

Policy Optimization in Dynamic Models with Genetic Algorithms

Björn Grossmann
Graduate Student at the Department of Mathematics
Center for Modeling and Simulation
University of Hamburg

Mai, 10th 2002

Contents

- ABSTRACT 3**

- BACKGROUND..... 4**
 - Why use GA's for Optimization of Dynamic Models 4
 - On the history of genetic algorithms.....4
 - How do GA's work? 4

- APPLICATION TO THE ISIS - MODEL..... 6**
 - The ISIS-model 6
 - Objective functions 8
 - Preparations of model for optimization..... 9
 - Policies 10
 - Results of optimization 11

- OUTLOOK..... 20**
 - Further application of method..... 20

- APPENDIX 21**
 - Conversion of Stella models to C++ for use with GA's 21

- REFERENCES 25**

Abstract

Dynamic models are often built in order to evaluate the influence of policies on a complex system. Results can help organizations or politicians take appropriate decisions. In the model, those decisions are represented by a set of parameters, referred to as “policy parameters”. With the help of an objective function, it is possible to determine whether chosen values of the policy parameters influence the model’s outcome favorably or unfavorably.

Finding policies that optimize (i.e. maximize or minimize) a given objective function by manually trying different parameter values may work fine with small models and just a few parameter values. However, as the number of parameters that can be simultaneously modified increases, or as we work with complex, non-linear models, we may never find out which values we should choose as to reach a global optimum.

We need a good algorithm to find them. Many traditional optimization-algorithms are of no help, as they require explicit gradient information or continuity of the objective function which often is not available in non-linear models. Gradient algorithms, for example, typically fail once they have reached a local optimum, which is like a peak in a mountainous area but not necessarily the highest peak. It might as well be the lowest peak. Trying to find good values by combining different values is not a good choice either, as it may require years of calculation. Genetic algorithms however do work with the conditions we have. All that is needed for them to work is a way to code sets of policy values (in our case as an array of real numbers, each of which corresponds to a policy value) and an objective function. Genetic algorithms, if properly implemented, do not easily mistake local optima for global ones. So we decided to use genetic algorithms to optimize models.

We decided to combine System Dynamics models which are build in the Stella modeling environment with genetic algorithms. This allows use of the graphic capabilities of Stella for modeling. Stella has an option to save the model equations as a text file which can be read by other programs. Stella can provide these equations in the order of execution which makes the task much easier. However, the text file produced by Stella has its own syntax; it is not compatible with any of the better known programming languages.

Therefore, a tool was developed that transforms Stella’s difference-equations into C++ code, adds code implementing the objective function as well as code specifying which variables to use as policy variables. Both, the objective function and information on which policy variables to be used is provided by the user in a separate file. Compilation and execution of the generated code is done automatically. A run of the dynamic models which I have selected for this research may take up to 20 seconds with Stella. These models are fairly big because they are used for consultation and research and have to have a high degree of detail so that they do provide rich useful policies in an adequate representation of socioeconomic-environmental systems. The genetic algorithm may have to run such a model several thousand times before finding an optimum. Therefore, the tool has to produce code that executes very fast.

The results of the optimization run are written to a file and can be visualized as graphs.

The results allow to gain very thorough knowledge of the problem treated in the model. Often, particular behavior of the system is found only from the results given by the genetic algorithm.

Background

Why use GA's for Optimization of Dynamic Models

Huge and complex dynamic models are often difficult to “validate” and evaluate (validation is difficult for System Dynamics models; it is preferable instead to use a broad repertoire of methods to “build confidence into System Dynamics models” as it was stated by Jay Forrester. With this tool I contribute more tools to this repertoire of methods. We may have certain objectives that we want to optimize. Different variables can be modified to reflect choice of policies. But manually evaluating the model for different sets of policies can take a long time and may miss the best set of parameter values. It may not even give a good result, although such a result is possible. As was said before, even when traditional algorithms, such as “hill climbing” algorithms are applied, a local optimum is often mistaken for a global optimum. Due to the nature of complex models, preconditions required for algorithms, such as continuity of the objective function are often not met. Genetic algorithms however, do not ask much of the objective function. They very rapidly get a broad overview of the whole solution space. Genetic algorithms can be applied to many types of problems, including combinatorial problems, such as finding an optimal strategy for the game “Solitaire” or for solving traveling salesman problems as well as for search and optimization problems. They usually find good solutions pretty fast, but may take a little longer for the “fine-tuning”.

On the history of genetic algorithms

Not that long ago, in 1859, Charles Darwin published his assumption in his work “On the origin of species by means of natural selection or the preservation of favored races in the struggle of life” that the fittest individuals of a population are most likely to survive and pass on their properties to the next generation. Claiming this, he founded evolution theory. At this time he did not know much about inheritance let alone about genes. In 1865 Gregor Mendel published his discoveries from working with peas with differently colored blossoms that later allowed to conclude the existence of genes. Mendel was not appreciated at that time. 16 years after his death De Vries, Correns and von Seysenegg independently repeated Mendel's discovery and only later found Mendel's work. Mendel had not been aware of the phenomenon of mutation which means that sometimes an individual is qualitatively different from its parents. Flemming discovered chromosomes which was the beginning of classical genetics.

Scientists are often amazed by the elegance of solutions nature provides. This was also the case with genetical inheritance and Darwin's theory of the survival of the fittest. In 1970, John Holland, at the university of Michigan, together with his team, first applied Darwin's theory and knowledge gained in genetics for mathematical optimization and invented what today is known as “genetic algorithms”.

How do GA's work?

At the start of the GA an initial population $P(0)$ is chosen, which consists of n individuals. The choice of the initial population is often done randomly. Each individual is represented by a genome, which can be an array of real numbers. In our case, the genome corresponds to a

set of policy values. Often genomes are coded as a sequence of 0's and 1's also called binary string. Each genome is assigned a *fitness* value, which is related to the *objective function* value of the genome. In maximization problems, the value returned for an individual by the objective function is often used as the fitness value, in minimization problems the reciprocal of the objective function value can be chosen. In the case of a dynamic model, the objective function can be something like the average unemployment ratio which we aim to minimize. It needs to be evaluated for separately for each individual, as most individuals have different genomes. Each evaluation requires a complete model run¹.

Once the fitness is evaluated for all individuals, the *reproduction operator* is applied to the population. It picks individuals from the population and puts one or more copies of their genomes in the mating pool. Individuals will be chosen with a probability proportional to their fitness. This means, that if f_i is the fitness of individual I_i and $f := \sum_{j=1}^N f_j$ is the sum of the fitness of all individuals (where N is the number of individuals in our population) the probability for selecting the individual I_i is $\frac{f_i}{f}$.

Next, the *crossover operator* is applied. Crossover usually involves two genomes. This means an exchange of parts of the genomes. If genomes are represented by strings, this would correspond to randomly cutting both strings at the same position and exchanging one of the two segments.

Next, the *mutation operator* is applied. Treating one genome at a time, it alters the state of individual bits of the binary representation of the genome, changing 0's to 1's and vice versa. Note that mutation only occurs with a small probability but can be vitally important for finding better individuals. In the case where all genomes have an identical value a specific position, this is the only operation that can alter it. Now we have found the next generation and can start again.

The algorithm needs to know when to stop. How can it know that it has found the results we are looking for? One criteria that is often used, is, to have the algorithm stop after a limited number of generations. The number of generations needed, depends on the problem. In the case of the ISIS model the best individual was found after about 100. Alternatively, we may decide to use convergence of either the genomes or the values of the objective function to determine when to stop.

The pseudo code in Listing 1 shows the steps used in GAs:

```

procedure evolution program
begin
  t ← 0
  initialize P(t)
  evaluate P(t)
  while (not termination-condition) do
  begin
    t ← t + 1
    select P(t) from P(t - 1) // reproduction
    recombine P(t)           // crossover and mutation
  end
end

```

¹ In the case of the ISIS model, model run starts at 1980, but policies are only applied after 2001, as we cannot alter past decisions. This allows us to calculate the run from 1980 to 2001 just once and use the state of variables for later optimisation runs starting at 2002 hence reducing required calculation time. Searching for policies we could have applied in the past can help understand what we could have done better, but was not our main focus.

```
        evaluate P(t)
    end
end
```

Listing 1: Structure of genetic algorithms

Excellent genetic algorithm libraries exist that may be used in your own programs. I chose to use GALib written by Matthew Wall at the Massachusetts Institute of Technology for use with optimization of models². I'd like to thank the author for the excellent work he did!

Application to the ISIS - model

The idea of using GAs to optimize dynamic models has been applied to the ISIS model, which is a typical complex dynamic model. It contains more than 50 variables that can be modified to reflect chosen policies. Although all variables could be modified, it is usually enough to start with those that are in the focus of present political or scientific discussion. A GA allows a wide and systematic exploration of the merit of new, unproven policies. Like most complex models, the ISIS-model sometimes behaves counter-intuitively, which means that it exhibits behavior which is counter to the initial expectations but represents likely reaction of the real world – only you did not expect this but something quite different. Finding counterintuitive behavior is highly valuable because it helps to avoid detrimental decisions which could easily be taken due to wrong expectations.

The ISIS-model

The “ISIS”(Information Society Integrated Systems)-model is the result of several large integrated interdisciplinary projects on the evolution of the information society and its interactions with the ecologic environment. These groups were led by Dr. W.D. Grossmann³. This model is now at the base of a wider collaboration with groups from different areas of science as well as with consultancy groups. For example, together with the Hamburg Institute for International Economics the goal of development is to combine their econometric national model with the ISIS model, as econometric models are not good at dealing with structural change whereas ISIS is build to analyze the present large scale socio-economic transformation in an information society with a new, information-rich economy. Together with the IPRC (International Pacific Research Center) and the GKSS Research Center Geesthacht a first attempt is done to find out how to use the present socio-economic transformation to simultaneously and proactively adapt to possible climate change. Mathematically, ISIS is a complex system of non-linear differential (or difference) equations.

The ISIS-model shows the interaction of economy, ecology, human life and work and the transition from traditional, information-poor, resource consuming industry (called “Mature Industry”) to the so-called “New Economy”, which characteristically uses huge amounts of information and communication for and in its products and services. The goods which the new economy produces could consume far less energy and resources than present material goods of the mature industry. In parallel, there is an increasing offer of non physical goods by this new economy, such as knowledge, information refinement or manifold highly dematerialized services, e.g. the so-called experience economy. However, this new economy

² GALib can be found at <http://lancet.mit.edu/ga>

³ The discussion of the ISIS-model is based on [3] and [4]

could as easily consume larger amounts of energy and of material resources than the present economy, because use of more information allows to access more resources and to build a wider variety of highly appealing resource-intensive products. Therefore, it is interesting to study these interactions with the goal of finding optimal policies which naturally gives a big model. Much of the development effort in this model went into making initial models much smaller to make them manageable.

Central for employment, unemployment and structural change is company formation and growth. This is encouraged by a multitude of locational factors. Many of these factors differ very much from those of the mature industry. E.g., the new economy needs an appropriate information (or telecommunication) infrastructure characterized by high-speed data-connection at low prices. It also needs a supportive social environment such as openness for new ideas or friendly public administration, such as “one stop bureaucracy” where the permit for a new company is processed within one day. European bureaucracy often needs months for this task. There are many policies which a region can implement if it wants to succeed in this new economy which are represented in the ISIS model by about 25 parameters. Application of the model for a specific region needs an initial analysis of regional strengths and weaknesses to find out where to start with the process of devising optimal suitable regional policies.

Old economy, as well as new economy are each divided into 7 phases of development called (e1) ... (e7) (economy phase 1, and so on, up to economy phase 7), or “new economy 1” (ne1) up to “new economy 7” (ne7), respectively, which represent very different states in the evolution of a so called basic innovation (such as, historically, the steam engine or now the new information and communication technologies). Historic evidence shows that each of these phases has an average length of about 7 to 8 years although this is a very idealized picture. In reality, there are several phases simultaneously, interacting, overlapping, and a new invention going back and forth in these phases (a good example is the attempt to build computer software which connects most common platforms be they mainframes, PCs, organizers or mobile phones, or the chip in the pizza which interacts with the oven to get the right treatment.

Once an idea for a new product or service is born(e1), a company may be founded (e2). A new basic innovation needs what for them is “Key-People” with appropriate new know-how, enthusiasm, persistence and perseverance. As we know from the extensive company formation in the U.S., a garage may suffice for a start-up company. First products become acceptable and are sold. But, when the innovation is really new and will become very common, it needs a third phase where it is adapted to the outside world (the car got lamps, good brakes)and where the outside is adapted to the new basic innovation (gasoline did no longer come from the pharmacy in bottles but from gas stations, and a road network became necessary and so on). This interaction with the outside world is, to a large extent, a political and social phenomenon and hence takes time and patience. Therefore, with an average duration of 7 years, the new product may have established its place on the market (e3). This is the groundwork for the following large scale expansion, if the new innovation is really great, and now companies can really and rapidly expand to global dimensions (e4). It may eventually become a global player (e5) and management may become used to the company being successful and reaction to new requirements may become really slow. This is unfortunate as competition becomes global in phase 6. The final phase e7 is that of - for some time - exceptionally high incomes, based on a very well known basic innovation which no

longer evolves. Within a very brief period of time this phase may end in what Laepple⁴ calls the “sclerotic milieu”, that frustrating stage of low creativeness, obsolete knowledge, low willingness to begin something really new and hopelessness. Examples were ample throughout most of the developed world in the former shipbuilding and steel processing regions.

As a young company grows, it changes its character dramatically from phase to phase which usually has the effect that its founders and first staff members become alienated and leave, or are removed or put on some good looking but irrelevant position. Also, almost all companies disappear through many processes, in particular through fusion with other companies. The statistics are that 3 million companies are started per year but only 30 per year make it into the list of the Fortune 1000.

This division into 7 phases is found in mature as well as new economy. The key people have to be good in each particular phase which implies that their capabilities and preference are also very different depending on the phase - they also follow this transition, or, as we do not know what is driving and what is driven - the companies follow the changes of their key people, not to forget interactions with a volatile and creative market in new products. Also the know-how follows this pattern from radical and unproven new to immovable, extensive, and stagnant in the last phase. Again this is due to very close interactions between know-how and companies, or key people, respectively. ISIS models interactions between economy, human attitude, knowledge and the wider environment. One submodel deals with population as we want to deal with employment and unemployment. Regenerative behavior and creativeness are represented in appropriate age classes (eight classes ranging from new-born to old age including children, teenagers, apprentices, ...). The model also includes land use divided into seven categories, in particular settlement area, agricultural land, forest, infrastructural land, protected areas, parks and gardens and a new concept, the so called urban nature sanctuary. Quality of life is calculated for different groups of the population, based on those widely discussed parameters such as income levels, density of population, land use and its corresponding value for recreation as well as climate conditions. One section describes the market for new and old economy based on supply and demand. For both new economy and mature industries (the equation for) economic growth depends on the value of demand for the respective products of that economic sector. A major driving force merits particular details, the so called information potential. Here the “classical” laws are put together, in particular Moore's Law on the speed and price of chips, Gilder's Law on the speed and price of information networks, Metcalfe's Law on the capability from interactions of such aggregates and so on.

Further documentation of the model and the corresponding ideas can be found in [3] and [4]

We will now describe the variables that are evaluated in the objective functions:

Objective functions

- unemployment ratio: minimizing unemployment is a major political aim. It is interesting to look at this on a long time scale as well as for specific dates, such as the time of the next election. The political opposition party may wish unemployment to reach a maximum so that people will most likely vote for a change. The present government may want a timely minimum so that people want to keep their present government.

⁴ Dieter L  pple, TU-Harburg, department of rural economics and reg. development

- quality of life: quality of life has become a major political aim in many regions. The California Economic Strategy Panel states: “Quality of life for employees is the main reason companies are here”. We can test policies that aim to increase quality of life just for specific groups so as to maximize overall welfare indirectly, for example by attracting new key people by exceptional quality of life according to their understanding. This would, simultaneously, and to be tested with the model, maximize economic growth.
- number of people employed in new economy: we may want to maximize this number, as new economy usually grows more rapidly and has a longer productive life expectancy than mature industry. Another reason may be that new economy often uses less resources and less energy than mature industry and people employed in this section may have “safer” longer lasting jobs.
- urban nature sanctuary: A new concept was developed for the new “Cyber cities” in Malaysia to reconcile man and nature, or, more prosaic, to attract those “crazy new wilds” who start new companies in the high-tech section of Cyberjaya. More idealistic: most people certainly would agree that it is great to preserve beautiful areas in nature. It seems however, that a lot of people appreciate personal advantages such as high salaries or high income more than nature, at least on a short term. So, it will be quite interesting and vital for our survival, to see, if we can find policies so that we can have both.
- maximize size of new economy
- (minimize) size of mature industry: Some people seem to appreciate high energy and resource consumption (such as those politicians who represent old industry) but most of us should be happy if mature industry could be minimized to mitigate “Global Change” and the “Green-House” effect.
- settlement area: if housing gets expensive, it may be a good idea to maximize its availability for immigrating new key people. However, one should carefully watch not to waste nature’s beauty to a degree that quality of life declines with the effect of repelling instead of attracting these people.
- infrastructural land: is favorable for mature industry, often repelling for new economy and not liked by new key people. This causes non-monotonous evolution of optima which poses a difficult optimization problem.

These considerations define a frame within which we have to optimize. Having defined suitable objective functions we would like to find variables that may be modified by our GA so that we get satisfying policies.

Preparations of model for optimization

The model contains several sets of parameters each set influencing one specific aspect of the model. For instance, the term “NE 123 vitality from locational factors” (vitality of new economy caused by locational factors) is defined as the product of “NE KH location mod” (availability of new know-how depending on location) and “Econ key cond 1” (key conditions for economy), which themselves are the product of other variables as shown by Figure 1.

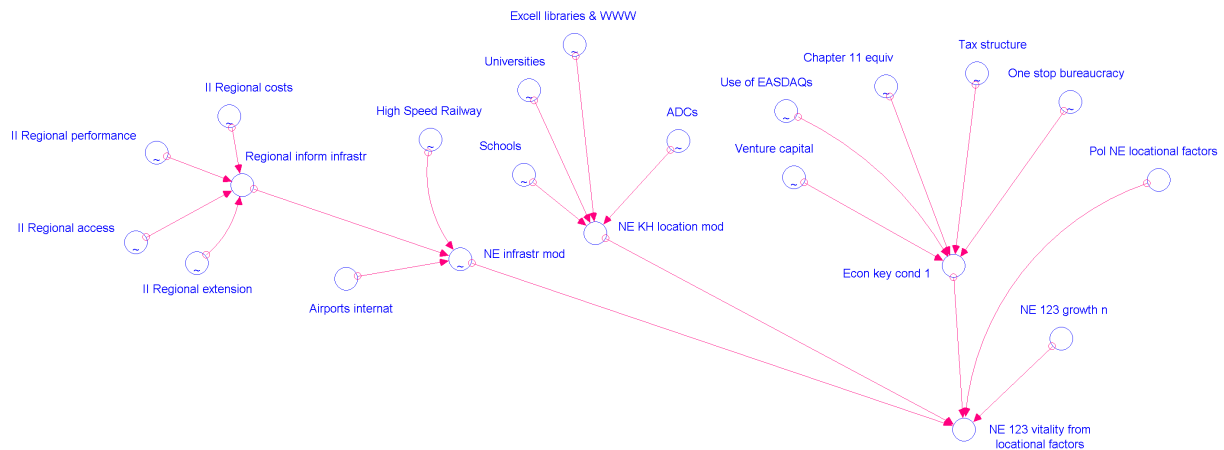


Figure 1: policies are often combined with multiplication

To make the optimization computationally feasible we have organized all variables with comparable effects into a few multiplicative terms. Hence, if one individual variable is modified (say it is multiplied by 2), the product would directly reflect this change (becoming greater by a factor of 2).

In this way we have introduced auxiliary variables into the model which are less realistic than the original ones but which can more effectively be evaluated by the GA. Once a desirable value for one aggregate variable is found, it is our job to retranslate it into the original variables to find the corresponding policy.

Policies

For different clients we have a look at suitable variables that they could influence. The GA is allowed to modify those variables within suitable ranges that appear reasonable.

- variables influencing foundation and growth of new economy:
 - o regional infrastructure (extension, access, performance and costs)
 - o international airports and high speed railway
 - o know-how (schools, universities, excellent libraries, ADC's (Advanced Development Centers)
 - o key conditions such as venture capital, use of new stock markets for small companies e.g. the EASDAQ in Brussels, Chapter 11 – equivalent laws on bankruptcy, fiscal law improvements, one stop bureaucracy
- variables influencing quality of life:
 - o quality of life for key people of the new economy
 - o quality of life from nature
 - o quality of life from spare time
 - o quality of life from ratio key people / population
 - o quality of life from key people new economy jobs
 - o quality of life from creative climate
 - o quality of life from housing
 - o quality of life from climate, traffic, pollution, divergence of agriculture
 - o quality of life from material standard of living
- variables influencing growth of new economy:

- policies for increase of number of key people for the new economy
- programs for immigration of key people for phases 1 to 3 of the new economy
- programs for immigration of key people for phase 4 of the new economy
- variables responsible for increase of know-how for new economy
 - policies for increase of know-how
 - improving quality of universities
 - extend training for employed population
- variables influencing birth-rate

Results of optimization

It is quite interesting to have the genetic algorithm do both optimization and what I will call “pessimization”. If the desired goal in the objective function is to minimize something, e.g. costs, then pessimization would mean to search for policies that do the opposite, that means maximize the objective function, in other words, maximize costs. Having knowledge of the minimal and maximal values allows us to tell in which range the results of policy strategies will be. In the example of unemployment ratio, which we will further discuss in this section, optimization means finding policies that minimize unemployment and pessimization means looking for those that maximize it. How can we increase, within the feasible range of our policies, unemployment as much as possible? What is the worst possible policy? Will unemployment, in spite of our worst efforts, nevertheless decrease? Does it stay as is or will it even increase? The specific value of doing both, optimization and pessimization, is that these two “bracket” the range that is accessible with our policies. Once we have this range, we can immediately judge on any specific policy, after it has been optimized with the GA, how good or bad it is.

Figure 2 illustrates the results of minimization of unemployment ratio starting in 2002 (0.2 on the y-axis means 20% unemployment). Unemployment is an important issue in Germany as we still have 4.3 million unemployed people which is about 12% of the workforce. The genetic algorithm shows us that by choosing the right options we may, according to the ISIS model, get rid of unemployment within only three years, or, if we chose the wrong options, may even increase unemployment. This is a theoretical result as we do not yet have costs associated with these policies. Efforts may be made in co-operation with the Hamburg Institute of International Economics to specify such policies in more detail and calculate their costs. In that way they become more practical.



Figure 2: minimization vs. maximization of unemployment

Now, which options should we take according to the results of the GA? Table 1 shows the values obtained for individual policies.

Policies for ...	Optimization	Pessimization	Range
... birth increase	2.5	0.5	0.5 – 2.5
... improving quality of life for new key people	2.5	0.5	0.5 – 2.5
... immigration of key people for phases 1,2 and 3	5	2.79	0.25 – 5
... immigration of key people for new economy 4	0.1	3.67	0.1 – 4
... increase of new key people	4	0.25	0.25 – 4
... for increase of new know-how	4	0.25	0.25 – 4
... locational factors	5	0.2	0.2 – 5

Table 1: Policies found by application of the GA

It is interesting to note what this says. Encouraging people to have more children is amongst the policies that can help decrease unemployment. The authors of the ISIS model were surprised by this result and will further analyze it. This is an example of an option one may never have tried manually. It is only due to the application of the genetic algorithm that the influence of an increase in the birth rate on unemployment becomes apparent.

Improving quality of life for new economy's key people is a good choice, too. Immigration of key people has been found to help decrease unemployment, as long as they have the skills required for phases 1, 2 and 3. It is interesting to note, that immigration of key people with skills for phase 4 does not seem to help decrease unemployment. This was a counterintuitive result, but is appreciated by the authors of ISIS as too much economy in phase 4 too soon suppresses further radical innovation which is the fuel for a long-lasting economic upswing based on the new basic innovation. Increasing education and training for new key people for phases 1, 2 and 3 seems to be an important issue as well.

We will have a closer look at what those variables do individually. We may find that it is enough to focus on one or two, in order to receive good results or we may see that all of them are equally important. Setting a policy's value to 1 corresponds to no specific action taken, to continue as we did before. So we will set all policy values to 1, except for the one we are evaluating individually.

We will start by evaluating what influence immigration of key people for new economy 1, 2 and 3 does have on unemployment. This option is often discussed in Germany. Some say this is a brilliant idea and will create new jobs, others fear that it might further increase unemployment. In fact, the ISIS model suggests that it depends on the qualifications of those we invite to immigrate. Figure 3 shows us that immigration of key people for phases 1, 2 and 3 indeed helps to decrease unemployment. The method of having both, optimization and pessimization as a reference allows to compare the effect of this policy to the worst case (plotted in dashed style), to the default behavior if we continue as before (dash-dotted line) and to a combination of all optimal policies (dotted line). We see that if we do nothing but encourage key people 1, 2 and 3 to immigrate, unemployment will disappear about two years later than with the best possible options. So this policy really seems to play an important role.

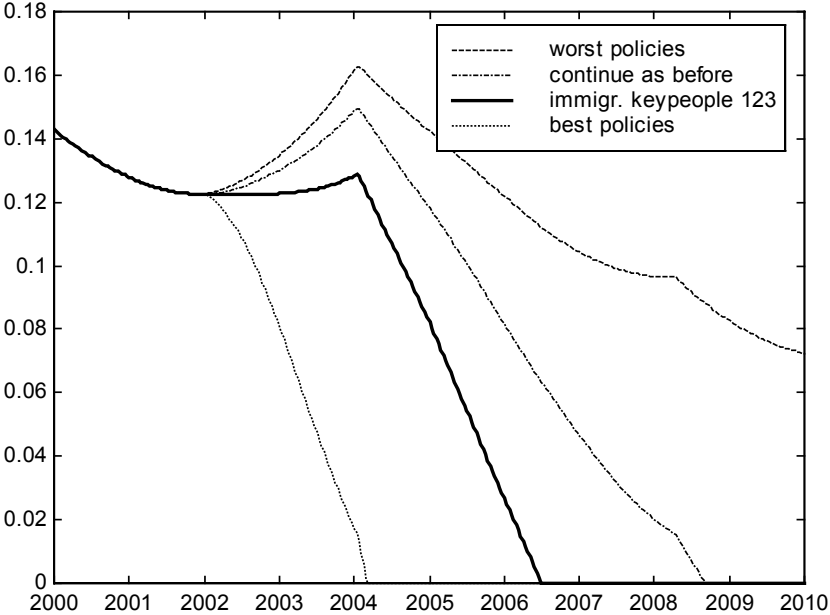


Figure 3: Unemployment decreases if key people for new economy phases 1, 2 and 3 are allowed to immigrate

Immigration of key people for phase 4 may not be helpful in order to decrease unemployment, as shown by Figure 4.

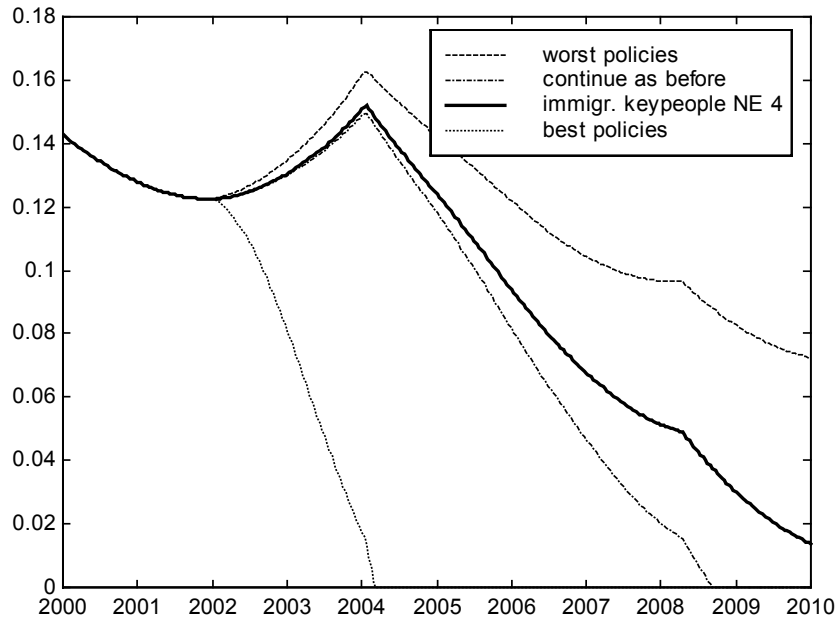


Figure 4: Immigration of key people phase 4 is not beneficial for decrease of unemployment

Another option would be to increase education and training of key people for phases 1, 2 and 3. Figure 5 shows that it would only take us one year longer to get rid of unemployment. We increase education and training programs by a factor of four, which may not be too easy, as teachers need to be found as well as people who are willing and qualified to undergo this training.

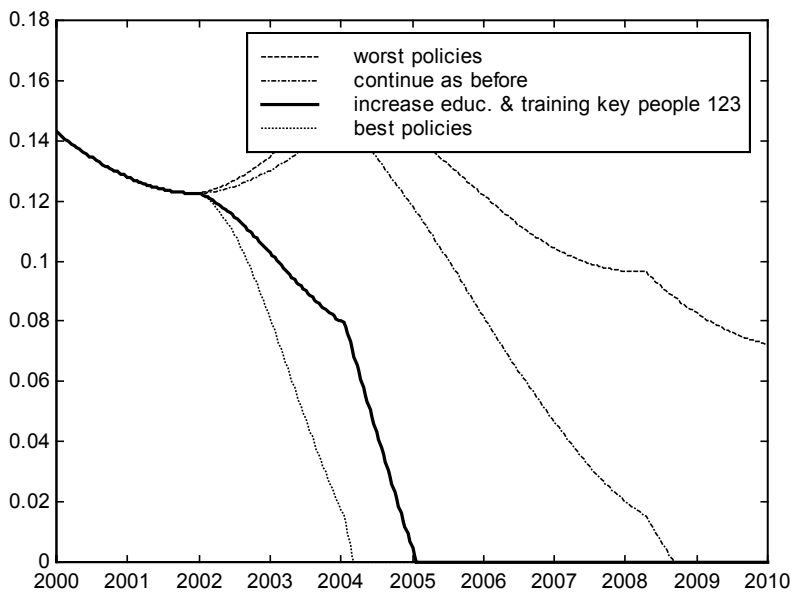


Figure 5: Education and training for key people 1, 2 and 3 is beneficial for decrease of unemployment

If we combine those two policies by encouraging immigration of key people and running education programs and training, we will get very close to the optimum, as shown by Figure 6:

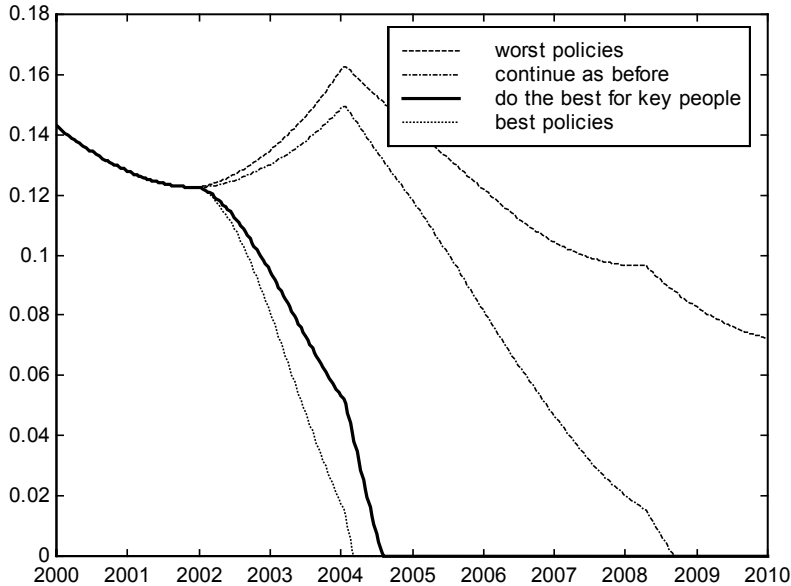


Figure 6: combining immigration and training programs gets us close to the optimum

We have not yet looked at the effects of applying policies for birth-increase, policies to improve quality of life for new key people and policies for improvement of locational factors individually, one by one. shows that each one of those policies does change very little on its own if applied exclusively.

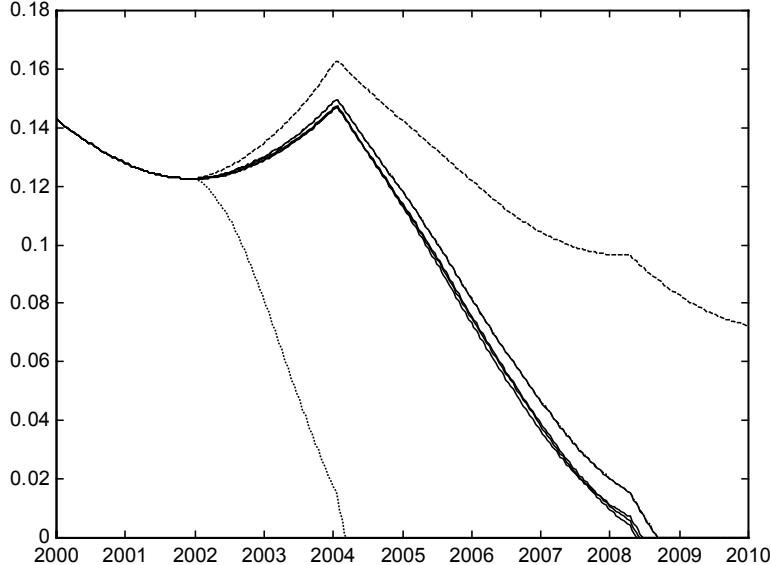


Figure 7: applying policies one by one does not change much

Without support for key people training and education or immigration of key people 1, 2 and 3, those policies are not even of great influence if applied in combination, as Figure 8 shows:

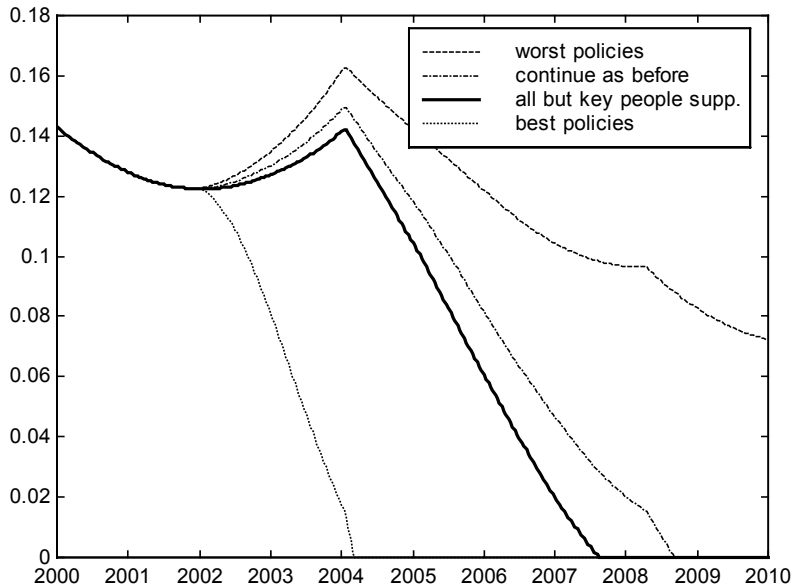


Figure 8: a combination of all policies except training and immigration of key people

We defined several objective functions for optimization. It is interesting to have a look if the policies required for optimization of individual objective functions are compatible with each other or whether they are in conflict and we need to decide between them, say either protect the environment or decrease unemployment. Fortunately and surprisingly, most objective functions are compatible. But there are policies as well that bring conflicts, e.g. increase immigration of key people for new economy phase 4.

The next task was a data analysis to determine the correlations between objective functions and policies. This was done with the software package “R”, a free software package for mathematical statistics.

Analysis is based on a matrix containing the results of the genetic algorithm, policy values found to be optimal for the corresponding objective functions, as shown in Table 2. Note: “BirthIncr” is an abbreviation for policies that help increase the birthrate, “QLNKP” stands for policies increasing the quality of life for key people in the new economy, “KP4Imm” means encouragement of immigration of key people for new economy phases 1, 2 and 3, “KP4Imm” is the same for phase 4, “NKHIncr” means policies for increasing know how needed for new economy, and “LocFac” stands for improvement of locational factors.

	BirthIncr	QLNKP	KP123Imm	KP4Imm	NKPIncr	NKHIncr	LocFac
Unemployment	2.5	5	4.0	0.100	4.00	4.00	5.0
empl. new economy	0.5	5	0.1	4.000	0.25	0.25	0.2
impr. of infrastruct.	2.5	5	4.0	0.102	4.00	3.90	5.0
quality of life	0.5	5	0.1	1.540	0.25	0.25	0.2
settlement area	2.5	5	4.0	0.100	4.00	4.00	5.0

Table 2: policy values found for objective functions.

The left column contains abbreviations for the objective functions, the other columns contain the values required in order to obtain the optimum. For instance, the line (2.5, 5, 4.0 0.100, 4.00, 4.00, 5.0) are the policies found for minimization of unemployment. See also Table 1 on page 12.

We have 7 policy values associated with each objective function. If we could visualize a 7-dimensional space, we could mark the position of policy values corresponding to optimization of each objective function. We cannot visualize a 7-dimensional space, but we may calculate the distances and plot a projection on the plane that preserves distances the best it can.

Table 3 shows us those distances:

	unemployed	employment new	Improvement of	quality of
empl. new econ.	9.3	economy	infrastructure	life
impr. of infrastr.	0.1	9.2		
quality of life	8.5	2.5	8.5	
settlement area	0.0	9.3	0.1	8.5

Table 3: distance of the policy sets corresponding to objective functions

Huge numbers indicate a wide difference in the policies required to achieve optimization, small numbers show compatibility of policies. For instance, if we compare the policy values required for optimization of settlement area to those required for a decrease of unemployment, we obtain a value of 0. And in fact, as we can verify in Table 2, the corresponding values are identical. If, however, we want to increase the number of people employed in new economy, this requires a set of policies that has a distance of 9.3 to the set of policy values corresponding to the fastest decrease in unemployment and 9.2 to optimization of infrastructure and 2.5 to quality of life and 9.3 to settlement area. In fact, this can be verified in Table 2 and is also shown in Figure 9. Correlation can be found when data points group around the diagonal, as is the case for unemployed and settlement area or unemployed and infrastructure, but not with unemployed and employed in new economy.

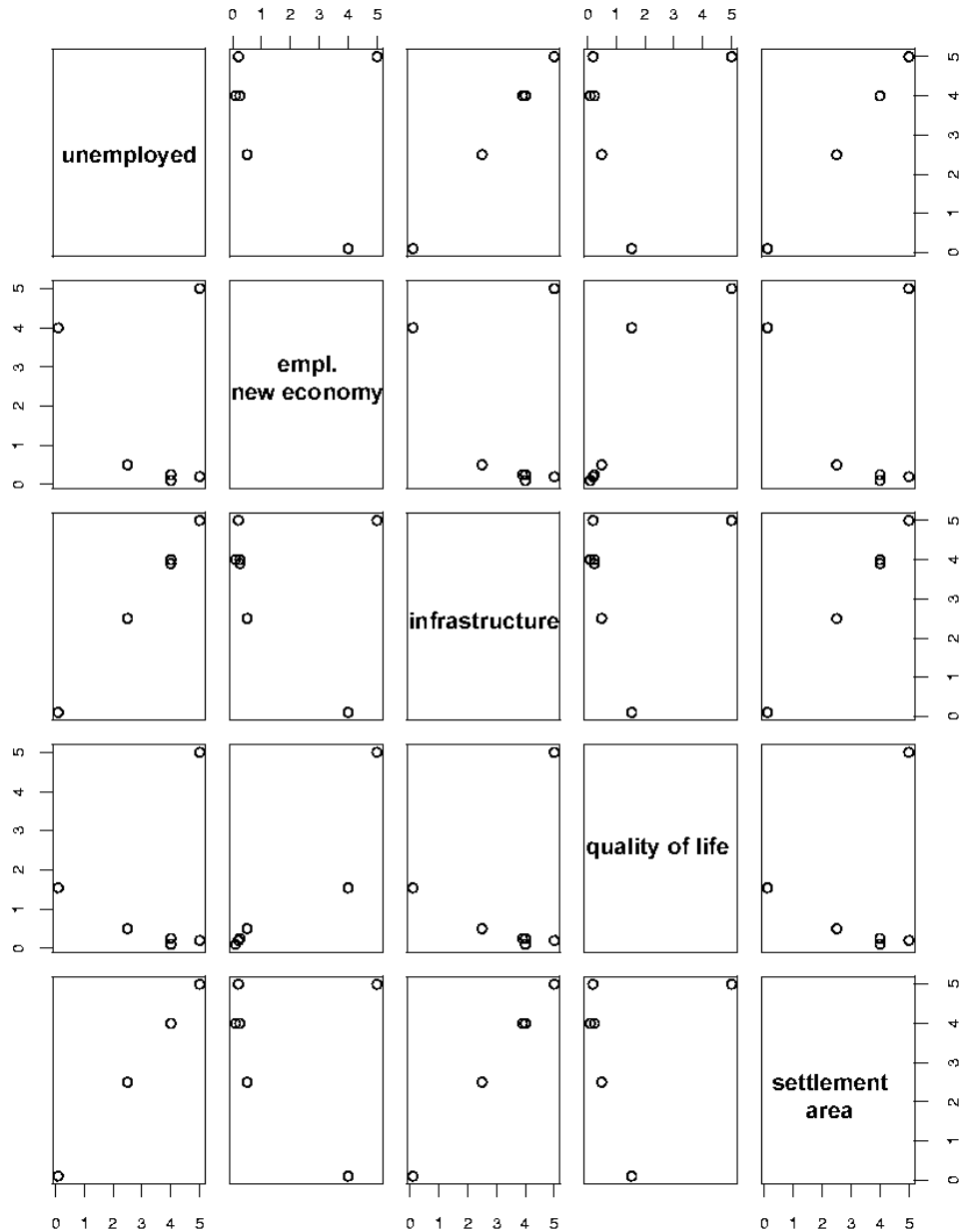


Figure 9: correlation of objective functions

As mentioned above, this 7-dimensional space can be projected on the plane. Figure 10 shows a projection that best preserves differences. Unfortunately we can not read much of the text, as most policy values overlap. But we may tolerate this, as it means that most policies are compatible with each other. One exception, as mentioned above, are the policies needed in order to increase employment in new economy.

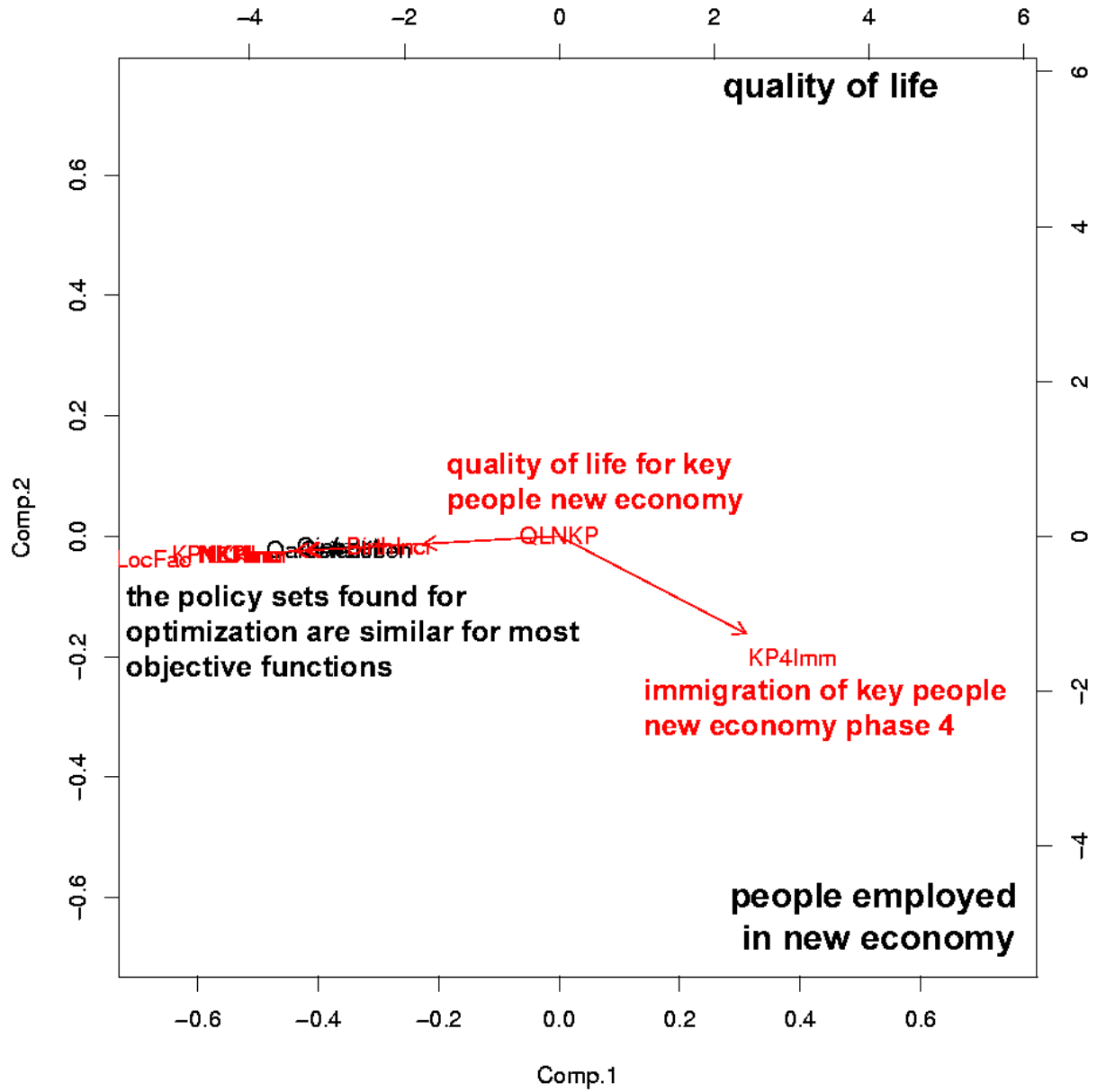


Figure 10: biplot of objective functions and policy sets

Outlook

Further application of method

While working on the application of GA's to dynamic models, some other ideas for application became apparent. A cost-benefit analysis for instance could enable us to find the best possible options given a limited budget. This could be done by introducing a “fitness function” that uses the results of the objective function but adds associated costs. Care should be taken to find the right balance, as putting too much weight on the costs might prevent the GA from finding improvements. Costs will have to be analyzed carefully. In some cases the costs of a policy do not grow as a linear function of the policy, but faster or slower. Take for instance the training of people: It is probably easier to give an initial education to untrained people and thus cheaper than refining the knowledge of experts where you need even better teachers and more individual teaching. The contrary may be true in other areas, for instance when installing infrastructure. Connecting a house with just one telephone line may be expensive, but if you install 20 lines it will cut the price of one line noticeably.

Another interesting option would be to introduce insights gained in game theory to the algorithms used for optimization. In the case of the ISIS model it became clear that different politicians are pushing different policies and are often judged by their electorate mainly on the results in their specific domain and not on the results on a global level. One example would be the sale of the licenses to use UMTS in Germany. Licensees had to pay a sum of about 40 billion US Dollars. It was considered a great idea of the secretary of finance as it seemed that he had found a way to reduce the debts of Germany. When working on the optimization of ISIS, improvement of information infrastructure was chosen as a policy variable. It was found that information infrastructure is beneficial for economic growth and for reducing unemployment. Could the strategy of the secretary of finance turn out to be a mistake? Could it be that it does more harm to economics on a long term scale than it helps at first sight? In fact, the resulting loss caused by a retarded start of UMTS may far exceed the exorbitant tax. As the combination of dynamic models and a good GA has become highly operational and as the ISIS model is being used in a growing number of co-operations the possibility emerges for further development of this combination.

Appendix

Conversion of Stella models to C++ for use with GA's

As mentioned before, the ISIS model is available in Stella / iTHINK. It needed to be transformed into C++ code. This was necessary to improve speed as the model must be run for every individual set of policy values for evaluation of the objective function. And it was done, of course, to be able to connect the results with the genetic algorithm and to control it as needed. Fortunately, Stella allows to export the underlying difference equations to a text file. This is the foundation of the required C++ code, as we would not want to rewrite the whole model! Instead, I wrote a program that reads the Stella equations, parses them and generates a C++ program from them complete with the logic required to run a model. This is not an easy task, as all expressions needed to be expanded to a syntax tree in order to be able to do the conversion. Take for example an expression like

```
x = IF a:1 < (IF n = m THEN o ELSE p) THEN pulse( -max(10e-7,b*2) ) ELSE c
```

It needs to be transformed to an expression like

```
x = (a_1 < (n == m ? o : p) ? pulse(dt, time, -1 * max(10e-7, b* 2.0)) : c)
```

Even if one might hope not to encounter nested “IF THEN ELSE” –constructs, it still is syntactically correct, so we need to be able to do a transformation. And, have a closer look at the details: we could, for instance not just write $-\max(\dots)$ but would have to write $-1 * \max(\dots)$. And we need to change all integer numbers like 2 to real number format, in order to make sure C++ does not silently use integer arithmetics instead which may lead to a completely different result. Do not mistake $10e-7$ as $10e - 7.0$. And make sure to find a way to differentiate between the assignment operator and the comparison operator. Both are written “=” in Stella, but need to be written “=” and “==” respectively in C++. A lot more things may happen. For instance, Stella may introduce an infix operator \oplus not known in C++. “Infix” means, you can write something like $a \oplus b$. In C++ you would need to transform this expression to $f1(a, b)$. I think, this makes clear that you would most likely not be able to do this conversion using search and replace in your text-editor unless maybe you are using emacs and are a guru of regular expressions. As we foresee iterations between model building using Stella, subsequent optimization with our tool and improving the model based on results from the optimization we need a tool that makes the combination of both methods as easy, reliable and fast as possible. Therefore we need this new precompiler.

The next thing that needed to be done, was to carefully program some build-in Stella functions like Pulse, Derivn, Mean and Round to make them available in the C++ tool.

And do not forget to compare the results with the original Stella results to make sure everything works fine. Before you finally succeed, you may go through hours of looking at equations, trying to figure out where this small difference comes from, that you encounter. I wrote a program just for this task, that visually showed where differences to the original values first appeared and how those tainted values infected others as they were used for input in other equations. Tainted values would be indicated in red numbers and you would have the names of the variables just next to it, allowing you to follow a hyperlink in order to find out where their value was set. That helped a lot and eventually helped me to track down the final

error to the tabular functions (also called “graph functions”) which Stella provides. I found out that the values exported to the equations were rounded values only. A number like 0.044 would have been converted to 0.0, making it impossible to obtain a good result⁵. It was not easy to communicate this problem to the vendors of Stella, but they finally agreed to help me and provided me with a patched version of their product. I really appreciate their help and would like to thank them for this support!

For readers familiar with difference equations here is another chunk of Stella equations and the corresponding C++ code generated, just to give an idea of what the conversion does:

```

NE_4_gr_KH_mod = GRAPH(NKH_4_init_ratio)
(0, 0.06), (2, 0.12), (4, 0.28), (6, 0.54), (8, 0.84), (10, 1.18), (12,
1.74), (14, 2.62), (16, 3.32), (18, 3.78), (20, 4)

NE_4_gr = ((NE_4_dmd-NE_4)*NE_4_growth_n)*NE_4_gr_KP_m*NE_4_gr_KH_mod *
NE_suppr_from_MatInd*NE_growth_dmd_mod*NE_vit_from_area_m

Nets:_size = GRAPH(TIME)
(1965, 1), (1975, 10), (1985, 100), (1995, 10000), (2005, 1e+006)

Info_infra_throughp = Nets:_speed*Nets:_size
Speed_per_computer = IF (TIME<2015) THEN EXP(((TIME-1965)/1.5)*LOGN(2))
ELSE EXP((50/1.5)*LOGN(2))
Info_througput = Microchips_number*Speed_per_computer
INIT Digital_information = 30
Release_time_counter = if (activates_release>0 and KH_switch<1) then
pulse(20) else 0

```

Listing 2: Original Stella equations

```

double ne_4_gr_kh_mod = Graph(TAB_FKT_ne_4_gr_kh_mod, 11,
nkx_4_init_ratio);
double ne_4_gr = max(0, ((ne_4_dmd - ne_4) * ne_4_growth_n) * ne_4_gr_kp_m *
ne_4_gr_kh_mod * ne_suppr_from_matind * ne_growth_dmd_mod *
ne_vit_from_area_m);
double nets_size = Graph(TAB_FKT_nets_size, 5, time);
double info_infra_throughp = nets_speed * nets_size;
double speed_per_computer = ((time < 2015.0) ? exp((time - 1965.0) / 1.5)
* log(2.0) ) : exp((50.0 / 1.5) * log(2.0) ) );
double info_througput = microchips_number * speed_per_computer;
double digital_information = 30;
double release_time_counter = max(0, ((activates_release > 0.0 && kh_switch
< 1.0) ? pulse(dt, time, 20.0) : 0.0));

```

Listing 3: Generated C++ equations

and an additional file containing tabular function data

```

double TAB_FKT_ne_4_gr_kh_mod[] = {0, 0.06, 2, 0.12, 4, 0.28, 6, 0.54, 8,
0.84, 10, 1.18, 12, 1.74, 14, 2.62, 16, 3.32, 18, 3.78, 20, 4};
double TAB_FKT_nets_size[] = {1965, 1, 1975, 10, 1985, 100, 1995, 10000,
2005, 1e+006};

```

Listing 4: Additional C++ definition of tabular functions

⁵ Not only did the table data output in the difference equations by Stella not match the data entered in the table manually, but more, if the values for tabular functions were set by dragging with the mouse, one digit was lost in rounding on the way to the readable table and another digit was lost in conversion to readable equations.

In order to make it easy to reconvert the model after modifications or to apply the method to other models, code generation goes beyond transformation of equations. A complete program will be generated, compiled and run as the result of one single command issued by the user. Generation of the C++ code involves plugging in the genetic algorithm library and inserting code needed for setting of policy variables and collecting data required for the objective function, such as the sum of variables in a specified time interval or calculation of average values. All those specifications are read from a configuration file in xml-Format. An example is given in Listing 5.

```

<?xml version="1.0"?>
<policy>

<timespecs time_from="1980" time_to="2010" dt="1.0/32.0"/>

<ga  gaNnGenerations="150"  gaNpopulationSize="130"  gaNscoreFrequency="1"
     gaNflushFrequency="50"/>

<variables>
  <var name="pol_birth_increase" min="0.5" max="2.5" start="2002"/>
  <var name="pol_ql_for_nkp" min="0.25" max="5" start="2002"/>
  <var name="pol_kp_123_immigr_" min="0.1" max="4" start="2002"/>
  <var name="pol_kp_ne4_immigr" min="0.1" max="4" start="2002"/>
  <var name="pol_for_nkp_increase" min="0.25" max="4" start="2002"/>
  <var name="pol_for_nkh_increase" min="0.25" max="4" start="2002"/>
  <var name="pol_ne_locational_factors" min="0.2" max="5" start="2002"/>
</variables>

<task name="unemployed">
<objectivefunction type="minimize" caption="Minimize unemployment ratio">
SUM(unemploy_ratio, 2002,2010)
</objectivefunction>

<output>
  <var formula="unemploy_ratio" name="Unemployment ratio"/>
</output>
</task>

<task name="employed_new_economy">
<objectivefunction type="maximize" caption="maximize employment new
economy">
SUM(kp_ne_123 + kp_ne_4 + kp_ne_56 + kp_ne_7,2002,2010)
</objectivefunction>

<output>
  <!--ne_emp_dmd_total-->
  <var formula="kp_ne_123+kp_ne_4 + kp_ne_56 + kp_ne_7" name="kp_ne"/>
</output>
</task>

... more tasks

</policy>

```

Listing 5: objective functions and available policy variables are defined in a text file

Expressions like `avg(unemployment_rate, 2002,2010)` are automatically expanded to code that calculates the average unemployment in the year 2002 to 2010.

This way, it is possible to use the conversion program without any exposure to C++ code or programming skills required.

Results are written in text files that can be easily transformed to Graphs by programs such as Matlab.

I am planning to enhance the program to include a graphical user interface. Ideally, you would chose a Stella model to analyze. Once it is read, you would be able to chose from a list of variables to be used as policy values by clicking on their names. After indicating the feasible range and creating objective functions by selecting variables from a list, you would press a button “compile and run” and have results pop up graphically, with the option to add and remove variables from plots as needed.

References

- [1] Lawrence Davis, *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publ. Inc., 1987
- [2] David I. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989
- [3] Grossmann, W. D. “*Realizing sustainable development with the information society - the holistic Double G-Link approach*”. In: *Landscape and Urban Planning* 50, Special Issue., 2000
- [4] Wolf D. Grossmann, *Entwicklungsstrategien in der Informationsgesellschaft*, Springer-Verlag Berlin Heidelberg, 2001
- [5] F. Herrera, J.L.Verdegay (Editors), *Genetic Algorithms and Soft Computing*”, Physica Verlag Heidelberg, 1996
- [6] Matthew Wall, *GAlib: A C++ Library of Genetic Algorithm Components*, <http://lancet.mit.edu/ga/>
- [7] Michalewicz, Zbigniew, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag Berlin Heidelberg, 1992