \begin{cases} \frac{\sum_i d(i,c+1) h_i^{(c)}(t)}{P(y_{c+1})}, & d(i,c+1) > 0 \\ 1, & d(i,c+1) = 0. \end{cases}$$

This means $\forall i \in \{1, \ldots, R\}$, each $\sigma_i^{(c+1)}(t) \geq 1$ or each $\sigma_i^{(c+1)}(t) \leq 1$. Therefore, each $[1-1/(\sigma_i^{(c+1)}(t))] \geq 0$ or each $[1-1/(\sigma_i^{(c+1)}(t))] \leq 0$. Since all $d(i, k+1)h_i^{(c)}(t) > 0$, (16) indicates that each $\sigma_i^{(c+1)}(t) = 1$. Specifically, $\sigma_r^{(c+1)}(t) = 1$. Now, we have a contradiction. If $k = m + n$, define $P(y_{m+n+1}) = H(x_1)$ and $d(i, m+n+1) = d(i, 1)$, the proof is similar.

2°: If the last positive item of (15) comes from household constraints, replace $P(y_k)$ with $H(x_k)$. The proof is similar.

IV. General Population Synthesis Using Bilevel Optimization

As can be seen in the previous sections, the IPU algorithm only considers individual and household. In our daily life, however, other social relationships may also extensively impact people's schedules and behaviors. Such social connections have tied individuals together, and will undoubtedly trigger particular personal activities. For example, the geographic locations of corporations and schools may probably determine the routine destinations of their affiliated individuals. Therefore, in this section, we extend the problem by incorporating more types of social organizations.

Mathematically, the major challenge from one type of social organizations to a more general case is that the heuristic search, like the IPU fitting, is only applicable to the organization-individual (household individual in the previous case) sequence but rather than the reversed one. When considering more social organization types, each type can generate a population dataset using IPU and these populations may probably not be consistent. Therefore, it is required to formulate the social and individual constraints equally. For simplicity, we consider two types of organizations, which means that the constraints come from three levels: 1) individual; 2) household; and 3) enterprise. Let $(z_{R_{m+1}}, \ldots, z_{R_{m+l}})$ be the enterprise variables, and $[E(z_{m+1}), \ldots, E(z_{m+l})]$ be the enterprise constraints. The general problem is modeled as Table III, where the two social relationships have their own weights. We denote the two coefficient matrices of household and enterprise as A_H^T (with $R_m \times m$ dimensions) and A_E^T [with $(R_{m+l} - R_m) \times l$ dimensions]. Their corresponding individual mapping matrices are $A_{H\to P}^T$, $A_{E\to P}^T$ [with $R_m \times n$ and $(R_{m+l} - R_m) \times n$ dimensions]. According to each level constraint, we have

$$\begin{bmatrix} A_H \\ A_{H\to P} \end{bmatrix} x = \begin{bmatrix} H \\ P \end{bmatrix}, \quad \begin{bmatrix} A_E \\ A_{E\to P} \end{bmatrix} z = \begin{bmatrix} Q \\ P \end{bmatrix}. \tag{18}$$

Equation (18) indicates that each social organization distribution needs to match its constraints, while the individual distribution converted by each organization also needs to match the mutual population constraints. At the population level, each individual distribution converted by a particular social relationship must be the same, which means

$$A_{H\to P} \cdot x = A_{E\to P} \cdot z. \tag{19}$$

In application, the equality of (18) may not strictly hold, since our data sources usually involve noises. However, (19) needs to be strictly guaranteed to keep the uniqueness of the population. This naturally leads to the following bilevel optimization problem:

$$\begin{aligned} \arg\min_{x,z} \quad & \|A_H \cdot x - H\|_2 + \alpha \|A_E \cdot z - Q\|_2 \\ & + \beta \|A_{H\to P} \cdot x - P\|_2 + \gamma \|A_{E\to P} \cdot z - P\|_2 \\ \text{s.t.} \quad & A_{H\to P} \cdot x = A_{E\to P} \cdot z, \quad x \geq 0, \quad z \geq 0 \end{aligned} \tag{20}$$

where $\lambda = (\alpha, \beta, \gamma)^T$ is regularized coefficients. Here, $\|\cdot\|_2$ can be changed to other types of norms. Basically, problem (20) is a linear optimization, and classic algorithms, such as gradient descent or multiobjective optimization [33], [34] can be exploited to compute the optimal solutions. For a more general case, one can easily extend the problem (20) which looks like

$$\begin{aligned} \arg\min_{x_i} \quad & \sum_i \alpha_i \|A_i \cdot x_i - C_i\|_2 + \sum_j \beta_i \left\|A_{j\to P} \cdot x_j - P\right\|_2 \\ \text{s.t.} \quad & A_{1\to P} \cdot x_1 = A_{2\to P} \cdot x_2 = \cdots \quad x_i \geq 0. \end{aligned} \tag{21}$$

This is a little more complicated but still solvable.

V. Numerical Test and Population Synthesis Experiments

In practice, the number of constraints is usually much smaller than the individual or social relation types, since statistical results only provide partial views of the variables' joint distribution. This causes the columns in Table III to be much fewer than its rows. Thus, problem (20) actually becomes an underdetermined system. In order to show the computational process of IPU and test our proposed optimization method, this section conducts experiments of pure numerical computation and Chinese population synthesis.

A. Numerical Computation

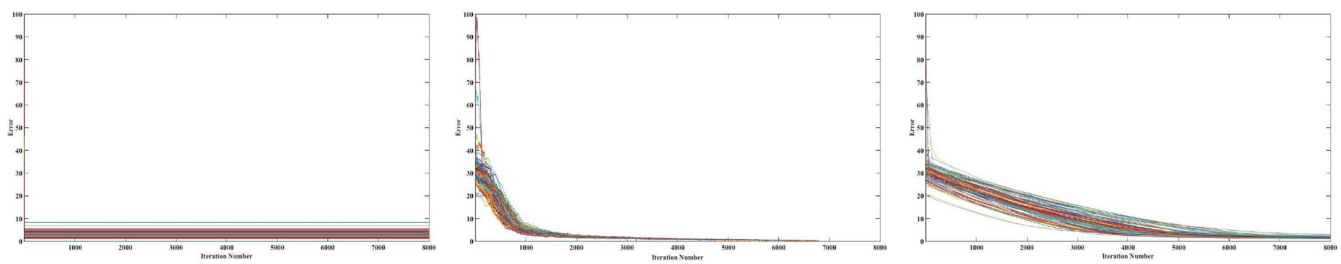
Numerical computations are designed as two groups: frequency matrix A_2 does not contain (first group) or contains (second group) zero cells. Both groups conduct 100 experiments to investigate the computations with different inputs. These experiments simulate 80 variables with 60 constraints, among which 20 are "household" constraints. In the first group, each experiment generates a frequency matrix A and a positive solution randomly from 1 to 100. Constraints are achieved by calculating the inner products of each column from A and the predefined solution. This process is to guarantee the existence of positive solutions of the sequations. Then, the three algorithms, IPU, simulated annealing, and gradient-descent-based bilevel optimization, are applied to solve the problem with the same initial weights randomly from 1 to 100. Fig. 1 shows the results of the first group. As our theoretical analysis, errors from IPU remain unchanged after the first iteration. While errors from the other two algorithms decrease in general. In addition, IPU relies on the initial weights since different inputs get different errors. In contrast, gradient descent and simulated annealing are more robust and can converge from various inputs. This group of results indicates that IPU cannot converge when A_2 does not have zero cells, and thus support our theoretical conclusion.

The second group of experiments is similar to the first group, only differentiating at the zero cells in A_2. When generating the frequency matrices, some (about 20%) cells are

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
GENERAL CASE OF POPULATION SYNTHESIS

HH Type 1	$\cdots$	HH Type m	HH Weights	Ind Type $(m+l+1)$	$\cdots$	Ind Type $(m+l+n)$
$d(1,1)$	$\cdots$	0	x_1	$d(1,m+l+1)$	$\cdots$	$d(1,m+l+n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$d(R_1,1)$	$\cdots$	0	$\vdots$	$\vdots$		$\vdots$
	$\ddots$		$\vdots$	$\vdots$		$\vdots$
0	$\cdots$	$d(R_{m-1}+1,m)$	$\vdots$	$\vdots$		$\vdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
0	$\cdots$	$d(R_{m+1},m)$	x_{R_m}	$\vdots$		$\vdots$
$H(x_1)$	$\cdots$	$H(x_m)$		$P(y_{m+l+1})$	$\cdots$	$P(y_{m+l+n})$
En. Type $m+1$	$\cdots$	En. Type $m+l$	En. Weights			
$d(R_m+1,m+1)$	$\cdots$	0	z_{R_m+1}	$d(R_m+1,m+l+1)$	$\cdots$	$d(R_m+1,m+l+n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
$d(R_{m+1},m+1)$	$\cdots$	0	$\vdots$	$\vdots$		$\vdots$
	$\ddots$		$\vdots$	$\vdots$		$\vdots$
0	$\cdots$	$d(R_{m+l-1}+1,m+l)$	$\vdots$	$\vdots$		$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
0	$\cdots$	$d(R_{m+l},m+l)$	$z_{R_{m+l}}$	$d(R_{m+l},m+l+1)$	$\cdots$	$d(R_{m+l},m+l+n)$
$E(z_{m+1})$	$\cdots$	$E(z_{m+l})$		$P(y_{m+l+1})$	$\cdots$	$P(y_{m+l+n})$



(a) (b) (c)

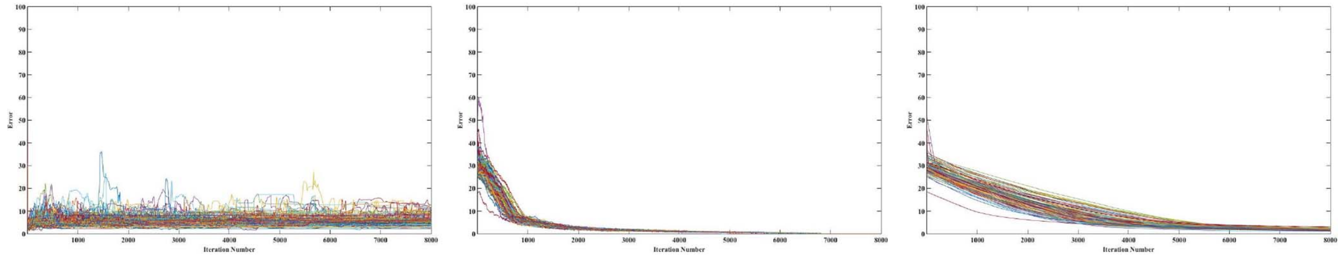
Fig. 1. Numerical computational results, A_2 does not contain zero cells. (a)–(c) are IPU, simulated annealing, and gradient descent, respectively.

randomly set to be zeros in A_2. Other inputs are all kept the same as the first group. Actually, real population synthesis mostly belongs to this group, since every type of household or other social organization cannot contain all types of individuals. Fig. 2 gives the results. As can be seen, gradient descent and simulated annealing almost perform a similar trend as the first group. But the IPU algorithm displays random error distributions during computation, which indicates that they cannot converge to a solution. Thus, the results also support our theoretical conclusion from case 2.

B. Chinese Nationwide Population Synthesis

To further validate our theoretical conclusions and test our proposed method, this section compares the three algorithms using real Chinese census data. As one of the most populous and heterogeneous countries, the Chinese population synthesis scenario is representative enough for such a validation. Chinese 5th national census in 2000 is used as our inputs. The first type is cross-classification tables of the total national population, households, and enterprises. Such tables, as illustrated in Table IV, give joint distributions of a part of (but not the whole) attributes from each level, respectively. For simplicity, we choose five individuals, four households, and three enterprise attributes to study (Table V). These attributes are sufficient to support our experiment, as shown in the following. The second type of data is disaggregate samples, each of which reveals the whole detailed attributes of an individual (with private information omitted). The samples are composed of stochastically extracted households from original census data, and include 345 167 households with 1 180 111 individual records. Compared with the total, our disaggregate samples account for 0.95% of the whole population.

The experiment starts with the computation of constraints for each level, that is, the values of $H(x_i)$, $Q(z_j)$, and $P(y_k)$, where $(1 \leq i \leq m)$, $(m+1 \leq j \leq m+l)$, and $(m+l+1 \leq k \leq m+l+n)$. This can be completed by the existing approach using the cross-classification tables listed in Table IV [28].



(a) (b) (c)

Fig. 2. Numerical computational results, A_2 contains zero cells. (a)–(c) are IPU, simulated annealing, and gradient descent, respectively.

TABLE IV
CROSS-CLASSIFICATION TABLES FROM DIFFERENT LEVELS

No.	Cross-Classification Tables	Level Types
1	Gender×Res. Prov.×Res. Type×HH Type	Ind.
2	Gender×Res. Prov.×Res. Type×Age	Ind.
3	Res. Prov.×Res. Type×HH Type	HH
4	Res. Prov.×Res. Type×Member Num.	HH
5	Prov.×Enter. Type	Enter.
6	Enter. Type×Enter. Scale	Enter.

TABLE V
INDIVIDUAL AND SOCIAL ORGANIZATION ATTRIBUTES

Attr.	Values	Num. of Values
Individual Attributes		
Gender	male, female	2
Res. Prov.	Beijing, Tianjin, ...	31
Res. Type	city, town, rural	3
HH Type	family, collective HH	2
Age Interval	0-5, ..., 96-100, ≥100	21
Household Attributes		
HH Type	family, collective HH	2
Res. Prov.	Beijing, Tianjin, ...	31
Res. Type	city, town, rural	3
Mem. Num.	1, ..., 9, ≥10	10
Enterprise Attributes		
Prov.	Beijing, Tianjin, ...	31
Enter. Type	Corporation, Industrial Unit, None	3
Enter. Scale	≤7, 8-19, ..., ≥10000	10

After that, we further analyze each household and enterprise type in the disaggregate samples. For a particular organization type, the numbers of its members are used as the row coefficients, and the number of organizations in such a type contained in samples is used as the corresponding initial weight. In our case, the numbers of constraints are 1023 from the household, 651 from the enterprise, and 5170 from the individual, which means that m, l, and n in Table III are equal to 1023, 651, and 5170, respectively. In addition, the samples contain 95 368 types of households and 1085 types of enterprises, which indicates that the coefficient matrix A_H^T has 95 368 rows ($R_m = 95\,368$) and A_E^T has 1085 rows ($R_{m+l} - R_m = 1085$).

When the coefficient matrices, constraints, and initial weights are determined, the three algorithms—IPU, simulated annealing, and gradient-descent-based bilevel optimization—are applied to iteratively compute the final weights, respectively. The regularized coefficients (α, β, γ) are all set to be 1. For the simulated annealing and gradient-descent methods, their search step lengths (also referred to as learning rate in some literature) are adaptively computed in order to keep the updated solutions significantly changed. Fig. 3 illustrates an overview of the final errors, in which the unit of iteration number is 10 000. Note that the IPU algorithm only computes individual and household constraints, while the other two deal with the whole three levels. As can be seen in the figure, the three methods are all able to reduce the L1 error defined by (4). The IPU algorithm quickly reaches equilibrium after about ten iterations. Its speed of convergence is much faster than the other two. However, this algorithm cannot further reduce the error during the later computation. Such a phenomenon clearly supports our theoretical conclusions in Section III. Simulated annealing converges much slower than the other two. At the early stage, it reduces the error quickly and steadily. After that, the speed of error decrease becomes much slower. Such a trend is consistent with the characteristic of simulated annealing, where the searching is stochastic and relies on effective heuristics. This trial-and-error process may cost extra iterations to find a "right" direction for error reduction. For our proposed bilevel optimization using gradient descent, the decreasing trend is much more obvious. The L1 error reduces below IPU at about 1 100 000 iterations, and finally converges to 1736 around 1 800 000 iterations. The error curve shows piecewise smooth due to the adaptive change of the step length (also referred to as a learning rate in some literature). The final result clearly indicates that the bilevel optimization outperforms the IPU method with 35.37% improvement relatively.

Computational performance is shown in Table VI. It can be seen that the IPU converges when its L1 error reaches 2686.5519, while our bioptimization converges to 1735.728. The average computational times are 37 min 30 s and 17 h 20 min 24 s, respectively. In contrast, simulated annealing converges much slower. It takes 17 h 55 min 30 s to reduce the L1 error into 2948.2651. Obviously, IPU gets the fastest convergence speed, and bilevel optimization achieves the highest accuracy.

Another important phenomenon from the experiment is that the errors of the three algorithms always stay positive and cannot be reduced to a near-zero level. This is because there exists a bias in our disaggregate sample. Specifically, the

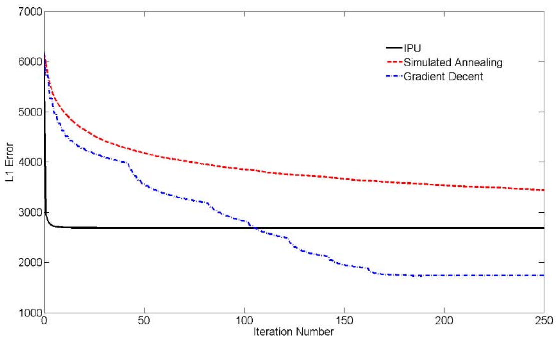


Fig. 3. Errors of Chinese national population synthesis.

TABLE VI
COMPUTATIONAL PERFORMANCE

	IPU	Bi-Level Opt.	Sim. Anneal.
Iter. Num.	1,262	1,830,000	2,700,000
L1 Error	2686.5447	1735.728	2948.2651
Ave. Com. Time	00:37:30	17:20:24	17:55:30
Environment	Software: Java 1.8 OS: Windows 7 (x64) CPU: Intel Core i7-4790 (8 cores, 3.6GHz) RAM: 8 GB		

disaggregate sample does not cover all types of households and enterprises that are contained by the cross-classification tables. In other words, there are a certain number of contradictory sequations, where the individual coefficient matrix has all-zero columns but their corresponding constraints $P(y_j)$ remain positive. According to our investigation, the number of such contradictory sequations is 517, and they also bring an error sum of 517.

VI. CONCLUSION AND DISCUSSION

Population synthesis is fundamental to the analysis of urban transportation and other social-ecological systems using the multiagent approach. As a representative method in this field, IPU can simultaneously fit constraints from household and individual levels. In this article, however, we prove that this algorithm is not able to obtain an optimal population in theory. Its error may probably remain stable after only a few iterations. After theoretical analysis, we extend the problem into a more general case, considering not only the household but also other social organizations. A bilevel optimization method is introduced to solve the general problem. Comparative experiments both on the pure numerical test and real Chinese population synthesis clearly support our theoretical conclusions and indicate that our proposed method can outperform the IPU algorithm with even more social organization constraints.

Basically, the extended problems (20) and (21) are convex linear optimizations, which means the classic algorithms, such as gradient descent, do not suffer from divergence. But this does not mean the error can be reduced to near zero, as illustrated in the experiments. The final accuracy depends on the quality of the disaggregate sample and further on whether contradictory sequations exist in the constraints. Thus, to eliminate the noise, preliminary data clean might be required according to the confidence of each data source. Another way is to use the role assignment without a disaggregate sample to minimize the overall deviations from social organizations and individuals [35], [36]. As a general model for society, a direct role assignment, together with its related E-CARGO, may be a feasible way to avoid the sample bias [37].

A second issue is the speed of convergence. As shown in our experiments, many classic searching algorithms, especially the ones with stochastic heuristics, may converge slowly. To accelerate the computation, a potential direction is to combine them with IPU, where the IPU is exploited to provide a near-optimal solution in the early stage and the heuristic searching uses such a solution for further optimization. Evolutionary computation can be also introduced for large-scale searching [38]–[41].

REFERENCES

[1] J. Wang, A. Sheng, D. Xu, Z. Chen, and Y. Su, "Event-based practical output regulation for a class of multiagent nonlinear systems," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3689–3698, Oct. 2019.

[2] H. Zhang, D. Yue, C. Dou, W. Zhao, and X. Xie, "Data-driven distributed optimal consensus control for unknown multiagent systems with input-delay," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2095–2105, Jun. 2019.

[3] P. Ye and D. Wen, "A study of destination selection model based on link flows," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 428–437, Mar. 2013.

[4] A. Grow and J. V. Bavel, Eds., *Agent-Based Modelling in Population Studies: Concepts, Methods, and Applications*. Cham, Switzerland: Springer, 2017.

[5] A. G. Wilson and C. E. Pownall, "A new representation of the urban system for modeling and for the study of micro-level interdependence," *Area*, vol. 8, no. 4, pp. 246–254, 1976.

[6] P. Williamson, M. Birkin, and P. Rees, "The estimation of population microdata by using data from small area statistics and samples of anonymised records," *Environ. Plan.*, vol. 30, no. 5, pp. 785–816, 1998.

[7] B. Farooq, M. Bierlaire, R. Hurtubia, and G. Flotterod, "Simulation based population synthesis," *Transp. Res. B Methodol.*, vol. 58, pp. 243–263, Dec. 2013.

[8] Z. Zhang and W. Pedrycz, "A consistency and consensus-based goal programming method for group decision-making with interval-valued intuitionistic multiplicative preference relations," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3640–3654, Oct. 2019.

[9] Y. Tang, Z. Deng, and Y. Hong, "Optimal output consensus of high-order multiagent systems with embedded technique," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1768–1779, May 2019.

[10] Y. Li, H. Li, X. Ding, and G. Zhao, "Leader–follower consensus of multiagent systems with time delays over finite fields," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3203–3208, Aug. 2019.

[11] R. Li and G.-H. Yang, "Consensus control of a class of uncertain nonlinear multiagent systems via gradient-based algorithms," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2085–2094, Jun. 2019.

[12] J. Hou and R. Zheng, "Hierarchical consensus problem via group information exchange," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2355–2361, Jun. 2019.

[13] N. Mu, Y. Wu, X. Liao, and T. Huang, "Input time delay margin in event-triggered consensus of multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1849–1858, May 2019.

[14] P. Ye, T. Wang, and F.-Y. Wang, "A survey of cognitive architectures in the past 20 years," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3280–3290, Dec. 2018.

[15] P. Ye, S. Wang, and F.-Y. Wang, "A general cognitive architecture for agent-based modeling in artificial societies," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 1, pp. 176–185, Mar. 2018.

[16] X. You, C. Hua, and X. Guan, "Self-triggered leader-following consensus for high-order nonlinear multiagent systems via dynamic output feedback control," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2002–2010, Jun. 2019.

[17] H.-X. Hu, G. Wen, W. Yu, J. Cao, and T. Huang, "Finite-time coordination behavior of multiple Euler–Lagrange systems in cooperation-competition networks," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2967–2979, Aug. 2019.

[18] K. Lu, G. Jing, and L. Wang, "Distributed algorithms for searching generalized Nash equilibrium of noncooperative games," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2362–2371, Jun. 2019.

[19] H. Zhao, S. Tang, and Y. Lv, "Generating artificial populations for traffic microsimulation," *IEEE Intell. Transp. Mag.*, vol. 1, no. 3, pp. 22–28, Nov. 2009.

[20] F. Gargiulo, S. Ternes, S. Huet, and G. Deffuant, "An iterative approach for generating statistically realistic populations of households," *PLoS ONE*, vol. 5, no. 1, 2010, Art. no. e8828.

[21] J. Auld and A. Mohammadian, "An efficient methodology for generating synthetic populations with multiple control levels," *Transp. Res. Rec.*, vol. 47, no. 1, pp. 138–147, 2010.

[22] J. Barthelemy and P. L. Toint, "Synthetic population generation without a sample," *Transp. Sci.*, vol. 47, no. 2, pp. 266–279, 2013.

[23] S. Huet, M. Lenormand, G. Deffuant, and F. Gargiulo, "Parameterisation of individual working dynamics," in *Empirical Agent-Based Modeling—Challenges and Solutions: Volume 1, the Characterisation and Parameterisation of Empirical Agent-Based Models*. New York, NY, USA: Springer, 2014, pp. 133–169.

[24] L. Ma and S. Srinivasan, "Synthetic population generation with multilevel controls: A fitness-based approach and validations," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 30, no. 2, pp. 135–150, 2015.

[25] N. Huynh, J. Barthelemy, and P. Perez, "A heuristic combinatorial optimization approach to synthesizing a population for agent based modeling purposes," *J. Artif. Soc. Soc. Simul.*, vol. 19, no. 4, p. 11, 2016.

[26] P. Anderson and B. Farooq, "A generalized partite-graph method for transportation data association," *Transp. Res. C*, vol. 76, pp. 150–169, Mar. 2017.

[27] P. Ye, X. Wang, C. Chen, Y. Lin, and F.-Y. Wang, "Hybrid agent modeling in population simulation: Current approaches and future directions," *J. Artif. Soc. Soc. Simul.*, vol. 19, no. 1, p. 12, 2016.

[28] P. Ye, X. Hu, Y. Yuan, and F.-Y. Wang, "Population synthesis based on joint distribution inference without disaggregate samples," *J. Artif. Soc. Soc. Simul.*, vol. 20, no. 4, p. 16, 2017.

[29] P. Ye and X. Wang, "Population synthesis using discrete copulas." in *Proc. IEEE Int. Conf. Intell. Transport. Syst. (ITSC)*, 2018, pp. 479–484.

[30] L. Sun, A. Erathb, and M. Cai, "A hierarchical mixture modeling framework for population synthesis," *Transp. Res. B Methodol.*, vol. 114, pp. 199–212, Aug. 2018.

[31] X. Ye, 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," in *Proc. 88th Annu. Meeting Transp. Res. Board*, 2009, pp. 1–24.

[32] W. E. Deming 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," *Ann. Math. Stat.*, vol. 11, no. 1, pp. 427–444, 1940.

[33] Z.-G. Chen *et al.*, "Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912–2926, Aug. 2019.

[34] H. Zhang, F. Teng, Q. Sun, and Q. Shan, "Distributed optimization based on a multiagent system disturbed by general noise," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3209–3213, Aug. 2019.

[35] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[36] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 42, no. 3, pp. 739–750, May 2012.

[37] H. Zhu, "Role-based collaboration and the E-CARGO: Revisiting the developments of the last decade," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 1, no. 3, pp. 27–35, Jul. 2015.

[38] X.-G. Zhou and G.-J. Zhang, "Differential evolution with underestimation-based multimutation strategy," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1353–1364, Apr. 2019.

[39] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, Mar. 2019.

[40] H. Xu, W. Zeng, D. Zhang, and X. Zeng, "MOEA/HD: A multiobjective evolutionary algorithm based on hierarchical decomposition," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 517–526, Feb. 2019.

[41] S. Elsayed, R. Sarker, and C. A. Coello Coello, "Fuzzy rule-based design of evolutionary algorithm for optimization," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 301–314, Jan. 2019.

Peijun Ye (Member, IEEE) received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2013.

He is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. He was a Visiting Scholar with the Department of Cognitive Science, University of California at San Diego, San Diego, CA, USA, from 2017 to 2018. He has authored/coauthored more than 30 papers, six patents, and one monograph. His research interests mainly focus on multiagent systems, social simulation, artificial intelligence, and intelligent transportation systems.

Dr. Ye serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and the Secretary for the Social and Economic Computing ACM Chapter.

Bin Tian received the B.S. degree from Shandong University, Jinan, China, in 2009, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision, machine learning, and automated driving.

Yisheng Lv (Senior Member, IEEE) received the B.E. and M.E. degrees from the Harbin Institute of Technology, Harbin, China, in 2005 and 2007, respectively, and the Ph.D. degree in control theory and control engineering from the Chinese Academy of Sciences, Beijing, China, in 2010.

He is an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include traffic data analysis, deep learning, machine learning, and parallel traffic management.

Qijie Li is currently pursuing the bachelor's degree with the School of Artificial Intelligence and Automation, Beijing University of Technology, Beijing, China.

Since 2019, she has been working as an internal trainee with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Science, Beijing. She has authored/coauthored two patents. Her current research focuses on the task scheduling for edge computing, which is supported by the practical training plan project "intelligent edge computing technology and device" of high-level talents cross-cultivation in Beijing universities and colleges.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the University of Arizona, Tucson, AZ, USA, and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China, under the support of the Outstanding Chinese Talents Program. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. He is also the President of Qingdao Academy of Intelligent Industries and a Professor with the Macau University of Science and Technology. He is the President of CAA's Supervision Council, IEEE Council on RFID, and Vice President of IEEE Systems, Man, and Cybernetics Society. His current research focuses on methods and applications for parallel intelligence, social computing, and knowledge automation.