SUPPORTING PENSION PRE-CALCULATION WITH DYNAMIC MICROSIMULATION TECHNOLOGIES

Dávid Burka and László Mohácsi
Department of Computer Sciences
Corvinus University of Budapest
1093, Budapest, Fővám square 13-15
e-mail: david.burka@uni-corvinus.hu

József Csicsman and Benjámin Soós
Faculty of Sciences and Informatics
Department of Computer Optimalisation
University of Szeged
6720, Szeged, Árpád square 2.
e-mail: csicsman@calculus.hu

KEYWORDS

Microsimulation, Demography, Pension system, Ageing population

ABSTRACT

Population ageing induces many challenges in the pension system of developed countries. It is necessary to support the decision-making processes regarding these challenges by forecasting different future scenarios. Long-term forecasts are required to understand the development process of the population and the pension system. The microsimulation approach has many benefits over other forecasting methods, though it requires high level of programing skills and significant computing capacity. Moreover, a long-term demographic microsimulation must be dynamic and it should preferably also include the relations between individuals. In this paper, we will introduce two different microsimulation based solutions for the above-mentioned forecasting tasks. The first one is a complex model – aiming to forecast the Hungarian population – built in SAS, that can highlight the advantages of the microsimulation approach. The second solution is a Simulation Framework (written in C#), that aims to drastically reduce the difficulties regarding microsimulation using the findings of the SAS model. Our goal is to introduce our systems in the hope of future collaboration with economists and demographers.

INTRODUCTION

In the last decade, all OECD countries induced some kind of pension reform (OECD, 2015). This is a good indicator of how urgent it is to manage the challenges regarding the changing structure of the population. Pension systems can be categorized whether they are based on funded or unfunded plans. Some hybrid systems exist, but even those usually prioritize one of the plans.

A funded or pre-funded plan means that the contributions are invested in a fund to cover the benefits of the individual after the retirement. Unfunded – or so called pay-as-you-go – plans on the other hand cover the pension costs of the retired with the contributions of the active population. The two approaches must face different kinds of challenges.

In a funded system, the adequate living conditions of the elderly are not properly secured. People, who outlive their life expectancy will exhaust their fund even if it had high yield, and they might end up in poverty. In an unfunded system if the total contributions do not reach the amount of the total pension costs, the government must supplement the shortage from taxes. In recent years, the main reason for the shortage is the ageing of the population and the shift of the active worker per pensioner ratio. The pay-as-you-go systems can care for their elderly through the whole retirement since the funds will not run out. However, the monthly benefits are fixed at the time of retirement, and if high life expectancy is paired with high inflation, the pension payments might not be able to secure appropriate living conditions.

The adequacy and the financial sustainability of a pension system are conflicting interests, and both plans must aim to balance these interests. Funded plans are more sensitive regarding adequacy while unfunded plans often have to face the problems of financial sustainability. The systems introduced in this paper aim to support forecasts for both types, however our own models are built for Hungarian examples, thus we will mainly focus on the financial sustainability of pay-as-you-go systems.

AGEING POPULATION

The main reason behind the uncertainty regarding the unfunded pension systems is the ageing of the population. The life expectancies continuously increase while fertility rates decrease, thus the ratio of the active population and the pensioners worsens every year in most of the developed countries. Since unfunded plans cover their expenses via the contributions of the labor market, the shift of the ratio creates an ever-increasing burden on the society. There are three main factors that influence the structure of the population: the increasing life expectancies, the decreasing fertility rates, and the international migration.

The technological advancement has improved medicine and granted acceptable living conditions for many

Proceedings 31st European Conference on Modelling and Simulation ©ECMS Zita Zoltay Paprika, Péter Horák, Kata Váradi, Péter Tamás Zwierczyk, Ágnes Vidovics-Dancs, János Péter Rádics (Editors)
ISBN: 978-0-9932440-4-9/ ISBN: 978-0-9932440-5-6 (CD)

people, thus increasing the overall life expectancies at birth. This resulted in the increase of the worldwide ratio of the elderly (60+) population from 9.9 percent to 12.3 percent between 2000 and 2015 (United Nations, 2015).

From an evolutionary point of view the more children the better the survival chances of the gene pool. However, as the society prospers, the fertility rates tend to decrease. The social survival of the children becomes a more realistic concern than the biological survival. The death of a child is unlikely in a modern society so parents tend to focus on other worries regarding their offspring. Good studies, acceptable career path, and even finding a partner are important to become an honored member of the society. Ensuring this requires resources – like time and money – per child, so a smaller number of children is preferred, thus the fertility rates decrease.

International migration can have a significant effect on the structure of the population. For example, in Central- and Eastern European countries – where the pension plans are typically unfunded – the migration of the young workforce to the economically more prosperous regions enhances the negative effects of population ageing (Cerami, 2011).

These global trends are not likely to change soon, so it is important to prepare for the effects of the ongoing processes in the structure of the population. The changes will affect the society just as much as the economy (Kulik, 2014), thus it will be impossible to simply forecast every single effect that might influence the financial sustainability of the pension system. Rather, it would be more beneficial to model the cause and effect of as many scenarios as possible. However, this approach requires models that go beyond the simulation of death and birth, and include the relations between individuals.

RELATIONS

The relations and overall circumstances of an individual have a strong influence on his or her life path, and these effects can be confirmed through the analysis of statistical data. Married people, who lived together for a long time, often die in a short interval after each other, and married men tend to live longer than bachelors. The education of a child is highly correlated to the education level of the parents. Moreover, the income of the whole household has a more significant influence on ones living conditions than the individual income.

Examining and implementing the above-mentioned connections in a model can highlight such complex, wide-branching effects in economic indicators, that would be impossible to notice through simple trend-based forecasting methods. This way the number of estimated parameters will be lower, and – if we assume that the implemented connections are correct – the error of the forecasted indicators can also be reduced. However, such a detailed model requires far more resources to realize than traditional forecasting solutions.

FORECASTING METHODOLOGY

Demographic forecasting methods can be categorized on a macro-micro scale based on how detailed they are. The simplest forecasting methods – like fitting a trend or regression – can estimate the future development of indicators based on historical data. The cohort-component method groups the population based on the properties of the individual, and does not differentiate between the individuals. These properties are usually age and gender, but more complex models can include for example education, ethnicity, or residency. The groups are represented with the number of individuals in them, and the group numbers are changed at every iteration step based on a transition probability matrix. Agent based models extend this functionality by using multiple smaller transition matrices and different rule sets – if state-transition is not appropriate – to iterate the properties of agents. The agents can represent groups or even individuals, thus microsimulation is the most detailed, special version of the agent based models. It follows the life path of every single individual throughout the whole simulation, and changes their properties in every iteration step according to transition probabilities, rules, or interactions with other individuals.

Microsimulation – being one of the extremes – has many advantages over other methods. It allows the implementation of complex logics like the family relationships between individuals and it results in such a detailed estimated dataset, that would otherwise only be obtainable by repeated data collections. Using rules and fragmented transition matrices, a microsimulation model can be easily extended. However, as we move on the macro-micro scale toward the micro side, the models become increasingly more complex. A simple regression can be done in most of the statistical software packages (i.e.: R, SPSS) and a cohort-component model can be realized in a spreadsheet, but implementing a microsimulation model requires high level of software development skills and significant computing capacity.

These difficulties are the reason, why the microsimulation methodology has not become as wide-spread as the cohort-component method in the field of demographic forecasting, even though it was introduced more than half a century ago (Orcutt, 1957). Microsimulation started to appear more frequently in the last few decades thanks to the advancement in technology (Merz, 1991). However, most of the implementations are ad hoc solutions since even if the computing capacity is given, the microsimulation based systems still require programming skills and many work hours. This resulted in the implementation and publication of many systems that are difficult to reuse and understand by other researchers. It is usually more beneficial to develop a new solution for every new problem, especially if the target countries differ, since it would take too much effort to utilize someone else's work.

Our main goal is to address this problem. We created two different solutions, that both aim to show the capabilities and benefits of a microsimulation based demographic forecasting model, while being modular and easy to use for anyone interested.

DATA STRUCTURE

The Hungarian Central Statistical Office (HCSO) uses a microsimulation based forecasting system since the 90ies (Csicsman, 2012). We also focused on building models to forecast the Hungarian population and improve the existing solutions, thus we used datasets provided mainly by the HCSO. We classified our data sources into two different categories: attributes (or properties) and parameters.

Attributes are data values representing the individuals. These values get changed throughout the iteration steps according to the parameters or the attributes of others. The initial attributes should be based on a wide spread data collection like a census or a micro census. Often there exists no appropriately detailed dataset, so multiple datasets must be merged through statistical matching. We based our models on the results of the Hungarian micro census of 2005.

Every transition matrix or rule in our system is based on historical data. These parameters are the core of the model, and allow the interaction between the individuals. Some parameters barely change with time. For example, the probability of a newborn being male has been relatively constant for the past half century. However, most of the time the parameters change throughout the years. The life expectancies increase while the fertility rates decline. All time varying parameters must be forecasted before the start of the simulation. Fitting a trend or creating a regression to estimate the future values of these parameters can be adequate, but sometimes the parameters depend on multiple factors. For example, in most models, the mortality rates vary based on the age and gender of an individual, thus the estimation requires the forecasting of a whole matrix. A good practice is to use singular vector decomposition to obtain a time varying vector component, that can be forecasted by traditional methods (Burka, 2016).

We used similar, automated techniques to forecast the parameters in our systems. In the following chapters, we will describe both our SAS based model and our Simulation Framework in detail, to give an insight on what solutions we used and how these systems can be beneficial for other researchers, economists, or demographers.

SAS BASED MODEL

The motivation behind this project was to apply dynamic microsimulation methods on real life problems. We decided to implement our thoughts and ideas into a fast and flexible SAS program. SAS (Statistical Analysis Software) is a software suite developed by SAS Institute, that contains many advanced statistical features (e.g.: data management, predictive analysis). All the tools needed for mining important information from raw data - like data quality management or advanced analytics – are built in the software by default. Most of the microsimulation models that allow a wide range of configuration options for the user are built on their own, unique language. The Hungarian MIDAS_HU (Dekkers, 2015) or the Slovenian LIPRO (Majcen, 2011) programming tools are great examples. One of the greatest benefits of SAS is that it is a widely used statistical software, thus a great portion of those who are interested in demographic forecasts are already familiar with its syntax. In this section, we would like to introduce what we have accomplished in the SAS approach so far, and which direction we are going to focus on in the future.

Beyond the essential information such as personal and household data, there are lots of additional details about a given household's shopping habits, financial position, or school records of the individuals, which can be used in a model. However, we only included the most important attributes in order to create a general-purpose system. This means, that the software provides basic functions of demographic modules, such as death, birth, marriage, divorce, and besides these the user must include additional properties about studied topic. It is important to note, that the development of a new module requires the knowledge of the SAS Base and SAS Macro language, and of course the configuration of the base program.

Our system's most important feature is modularity. This means, that all the basic and user written modules can be added or removed by the user. An appropriate use of this feature can improve performance, and make it possible to split the research into smaller segments. We used the built-in procedures of SAS to forecast the parameters of the modules based on historical data between 1995 and 2015. The Proc Forecast toolbox can be used for exponential smoothing, applying Winters method, or thanks to the modularity of the system, we can fit the needs for different forecasting solutions, depending on the current task.

Thus far, we have mentioned fictional population that can be manipulated by demographic modules over time. Every year is an iteration step, and within every step the modules change the properties of the individuals. However, there are certain tasks that cannot be managed on the entity level. These tasks are going to be handled by sampling procedures, that take a set of individuals out from the base population, execute the given action on the selected group, then puts the manipulated portion of data back into the original dataset. The matching algorithm used for managing marriages can be a good example for this methodology. We select a portion of women grouped by age intervals, and we pair them with an equal portion of men with similar attributes (relevant matching conditions can be set to strict or loose). Finally, we remove the couples from their previous households, and assign a new household ID for them, thus creating a brand-new household for the married couple.

There are two different approaches to sampling: Unrestricted random sampling (URS) allows replacement in the selection, while Simple random sampling (SRS) is a selection with equal probability and does not allow replacement. In case of marriages URS would mean, that as the algorithm loops through the potential partners, one individual could get assigned to multiple partners. In this case, a breaking rule could decide who the individual gets partnered with. However, this would ruin the equal probability of distribution. The SRS approach solves this problem by removing the already selected individuals from the list of potential partners, but the additional queries will increase the runtime significantly.

During the simulation of a given demographic event, we can define conditions for every scenario that should be handled. For example, the death of different members in a household should be treated with a different approach, depending on the individual's role in the family. We can manage these set of rules like as a module. It is easy to modify, attach or detach them, thus we can set different levels of elaboration. However, this can influence runtime significantly.

SIMULATION FRAMEWORK

One of the main reasons, why microsimulation based models did not became wide-spread is, that high level of software development skills is necessary to implement them. Modular solutions, like our SAS based model could help researchers with basic programming skills and/or SAS knowledge to implement their own model. Our goal was to further reduce the necessary skill level while also managing the other disadvantages of a microsimulation based model. To achieve this, we started the development of a Simulation Framework based on three major requirements: speed, flexibility, and ease of use.

A demographic microsimulation requires a lot of computing capacity. For example, in the case of Hungary, the population consists of approximately 10 million entities. The aim is to create long term forecasts of 50 years, and multiple attributes must be recalculated at every iteration step. Considering, that managing the relationships further increases the complexity of the algorithm, the runtime of a simulation can easily reach multiple hours. Parallelizing of the simulation can reduce the runtimes significantly (Mohácsi, 2014). It is important to note though, that a parallel code can be far more complex than a single core solution, thus its realization requires high level of software development skills.

A fast framework, without a wide range of configuration options would quickly lose its purpose. Our goal is to reduce the limitations of the system as much as possible, and allow the users to create any model, that fits their needs. The framework should be used to compare multiple – even hundreds of – future scenarios.

Most published microsimulation based models can be accessed, and with enough time and development skills they can be modified for one's own needs. However, the resources spent on learning the system are taken away from understanding and implementing the model itself. It is important for a development environment to be intuitive. The user should be able to learn the handling of the available tools in a short time, thus reducing the resource costs of incorporating the results of others into the user's own model.

It is clear, that the above-mentioned properties are conflicting requirements. The optimization of the code makes it hard to understand, and the more configuration options are available, the harder it becomes to speed up the simulation. Our main goal was to find an appropriate balance between speed, flexibility, and ease of use, while developing our Simulation Framework. We had to keep a scripting feature, otherwise it would have been impossible to avoid limitations, thus we decided to implement a layer on top of the main code. The relatively simpler algorithms of the simulation steps – that implement the state transitions and rules described before – can be built from simple blocks in an intuitive graphical user interface. The optimized, parallel algorithms, that control the whole simulation, are hidden behind the scenes, and should not be accessed by the user. The blocks represent simple functions, variables, or matrices, but often have relatively complex algorithms behind them.

The configuration of the framework happens with Excel sheets that follow some simple rules, and forms that aid with the final settings based on the content of the sheets. The dynamic code is built in two layers. First the blocks available in the graphical programming interface – that allow the configuration and implementation of the model – are created based on the initially imported datasets. The second layer uses the structure built in this interface to generate the code for the simulation and the necessary queries to only save the indicator values of interest instead of the enormous dataset. Finally, the finished code runs on all the available processor cores. This approach significantly decreases the runtime since it saves time on directly writing down the different dynamic variables instead of searching for them in queries.

The main algorithm is relatively straight forward except of the thread safe random generator. This object generates a random seed for every single individual at the start of an iteration, and the individuals use these seeds in their own random generator to control their "decisions". The extra random number generators slightly worsen the performance, but in exchange we can ensure that the random number generation is deterministic, thus we can reproduce our results any time.

The relationships module is the most restricted part of our solution. It operates on the household level, since data is usually available for households instead of relationships. So, the individuals are divided into households at the start of the simulation. The structure of a household can change four ways: a new member can be born, a member can die, a member can leave or a member can join. Birth and death is handled by the default modules. If an individual leaves the household we automatically create a new single member household for him or her, and if two individuals decide to get into a relationship we move them into a brand-new household instead of joining their households together. This approach allows us to handle these two events in a single interface – like the simulation step interface – since in both cases the individuals join a new household.

If an individual leaves a household or gets into a relationship, they get a tag that represents the reason they joined the household for. Leaving is relatively simple, the individual needs a tag for the reason of leaving (i.e.: divorce, growing up or leaving family home), but getting into a relationship is more complicated. We wanted to allow any kind of relationship in our model, so the user can set up relationship types in the settings menu. A type (i.e.: marriage, life companion, flat mate) can have restrictions towards the available partners. For example, in our models we only allow women to choose partners and we do not allow same sex marriage so we exclude women completely from the available partners. The saying that "opposites attract" is statistically not true, most people choose partners with similar properties to themselves. (Of course, this only includes properties that can appear in a database.) In our framework, the user can select properties that are relevant in the choice of a partner regarding the given relationship type, and in the simulation the available partners will be divided into an n dimensional matrix of groups where n is the number of selected relevant properties. An individual will choose a partner based on a probability distribution that defines the weighted Euclidean distance in the n dimensional space between the individual and the future partner. The distribution function and the weights are set by the user. Usually a distribution will prefer a partner who has similar properties, namely one who is closer to the individual. Unlike the main simulation step, the relationship algorithm is not parallelized as of now, thus it increases runtimes significantly.

ASSESSMENT

In this section, we would like to provide some insight on how our solutions perform. However, the comparison of two different solutions can only be effective if both implemented the same base model. This means that comparison requires the exact recreation of a model. In most cases papers in the topic do not supply access to the dataset, since it is not relevant to recalculate the results, thus the models cannot be recreated for comparison in terms of runtime.

For this reason, we can only effectively compare our own two solutions. It is important to note though, that our goal is to introduce our solutions and the detailed description of the models is outside the scope of this paper. Moreover, we intend to use the two solutions to implement fundamentally different models that are still in the development phase. Thus, we implemented the same simplified model with both of our solutions.

We forecasted the population from 2004 to 2054 and compared the results to the summarized population numbers between 2005 and 2015, that are available in the public online database of the HCSO. The initial population is based on the micro census of 2005. The census is made in the middle of the year, thus death and birth numbers of 2005 would be incomplete, so we start the simulations from 2004 with the appropriate part of the data. We used the Lee-Carter method (Lee, 1992) to forecast the mortality rates and we modified the method for the fertility rates. Both parameter forecasts were based on the historic values between 1995 and 2004. Figure 1 shows the results of these forecasts. The SAS model underestimated the number of deaths and the Simulation Framework overestimated the small increase in the birthrates. However, both solutions managed to forecast the size of the population for 10 years with a maximum error of less than 1.5 percent. The difference between the result of the two theoretically identical models appears because of the different random generation of our solutions. This further supports our claim, that a fast forecasting is necessary to allow the comparison of multiple random seeds and configurations in a short time.

A pure birth and death model would not properly introduce the possibilities of our solutions, thus we included other modules in our tests. We implemented a simple pension and income module based on the model of Péter Mihályi (Mihályi, 2016). The incomes are fixed as the overall weighted average gross income in the micro census of 2005. The age when some individual starts working is 21 years and the retirement age limit is 65 years. Based on the original model the overall percentage of contributions from the gross income is 20 percent. Assuming, that an individual is retired for half as long as he or she has worked, and that the system works well – so on average everyone gets the same amount of pension till their death back as they contributed – the amount of pension is set as 40 percent of the gross income. We also included a marriage and divorce module. According to the statistical data on average 1 percent of marriages end with a divorce every year and 2 percent of singles marry annually. We chose to implement the mate choice so that only women can choose partners, and their choice is only dependent on age, so with the highest probability they choose a man of the same age and as the age difference grows the probabilities decrease exponentially. With these additions, we aimed to simulate a more complex model to present runtimes, but to simplify implementation we excluded the influence of these properties on the mortality and fertility rates, thus the additions did not change the results seen in figure 1.

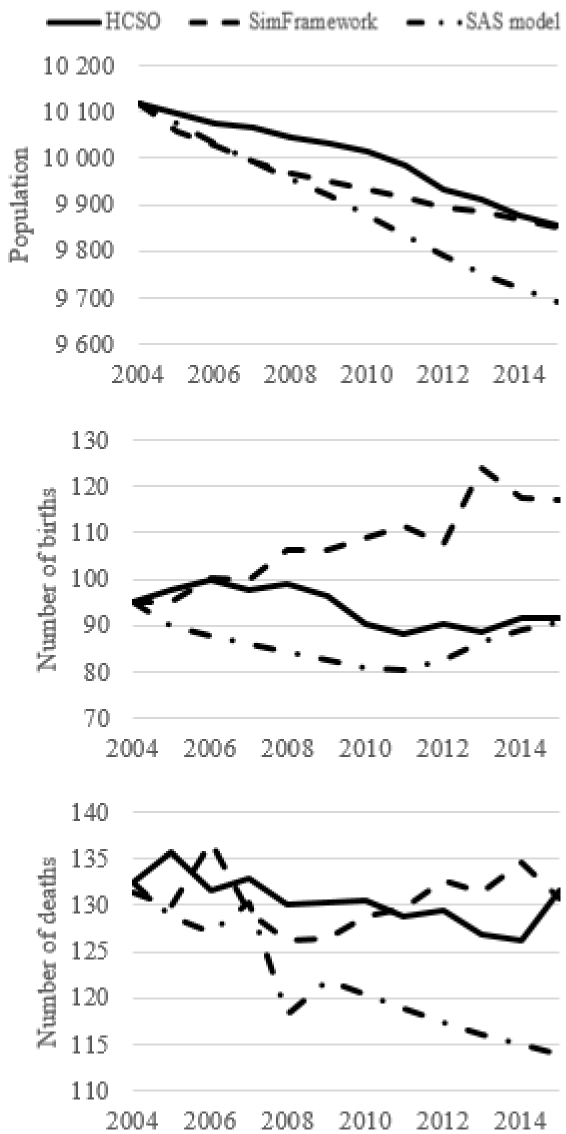


Figure 1: Simulation Results (in thousands) Compared to the Actual Values Between 2004 and 2015.

Table 1 summarizes the runtimes of the above described extended models. We showed the results with and without the relationship module, since it is the most resource heavy part of our solutions. The simulations were tested on a personal computer (Intel Core i7-3632QM CPU @ 2.20 GHz). The table shows the runtimes of the Simulation Framework for both sequential runs and parallelized runs with 4 cores. The speed up is basically linear and the relationship runtimes get simply added, since those are unaffected by parallelization.

Table 1: Runtime Comparisons of 50 Year Simulations

Model	Standard	With Relationships
SAS model	35:27	1:42:14
SimFramework (1 core)	40:18	49:06
SimFramework (4 core)	10:44	19:16

It is clear, that the Simulation Framework outperforms the SAS model, but it is important to note that the latter project is in a less developed stage. The SAS model is also yet to be implemented as a parallel solution to improve runtimes. Moreover, we implemented exponential smoothing to forecast the parameters, thus in the future, we would like to implement more sophisticated algorithms. The Simulation Framework is also in the middle of rework since many UI features (other than the block programming interface) are outdated. However, the most important future task is to improve the parallelization of the household module. We are continuously fixing bugs, improving the documentation and still actively developing the framework. The latest version to date can be found on GitHub [dburka001/SimulationFramework]. Since the SAS model is a business project, its code is not yet published, but can be discussed with developers via e-mail.

CONCLUSION

The continuous ageing of the society results in structural changes in the population of every major country. The changes create new challenges that the governments must face. Ensuring the adequacy and financial sustainability of the pension system is among the most critical challenges. The forecasting of the population is necessary to aid the decision-making process regarding the sustainability of the pension system. However, the development of demographic processes is slow, so only long-term forecasts can highlight the ongoing changes.

We showed, that the most appropriate method to support the decision-making process regarding the issues we raised is a complex dynamic microsimulation solution. We discussed the advantages of the chosen approach, and also analyzed the difficulties that prevented microsimulation to become a wide spread method for demographic forecasting. We proposed two solutions that aim to offset the disadvantages of microsimulation. We hope that in the future our systems can facilitate the use of dynamic microsimulation techniques in the field of demographic forecasting and allow us to collaborate with other researchers, economists, and demographers.

REFERENCES

Burka, D. 2016. “Supporting the Hungarian Demographic Pre-calculations with Microsimulation Methods”. *SEFBIS Journal*, 10, 13-23.

Cerami, A. 2011. “Ageing and the politics of pension reforms in Central Europe, South-Eastern Europe and the Baltic States”. *International Journal of Social Welfare*, 20(4), 331-343.

Csicsman, J., László, A. 2012. “Microsimulation Service System”. *Hungarian Electronic Journal of Sciences*

Dekkers, G., Desmet, R., Rézmovits, Á., Sundberg, O., Tóth, K. 2015. “On using dynamic microsimulation models to assess the consequences of the AWG projections and hypotheses on pension adequacy: Simulation results for Belgium, Sweden and Hungary”. *Federal Planning Bureau – Central Administration of National Pension Insurance*

Kulik, C. T., Ryan, S., Harper, S., & George, G. 2014. “Aging populations and management”. *Academy of Management Journal*, 57(4), 929-935.

Lee, R. D., Carter, L. R. 1992. “Modeling and Forecasting U.S. Mortality.” *Journal of the American Statistical Association*, 87(419), 659-671.

Majcen, B., Cok, M., Sambt, J., Kump, N. 2011. “Development of pension microsimulation model”. *Institute for Economic Research, Slovenia*

Merz, J. 1991. “Microsimulation – a survey of principles, developments and applications”. *International Journal of Forecasting*, 7(1), 77-104.

Mihályi, P., Vincze L. 2016. “A „Nők–Férfiak 40” nyugdíjkoncepció pénzügyi következményeinek szemléltetése a felosztó-kirovó rendszerben.” *Gazdaság és Pénzügy*, 3(1), 3–24.

Mohácsi, L. 2014. “Gazdasági alkalmazások párhuzamos architektúrákon = Business Computing and Parallel Architectures” *Doctoral dissertation, Corvinus University of Budapest*.

OECD 2015. “Pensions at a Glance 2015: OECD and G20 indicators”. *OECD Publishing*, Paris.

Orcutt, G. H. 1957. “A new type of socio-economic system”. *The review of economics and statistics*, 116-123.

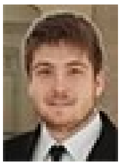
United Nations, Department of Economic and Social Affairs, Population Division 2015. “World Population Ageing 2015“ (ST/ESA/SER.A/390).

AUTHOR BIOGRAPHS

DÁVID BURKA was born in Nagykanizsa, Hungary. He acquired his degree in Business Informatics at the Hungarian Corvinus University of Budapest and is currently working on his PhD in the same field and institute. He obtained his second master’s degree as an Info Bionics Engineer at Pázmány Péter Catholic University of Budapest. His research focuses on simulation and modelling in the fields of demography, social choices, and neurobiology. He currently teaches at Corvinus University and works at the Sleep Oscillation Research Group of the Research Center for Natural Sciences.

JÓZSEF CSICSMAN was born in Budapest, Hungary and attended the University of Szeged, where he studied mathematics and obtained his degree in 1976. He worked for 20 years for the Hungarian Central Statistical Office before founding his private company, Calculus in 1996. From 2001 he was active in Budapest Technical University where he is now leading a Microsimulation research group in the field of simulation for Demographic and Economic changes, and different kind of Statistical Matching applications. He gives lectures on Practical usage of Data Mining and Statistics at BUTE and University of Szeged. His e-mail address is: csicsman@calculus.hu and his webpage can be found at http://calculus.hu.

LÁSZLÓ MOHÁCSI was born in Budapest, Hungary. He attended the University of Óbuda, where he obtained his BSc degree in Electrical Engineering. He acquired a MSc degree in Informatics at the Eötvös Loránd University of Budapest. He obtained another MSc degree in Biomedical Engineering in a collaborative program of the Semmelweis University and Budapest University of Technology and Economics, Hungary. He got his PhD in the ICT Doctoral School of the Corvinus University of Budapest, where he is currently working as a senior lecturer at the Department of Computer Science. His research focuses on the economic application of parallel computing and economic modelling. His e-mail address is: laszlo.mohacsi@uni-corvinus.hu.



BENJÁMIN SOÓS was born in Cegléd, Hungary. He is a student at the University of Szeged, and he is preparing for getting his BSc degree in Business Information Technology. Within the confines of a course about Statistical Softwares, he learned the use of the SAS language, and got acquainted with the tutor, József Csicsman. This relationship led to a thesis work with the title of “Demographic Forecasting with Dynamic Microsimulation”, and to a job at Új Calculus Ltd. As a subcontractor, he currently works at CIB Bank as a Junior SAS Developer. He has the enthusiasm for learning more about dynamic microsimulation techniques and developing a SAS based applications. His e-mail address is: soosbenji@gmail.com.