A temperature-based, stochastic model of mate-choice with incomplete information and evolving representations

Thesis submitted to the
Institute for Graduate Studies
in Social Sciences
in partial satisfaction of the requirements for the degree of

Master of Arts
in
Cognitive Science

by
Elif T. Kuş

Boğaziçi University
2005

The thesis of Elif T. Kuş
is approved by

Assoc. Prof. Dr. Ali Tekcan ____________________________

Prof. Dr. A.C. Cem Say ____________________________

Prof. Dr. Ethem Alpaydın ____________________________

Prof. Dr. Robert M. French ____________________________

Assist. Prof. Karanfil Soyhun ____________________________

June 2005

ACKNOWLEDGMENTS

I would first like to thank my supervisors Ali Tekcan and A. C. Cem Say for agreeing to oversee this work and for the time and energy they were willing to put into its realization. Their comments on the work have improved its quality, scope and clarity. I would like to thank Professor A. Sumru Özsoy for the support, both intellectual and financial, that allowed me to complete this thesis. And I would like to thank all of the members of my committee for taking the time to read and comment on this thesis.

From the start I knew that I wanted to write a cognitive science thesis rather than a thesis that was actually an experimental psychology or a computer science thesis with a cognitive science dressing. I'm happy to have achieved it and I certainly couldn't have done it on my own.

My biggest thanks go to Professor Robert French who taught me that writing a thesis is a lot like running a marathon. Having now finished it, I finally understood what he meant. He has been a tremendous coach, supporting me throughout, both intellectually and emotionally. He has been a role model for me both in his way of doing research and in his attitudes toward research and life, in general. Discussing every aspect of this work with him was, for me, the most enjoyable part of the thesis process.

I'd also like to thank Professor Boicho Kokinov and the people at the Center of Cognitive Science at the New Bulgarian University for organizing the International Summer School in Cognitive Science and for giving me a scholarship. I benefited from the summer school immensely. This Summer School allowed me to meet professionals in the field of cognitive science as well as other graduate students with interests similar to my own. The short, but dense, courses gave me a feeling for what cognitive science really is, its interdisciplinary nature, how it is performed, etc. I would especially like to thank Peter Todd for the course he gave on Decision Making. It was during this course that I got my first exposure to mate choice models and social simulations. His course planted the seeds from which this thesis grew.

I would also like to thank Ali Onur Uyar and Umut Özge for introducing me the field and encouraging me to take courses in it when I was in ODTÜ. Thanks to Cem Bozşahin for his contagious enthusiasm for Cognitive Science. I'd like to thank my colleagues Ali Albert Salah, Amaç Herdağdelen, Özde Ak for the stimulating discussions we had.

I'd also like to thank to the people who attended my presentations prior to the thesis defense. Their feedback significantly improved this thesis. I'd like to thank all my colleagues at the Western Language department, in particular, Zeynep Kulelioğlu, Hasan Mesut Meral and Beste Can Yolcu for their support. I would like to thank Reşit Canbeyli for welcoming me into the cognitive science program at Boğaziçi in the first place. I would like to thank A. C. Cem Say (once again), Mine Nakipoğlu, Karanfil Soyhun and İlhan İnan for their crucially important support throughout the course of the Master's program.

I'd like to thank my running partners and close friends Elif Yıldızhan, N. Sevgi Öztok, Emin Elker for listening to me 3 times a week, my close friends Özlem Kutlu and İpek Saraç for their emotional support, and my always-online, far-away-but-always-around friends, Murat Şensoy, Murat Parlakışık and Çağatay Köksoy, who gave me a number of important pointers for my code for the mate-choice program described in this thesis. And I'd like to thank to Seeyam Brijmohun for having proofread the entire thesis.

Last but not the least, I want to thank my mother and sister who were always by my side while working on the thesis. Without their love and support this thesis would have never happened. And finally, I would like to thank my father, who will unfortunately never read these lines, for having understood the value of higher education and supported my enthusiasm for the cognitive science program at Boğaziçi University.

ABSTRACT

A temperature-based, stochastic model of mate-choice with incomplete information and evolving representations

by

Elif T. Kuş

This thesis is about mate-choice, specifically, about how population-level mate-choice behavior can emerge from a number of underlying preference mechanisms possessed by the individuals in the population. Modeling the decision-making processes involved in mate-choice is interesting because it involves the interplay of a number of competing pressures and constraints. Specifically, individuals must attempt: i) to find the best mate possible, ii) with only partial knowledge of the individuals in the pool of potential candidates, and iii) in a limited amount of time. In this thesis we develop a model of mate-choice that differs with current mate-choice models in five major respects — namely, it incorporates computational temperature as a measure of choosiness; it uses, for each individual, a multi-dimensional vector of mate-value for a variety of characteristics, instead of a single, overall mate-value describing the individual; it employs a fluid representational structure for potential mates that evolves over time as new information about that person becomes available; it uses subjective mate-values, since mate-value is largely, although certainly not completely, subjective ("beauty is in the eye of the beholder"); and it incorporates a self-esteem parameter that acts as an internal gauge measuring the mate value of a particular individual with respect to potential mates.

With the context of these constraints, we show that this model (the "Standard" model) qualitatively reproduces empirical data on first-marriage rates at various ages, male-female marriage age shifts (women initially marry earlier than men), mate-value correlations among married couples, and changes in marriage-rate curves when pressure to marry early is decreased. Simulations are also run in which the Standard Model is compared to a number of parameter variations to study its performance under conditions which would be possible in a real environment.

KISA ÖZET

Eksik bilgi ve evrilen temsilli, sıcaklık-temelli, stokastik bir eş seçme modeli

Elif T. Kuş tarafından

Bu tez; eş seçme, daha ayrıntılı olarak, populasyon seviyesinde eş seçme davranışlarının, populasyondaki kişilere ait belli bir sayıda altta yatan tercih mekanizmalarından nasıl belirebildiği ile ilgilidir. Eş seçerken kullanılan karar verme mekanizmaları ilginçtir, çünkü rekabet halinde olan itici güç ve kısıtların etkileşimi ile ilgilidir. Daha ayrıntılı olarak, bireyler şunları yapmalıdır: i) olabilecek en iyi eşi bulabilmek, ii) bunu sadece potansiyel adaylar havuzundaki bireyler hakkında sadece kısmi bir bilgi ile yapmak, iii) bunu kısıtlı bir sürede yapmak. Bu tezde, var olan eş seçme modellerinden temelde beş açıdan farklı bir eş seçme modeli geliştirdik. Bu beş özellik sırasıyla şöyle: berimsel sıcaklığı ('computational temperature') seçiciliğin bir ölçüsü olarak içermektedir; her birey için, kişiyi tanımlayan tek boyutlu genel bir eş değeri ('mate value') yerine kişinin çeşitli karakteristikleri için çok boyutlu bir eş değeri ('mate value') vektörü kullanılmaktadır; potansiyel eşler için, o kişiyle ilgili yeni bilgi geldikçe zaman içerisinde değişen akışkan temsili bir yapı uygulanmaktadır; öznel eş değerleri kullanılmaktadır, çünkü eş değeri, tamamen olmasa da, ağırlıklı olarak özneldir; ve belli bir kişinin diğer potansiyel eşlere göre kendi eş değerini ölçen dahili bir ölçme aleti gibi davranan bir öz-saygı (öz-değer, 'self esteem') parametresi içermektedir.

Bu kısıtların bağlamında, bu modelin ("Standard modelin") çeşitli yaşlardaki ilk evlenme oranları ile ilgili deneysel (ampirik) veriyi, erkek-kadın evlenme yaşı ötelemesini (kadınlar başta erkeklerden daha erken evlenirler), evli çiftlerdeki eş değeri ilintilerini ve evlenme baskısı azaldığında evlenme oranı eğrisindeki değişiklikleri nitel (kalitatif) olarak yeniden ürettiğini gösteriyoruz. Simülasyonlar,

aynı zamanda Standart Model'i birkaç değişik parametre çeşitlemesiyle (varyasyonla) karşılaştırmak amacıyla koşturuldu. Standard Model'in, gerçek bir ortamda olabilecek durumlarda başarımının (performansının) incelenmesi için belli sayıdaki parametre çeşitlemeleriyle (varyasyonlarıyla) karşılaştırılması amacı için de benzetimler (simülasyonlar) koşturuldu.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Overview

This thesis is about mate-choice, specifically, about how population-level mate-choice behavior can emerge from a number of underlying preference mechanisms possessed by the individuals in the population. We begin by introducing a number of concepts and tools from evolutionary psychology and computer science that will be used in the mate choice model developed herein. We will briefly discuss the major issues involved in human mate-choice, provide an overview of our evolutionary perspective on this question, and discuss the pertinence of computer modeling in the area of mate-choice.

1.1. Introduction

Most men and women, at some point in their lives, form a long-term pair bond that is usually made official by an act of marriage. These marriages are the direct result of mate-choice decisions made by both parties.

Modeling the decision-making processes involved in mate-choice is interesting because it involves the interplay of a number of competing pressures and constraints. Specifically, individuals must attempt:

i) to find the best mate possible,

ii) with only partial knowledge of the individuals in the pool of potential candidates and

iii) in a limited amount of time.

Let us consider these points in more detail. First, it is clear that each individual's goal is to find the most suitable mate possible. Second, individuals, when they first enter the mating pool, do not have an accurate idea of the kinds of individuals already present in the pool and, crucially, do not know the kinds of individuals who are likely to agree to go out with (and, ultimately, marry) them. The process of dating allows them to gradually build up more accurate representations of a number of individuals within that population. However, they do not have sufficient time to go over the whole mate pool and gather complete information of all of the characteristics of all members of that pool. They must, therefore, make mate-choice decisions based on partial information gathered in a limited amount of time.

Yet studies have shown that most individuals are, nonetheless, generally able to pair up with a mate with similar characteristics in a reasonable amount of time (Kalick & Hamilton, 1986). In other words, individuals have developed strategies that will lead them to learn enough about possible mates, and on the basis of that partial information, pair up accordingly. While decisions involved in mate-choice are highly personal, there are nonetheless a number of documented constraints that all members of the population are subjected to, such as, limited reproductive lifespan, social norms, etc. Demographic studies have shown that there are regularities across populations in the pattern of age at which people first get married (Coale, 1971; Todd, Billari & Simão, in press). The question that will be explored in this thesis is: What are the mechanisms at the individual level that enable individuals to match up appropriately and, in so doing, generate realistic (i.e. similar to those found in the empirical literature) patterns at the population level?

We approach this question from the perspective of evolutionary psychology, making use of findings from social psychology and from demographic studies. We have developed a simple computer model of mate-choice based on a small number of basic, empirical assumptions involving, in particular, the degree of mate "choosiness" (Jennions & Petrie 1997) and the male-female preference profiles for various characteristics found in their mates (Buss & Barnes, 1986). We will show that the high correlation between the characteristic profiles of married individuals emerges from the model, along with empirically observed first-marriage profiles in the population.

1.2 Mate choice: an evolutionary perspective

Mate choice, which is realized through individual decision-making processes, gives rise to the evolutionary dynamics of sexual selection (Darwin, 1859, 1871; etc.). It is important to distinguish between natural selection and sexual selection. According to natural selection, an individual who survives long enough to mate with a member of the opposite sex who is physically fit, mentally alert, is good at avoiding predators, can provide good parental care and/or sufficient resources, etc., is more likely to have healthy and fertile offspring, who, in turn, are more likely to survive. In short, natural selection is based on the *survival* of those individuals best adapted to their environment. But Darwin was deeply troubled by the existence of animals that were clearly *not* optimally adapted for success at survival, but, rather, seemed to be optimally adapted for success at reproduction. This fact eventually led him to propose the mechanism of sexual selection. Males, typically, make use of ostentatious displays to attract females and the female then chooses her mate. Sexual selection is about mate-selection and not about "mere" survival. Survival without offspring is, in the Darwinian view, little different from no survival at all. The mechanisms of sexual and natural selection are often at odds with one another: The peacock has a tail that attracts females with its beauty and size, yet the size of it hampers its movement increasing the chances of its ending up being a prey. In any event, the display-and-selection mechanisms of sexual selection are key to our model: Males ask out a variety of females, but on each occasion, it is ultimately the female who will accept or decline the invitation.

This perspective enables one to formulate a rough computational theory of mate choice (Miller & Todd, 1998). The goal of choosing a mate is to find the best mate under various constraints, such as limited search, information, and lifespan. The best mate would be the mate who has genes giving rise to healthy and fertile offspring and that has parenting ability and sufficient resources to provide a good environment for the child to grow up in. The process of finding such a mate can be analyzed in three steps:

(i) Individuals search through a series of potential mates and decide which ones to court or accept based on their overall attractiveness.

(ii) The overall attractiveness of an individual is judged by integrating a number of sexual cues (thereby unconsciously estimating underlying characteristics such as health, intelligence and parenting abilities).

(iii) To be able to assess the overall attractiveness of a potential mate, the mate-seeking individual should be able to identify and interpret sexual cues and be able to make use of them efficiently.

Sexual cues reveal information about sexual prospects. Some of the studied sexual cues are height, intelligence, facial averageness, kindness, scent, male jaw size, female waist-to-hip ratio and political status (Miller & Todd, 1998). These cues are used to compute overall attractiveness and to decide whom to court. The cues are somehow weighted and integrated to yield overall attractiveness. There are different models of integration. One is the linear model where each single cue is multiplied by a weight representing its importance and summed up to a single value. Another is the sequential aspiration model where the sexual cues have different assessment times and minimal thresholds. The cues are assessed hierarchically where the individual is evaluated further only if the thresholds are exceeded. People date a number of individuals and settle on one of them. Mate choice is a process of sequential search rather than a selection from a set of known options. A number of sequential mate search strategies have been suggested.

We have adopted a hybrid approach in our model. For the male, the choice of who he asks out is made in a "parallel" fashion (i.e., he has a number of options of women to could potentially ask out, and he chooses one person among the options to ask out). On the other hand, the woman chooses "sequentially" (i.e., she accepts or rejects each person who asks her out when he asks; she does not accumulate requests and then chooses among them.)

1.3 Our mate-choice model: an overview of its principles

In the present study, we focus on the means of searching for a potential mate, the decision to court and eventually the decision to settle on one of the prospects. This search is carried out under certain constraints on the means of searching the space of potential mates. We present *a stochastic model*, in the sense that it is fundamentally driven by probabilities, rather than certainties. So, for example, if a potential mate has a mate-value of 8, and another a mate-value of 5, the first will not necessarily be chosen. The latter will also have a chance (under some circumstances, to be discussed later, equal to that of the first individual) of being chosen.

The selection process relies on the use of the perception of the potential mate's individual characteristics, a preference for each of those characteristics, and two internal variables that gauge the environment. One of these variables is self-esteem, which is a rough measure of the individual's mate-value assessed by the individuals in the population, and the other is temperature, which measures the degree of choosiness (Jennions & Petrie 1997) of the individual in selecting a mate. Each individual in the population is represented by thirteen characteristics and his/her respective preference for each of these characteristics in an individual of the opposite sex. The characteristics of individuals are gradually revealed through encounters and serial dating. Thus, mate information improves gradually as the number of contacts between two individuals increases. In other words, a given individual initially does not have anything like a complete representation of any other individual. This internal representation of each potential mate is gradually built up through contact with that individual. Gradual representation-building – as opposed to static mate-values which are displayed for all to see – plays a crucial role in this model.

For modeling the individuals, besides notions from evolutionary psychology, the sociometer theory of self-esteem and findings from mate preference studies from the social psychology literature have been used (Baumeister, 1999; Buss & Barnes, 1986; Clark, Shaver & Abrahams, 1999; Ellis, 1992; Kirkpatrick & Ellis, 2001; Leary, Tambor, Terdal & Downs, 1995). The notions of *a parallel terraced scan and context-dependent computational temperature* taken from the computational modeling of analogy-making (Mitchell & Hofstadter, 1990; Mitchell, 1993; Hofstadter et al., 1995; French, 1995) have been used as search and control mechanisms (see below). Neither of these mechanisms has been applied in a systematic way in any current model of mate-choice. The outcomes are evaluated against demographic findings.

Further, our model differs from all other current models in that it uses *a multi-dimensional mate-value for each individual*, information about potential mates is dynamic and partial, individuals have their own mate-preference schemes, even though overall mate-preference trends from the literature were used to develop these profiles; and, while decisions are made mutually, ultimately the choice to accept or to refuse a date proposition (or a marriage proposition) lies with the female.

Information gathered by a particular individual about a member of the opposite sex accrues gradually as a function of the number of contacts and dates between the two persons. In other words, there is no fixed representation of any individual that is available to all other individuals. As in real life, representations by various individuals of the same person can differ significantly depending on how often the two have met. This is because each individual has a different mate-value preference profile (i.e., his or her set of weightings for each of the characteristics of a potential mate), in addition to having not necessarily acquired the same information for a potential mate, *mate-value is observer-dependent*.

The decision to date is initiated by the males, and the females accept or reject these courtship offers. Different than the other human mate models, *we assumed female choice*.

Specifically, we argue that individuals act in a stochastic manner, with incomplete knowledge of potential partners and having encountered only a limited number of potential partners. Individual representations of potential mates change with time and each individual stores a record of the representations developed during encounters in the recent past. The willingness to conduct a parallel exploration of the space of potential mates (i.e., encounter numerous people of the opposite sex) depends on two internal parameters associated with each individual called *self-esteem* and *temperature*. Self-esteem is a function of overall prior dating success and age. The higher self-esteem the more willing is the person to take risks and dare to ask out an individual. Temperature ("choosiness") is a function of both an individual's level of involvement in the present dating process and his/her age. The higher the temperature, the more willing the individual is to explore; the lower the temperature, the less willing – generally meaning that he has found a partner and is developing that relationship. When temperature becomes low enough, "marriage" occurs and the two individuals drop out of the mate pool.

Moreover three variations of the standard model are tested. These variations allow us to analyze the performance of the proposed variables and mechanisms. The variations are the equal-choice variation, the single-mate-value variation, and the no-self-esteem variation.

1.4 Evolutionary psychology and social psychology

In its early years cognitive science tended to view the human individual on its own, largely, if not completely, isolated from his or her environment. However, research in evolutionary and ecological psychology has revealed the importance to cognition of both societal interaction and the environment (Bechtel, 1998; Sperber, 1999). Recently, cognitive science has widened its scope to include the societal and physical environments in which human cognition operates. Within this study the problem of mate choice has been included within the joint framework of evolutionary psychology, which attempts to view the human mind as a product of their past as hunter-gatherers, and social psychology, which views the human mind in its societal context and focuses on our group behavior today. It is necessary to briefly review these two areas, which are relevant to this thesis.

1.4.1 Evolutionary psychology

Evolutionary psychology is rooted in evolutionary biology, which, in turn, is based on the framework of Darwin's theory of evolution. Evolution is a change in the pool of a population over time (Cosmides & Tooby, 1997). As discussed above, it is generally accepted that there are two mechanisms of evolution (Miller, 1999; Wright, 1995), natural selection and sexual selection. In this thesis, we are concerned almost exclusively with the latter.

There are generally considered to be two types of sexual selection: intersexual and intrasexual selection. Intersexual selection is the female choice of males. The most charming male will win. Intrasexual selection refers to male-versus-male competition for the female. In this case, the “better” (i.e. stronger, smarter) male will win the fight. Intersexual selection results in the evolution of brighter colors and ornamentation in male individuals; intrasexual selection results in large body size and natural weapons.

Darwin identified female choice (cf. Darwin, 1859, ch. 4) but he did not offer an explanation for it. Trivers (1972) proposed that the reason for female choice was the differential parental investment that each sex makes in its offspring. The male invests his sperm, of which he has plenty, whereas, in contrast, the female invests her few, large, energy-costly eggs, which are produced relatively infrequently (e.g. once a month, once a season). Further, in the case of mammals, the female carries her future offspring in her womb, demanding significant energy resources. That the female invests more in offspring than the male has made her choosy in mate choice. The high-investing female tries to find a mate willing to invest in offspring (by nest building, providing resources or protection, etc.) while the low-investing male tries to mate with as many females as possible.

The ultimate aim of sexual reproduction is to produce offspring who are, themselves, likely to survive and reproduce. An individual can positively influence the fitness of his or her offspring by choosing a mate who can offer high-quality genes and can provide good parental care. From the point of view of evolutionary psychology, this is an adaptive goal that is subject to constraints and has led to the emergence and development of various display and selection mechanisms by males and females, respectively.

Evolutionary psychology is an approach to psychology that aims to discover and understand the mind by making use of evolutionary biology. Unlike the study of vision, reasoning, or social behavior, evolutionary psychology is not a specific subfield of psychology. Rather, its aim is to synthesize the principles of modern evolutionary theory with current formulations of psychological phenomena (Tooby & Cosmides, 1992). According to this view, the human mind is an integrated collection of adaptive mechanisms that are functionally specialized, usually domain-specific and universal. These mechanisms are presumed to be the result of natural and sexual selection, especially during the early years of our species. Within this framework, psychology is the study of a diverse collection of interacting neurological programs, each of them built to solve problems faced by our ancestors.

Our ancestors lived in hunter-gatherer societies in an environment resembling the African savanna for 99% of our evolutionary history (Cosmides & Tooby, 1997).

During this time, the human brain, consisting of specific neural circuits, was gradually modified by natural and sexual selection to solve a wide range of problems. These neural circuits evolved adaptations that met the challenges in the hunter-gatherer environment. There are many neural circuits that are specialized for solving different adaptive problems. Tooby and Cosmides' (1992) somewhat surprising and powerful conclusion is that the mind is a collection of specialized modules rather than a general purpose problem solver.

In light of these principles, evolutionary psychology is concerned with underlying mechanisms that are universal and are related to reproductive success. The ultimate measure of reproductive success is the amount of one's genetic material that has been passed to the succeeding generations.

Finally, evolutionary psychology enables us to see human beings in the context of their evolutionary history and their adaptation to their environment. Social psychology, on the other hand, concentrates on the present day.

1.4.2. Social psychology

Social psychology is the study of the nature and causes of human social behavior by examining the relationship between mind(s) and social behaviors (Fiske, 2004). Social psychology attempts to understand the relationship between minds, groups, and behaviors in three general ways. First, it tries to see how the thoughts, feelings and behaviors of individuals are influenced by the actual, imagined, or implied presence of other(s). Second, it tries to understand the influence that individual perceptions and behaviors have upon the behavior of groups. Third, and finally, social psychology tries to understand groups themselves as behavioral entities, and the relationships and influences that one group has on another group. In the present work, we are mainly interested in how the individual is influenced by the presence of other individuals and groups. Specifically, we are interested in the topics of intimate interpersonal attraction, self-esteem and mate preference.

Evolutionary psychology and social psychology are closely related in terms of their respective domains. Most of the adaptive problems, our human ancestors confronted, were social ones (Buss & Kenrick, 1998; Kenrick, 1994). This overlap has generated a new subfield called evolutionary social psychology which studies the problems of social psychology with an evolutionary approach. The evolutionary approach to social psychology allows one to consider the human social behaviors under the perspective of evolved function. Adaptations are the primary product of the evolutionary process; many mechanisms of mind are designed to solve specific adaptive problems. The sociometer theory of self-esteem and the mate preference studies used in the present model are part of this evolutionary approach to social psychology.

1.5 Computer simulation and modeling

Computer simulation is an ideal means of studying population and behavioral dynamics based on the findings of evolutionary psychology, social psychology, and sociology.

Simulation and models are used in many fields with different purposes. A cardboard model of a house is used by architects for evaluating the appearance of the prospective house. A weather forecast simulation is used by meteorologists for predicting the following week‘s weather. Drug simulations are used to estimate which drugs will be appropriate for interactions with a certain protein, thereby narrowing the number of chemicals to be used in drug trials.

Simulations and models necessarily operate in idealized environments in which an attempt is made to isolate the set of variables relevant to the particular problem being studied. To study weather patterns in Northern Europe, we do not need to include the mass of the earth but we do need to include variables describing the path of the Gulf Stream; to study planetary motion, on the other hand, information about the path of the Gulf Stream is irrelevant, but the mass of the earth is crucial. A cardboard model of a house maintains the same proportions among the dimensions of the walls, windows, and doors, but the material used and the overall size is different. A weather forecast simulation consists of numbers and mathematical equations but it does not contain any clouds or sunshine. Drug simulations again do not contain any of the chemical substances present in the material that is modeled. Nonetheless, all of these simulations are capable of producing empirically valid outcomes. Simulations and models are idealizations of an event, an object or a phenomenon. A model can produce valid outcomes when the variables and properties included in the idealization are those that play a role in the modeled phenomena. Good models should ultimately be able to not only reproduce existing empirical data, but also to make novel predictions that can subsequently be empirically tested (French & Cleeremans, 1996).

1.6. The goal of modeling: formalization and explanation

The uses of models and simulations in science are mainly two, which are formalization and explanation (Gilbert & Troitzsch, 1999). Simulation and models serve as tools for formalization. To build models and simulations scientists take theories that have been textually expressed and formalize them into specifications that can be converted into a computer program. One of the great values of computer simulation is that these specifications of a theory have to be sufficiently detailed and free of ambiguities to be able to actually be implemented. It is often during the implementation phase that gaps and hidden assumptions in the theory come to light. Computer programs are not compatible with vague, ambiguous arguments.

The second contribution of modeling is explanation. If the computer model simulates the relevant parts of the process under consideration and thus produces similar outcomes, it can potentially reveal the underlying mechanisms of the phenomenon, thereby providing an explanation of it. At the very least, modeling provides a set of *sufficient* conditions for the production of a given phenomenon. Simulations used for explanation are often software implementations of abstract thought experiments whose goal is to explore mechanisms underlying observed patterns (Gilbert, 1997). If the goal is to deepen our understanding of some fundamental process then the idealized form – i.e., the *essence* – of the phenomenon is what the model must capture; a realistic representation of all the details of the particular setting is irrelevant. Counterintuitively, making models overly realistic may add too much complexity and actually *decrease* their explanatory value, simply because the wealth of details make it much harder to know precisely what in the model produced the desired output (Macy & Willer, 2002).

Simulations are similar to mathematical models in that they both provide explanations and are means of formalizing a particular real-world situation. However, real-world phenomena are usually highly non-linear. And when it comes to nonlinear phenomena (or when there are simply too many variables to deal with), mathematical models, in general, fall short in that either the phenomena cannot be adequately represented or cannot be solved analytically. Computer simulation is often the only available means of simulating these phenomena.

A computer simulation is a computer-executable model. The investigation of a process through simulation begins with a model of the phenomenon of interest. This model is based on a set of assumptions derived from observation and theory. The model will capture only some part of the phenomena, which is why it is important to make the appropriate assumptions, in particular, appropriately choosing the variables that are believed to be involved in the targeted process. The model is then formalized, after which it can be implemented in the form of a computer program.

The next step is running the program and recording the output. This output is compared with real world data and other theories (that were not used during building the simulation). This is sketched in Figure 1. This comparison of the model's output — in particular, predictions produced by the model — and empirical data will, in general, lead to the revision of the model. The revised model will then make new predictions, again to be tested in the empirical domain. This continual back-and-forth interaction between explanatory models and empirical data is discussed in French & Cleeremans (1996).

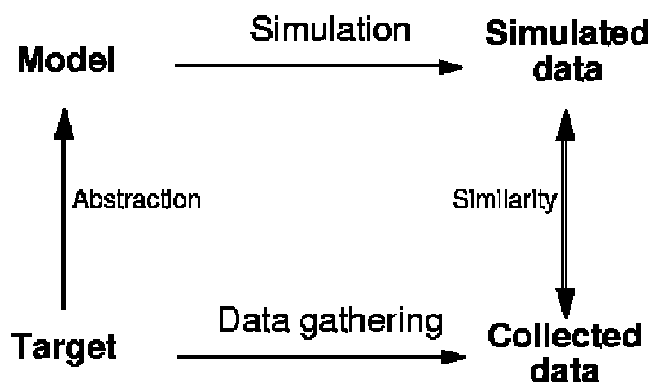

Figure 1. The logic of simulation as a method (In Gilbert & Troitzsch, 1999).

1.7. Multi agent-based modeling of population dynamics in social phenomena

Specifically in this study, the method of multi agent–based modeling in social simulations is used. Social simulation is the use of computer simulations for modeling the dynamic behavior of societies and their relevant environment. Arguably the most important advantage of using simulations in modeling behavior at the level of societies is that it is possible to model nonlinear phenomena and emergent properties.

Very simple rules programmed at the individual level can produce unpredictable, complex population–level behavior (Gilbert & Troitzsch, 1999). A typical example is the communication of ants by pheromones (Bonabeau & Meyer, 2001). A pheromone is a chemical marker which ants can smell. For finding food ants move randomly around their environment and leave behind a pheromone trace. When they find food they go back to the nest following their own pheromone trace. Ants follow not only their own pheromone paths but also the traces of other ants, and they are more inclined to follow one path over another if it has a stronger pheromone scent. When the other ants coincide with this food-finder ant's trace they are more likely to follow it, as the original ant already went over it twice. And as more ants go over it, more ants follow the path. When the food resource is exhausted, the pheromone trail's intensity will decrease, leading ever fewer ants to follow it. The simple ant-level rule "Follow the pheromone," causes complex population-level behavior — namely, "Communicating the location of food to the population, thereby causing many ants to go to the food." This illustrates perfectly the key notion of emergence of high-level phenomena from the simultaneous low-level actions of many agents.

In short, simulations allow us to connect the micro level of individuals' behaviors to the macro level of the global properties of the population. Furthermore, the dynamic aspects of change can be represented. Finally, simulation is an experimental methodology, which means that after setting up the simulation, it can be run many times and the results of different parameter settings can be explored. The obvious advantage of agent-based models of social phenomena is that we can manipulate the variables that govern low-level behavior and observe the effect of these changes on the high-level behavior of the population. This is rarely, if ever, possible for real-world social phenomena. The experimental environment of computer simulation provides unrivalled possibilities for studying the result of manipulating variables that we could never hope to manipulate in a real-world population.

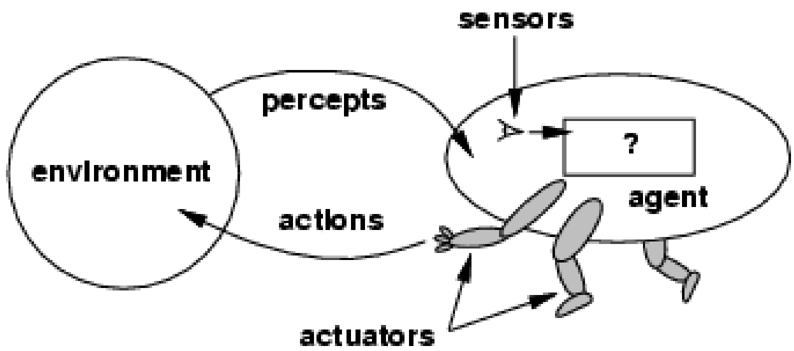

Figure 2. The agent interacting with its environment. (In Russel & Norvig, 2003)

In agent-based simulations there are a number of agents that are represented explicitly and univocally. Agents are generally relatively simple, self-contained programs that have an internal data representation (state or memory), can modify their internal data representations (perceptions) and can modify their environment or their own behavior (Russell & Norvig, 2003; see Figure 2).

These model agents represent human individuals, non-human animals or collectivities such as firms, or even, states. Agents interact with each other or the environment by means of variables and rules that may vary between agents. What distinguishes an agent-based approach from other modeling techniques in the social sciences is the set of properties of agents (Macy & Willer, 2002). First, agents are autonomous. Population level patterns emerge from local interactions among autonomous decision-makers (i.e. system-level self-organization occurs). Second, agents are interdependent. Agents either influence each other directly in response or change some aspect of the environment, which in turn affects the other agents. Third, agents follow simple rules. Global complexity emerges from the simple behaviors of many agents. The apparent behavioral complexity of individuals is largely the reflection of the environment. Fourth, agents are adaptive, backward-looking, and have memories of encounters (and the result of those encounters) in the recent past. When agents are also adaptive, their interaction can generate a "complex adaptive system" (Holland, 1975). Some attributes of humans that have been simulated by multi-agent social simulations are knowledge and belief, inference, social models, knowledge representation, goals, planning, language and emotions (Gilbert & Troitzsch, 1999).

1.8. Summary

In this chapter we have briefly laid out the principles from evolutionary and social psychology on which our model will be based. We have discussed the underlying principles and validity of evolutionary approaches to understanding social behavior. Most importantly, we have attempted to present a succinct overview and defense of agent-based approaches to modeling of high-level social phenomena. We maintain that an agent-based approach is an appropriate manner to model mate-choice. In the following chapter we will review a number of other current models of mate-choice before presenting the implementation details of our model in Chapter 3.

2. Previous work on models of mate choice

The recognition that sexual selection emerged as the combined effect of many low-level (i.e., individual) decision-making behavioral processes led to the development of new mate-choice models that incorporated individual strategies (Bergstrom & Real, 2000). These mate-choice models vary a great deal in their scope and aims. Most of them involve non-human mate choice, probably due to the traditional focus on non-human animals in many early studies in evolutionary biology.

2.1 Animal mate-choice models

One method that has been frequently applied to mate choice is game theory (Parker, 1983; McNamara & Collins, 1990; Johnstone, 1997; Alpern & Reyniers, 1999; Bergstrom & Real, 2000). Game theory is the mathematical study of interactions of individuals in a group described by rules of interaction and alternative choices (Levine, 2001).This method seeks to find rational strategies in situations where the outcome depends not only on one's own strategy and market conditions, but upon the strategies chosen by other players with possibly different or overlapping goals.

Parker (1983) was the first to do a formal analysis using the game-theoretic approach (Johnstone, 1997). One major underlying assumption of this model was the notion of mutual selection, meaning that both parties had the right to accept or refuse a mate. One major finding of this model was that mutual choice lead to assortative mating (i.e., mating with a mate that is similar in traits or/and characteristics).

McNamara and Collins (1990) further refined this model by describing a single stable strategy which resulted in assortative mating (i.e. selective mating between individuals that have similar characteristics) (Simão & Todd, 2002). However, McNamara et al.'s and Parker's models both assume that all potential mates are accessible at all times, that there is unlimited time for search, that mate-values are static over time, and that all individuals in the population have perfect information about all potential mates, their mate-values, etc. (Bergstrom & Real, 2000; Simão & Todd, 2002). These assumptions are unrealistic assumptions for a human population (Simão & Todd, 2002).

Relaxing the previous model's assumption of unlimited time for search, the model by Johnstone (1997) is based on serial, mutual and competitive mating behavior. The model makes predictions about the relation of mate qualities to individuals' choosiness during different times of the season. The model is based on the assumption that individuals know their own mate-value, as well as the overall distribution of mate-values in the population.

The model developed by Alpern and Reyniers (1999) explores the consequences of homotypic preferences (i.e., individuals preferring mates similar to themselves) rather than type preferences (individuals preferring mates with high mate-values), which had been assumed in previous game-theoretic models. The predictions of this model involve the choosiness of the individuals depending on the time of mating and their ranking in the mate distribution scale.

Bergstrom & Real (2000) criticized the assumption of uniformity of preferences (for both homotypic and type preferences) of the previous models and propose a model in which individuals vary considerably in their mating preferences.

There is a sexually asymmetric decision mechanism, the male courts a number of females and each courted female accepts or rejects the male. These researchers were able to model empirical data that had not been previously modeled; nonetheless, they assume complete (and static) mate-value information for all members of the population, in addition to assuming that the set of potential partners for each individual is known beforehand. This latter assumption, which engenders global stability, unfortunately diminishes the empirical validity of this model (Simão & Todd, 2002). Real-world situations are more likely to involve dynamic social networks rather than globally stable ones.

Game-theoretic models of mate-choice make two important assumptions that undermines their theoretical validity — namely, they assume that i) all participants have perfect information and ii) search is not necessary (i.e. all potential mates are immediately available). These fundamental assumptions have been criticized elsewhere, in particular, by Simão & Todd (2002, 2003).

Upon the empirical observation that in four bird species the females selectively visit males and repeat visits to males before mating, Luttbeg (1996) suggested that previous models, that assumed females randomly encountered males and immediately were able to gather complete information on the male, were too restrictive. He further assumed that the animals used a comparative Bayesian methodology for choosing a mate. The model made a number of predictions which were tested empirically.

Mazalov et al. (1996) also criticized the assumption of the availability of full information of mate-value distribution and proposed a stochastic model with adaptive search, which was used because the distribution of mate-values was not known a priori. In this model, agents learn the overall mate-value distribution as new potential mates are encountered. The mate-values aren't stored but a new average mate-value is calculated after each new encounter. This model revealed that a learning ability is useful when mate-value distributions vary and when there is sufficient time for choosing.

2.2 An overview of human mate-choice models

The models mentioned above were either on both human and non-human, or only on non-human mate-choice. The first human mate-choice simulation was done by Kalick and Hamilton (Kalick & Hamilton, 1986). The aim of their model was to resolve two conflicting findings in mating behavior. According to studies in mate preferences, individuals tend to prefer individuals of the opposite sex that are highly physical attractive regardless of the individual’s own physical attractiveness. On the other hand, it was found that there is a correlation in physical attractiveness among married couples (Kalick & Hamilton, 1986). Based on this, it was theorized that people tend to prefer people to whom they are physically similar. Kalick and Hamilton set up a social simulation where they used these two findings and managed to demonstrate that it was possible to achieve a high intracouple physical-attractiveness correlation, even when individuals preferred the most attractive individual. The simulation revealed that, since high-value mates seek high-value mates as well, they accept only high-value mates and tend to leave the population early. They are therefore no longer available to serve as a highly attractive beacon for less attractive individuals. Their model is narrow in scope and models only one specific phenomenon, but has a clear result. And, while the model has subsequently been criticized for the overly high number of dates for each individual and the low percentage of the population who got married (Aron, 1988; Simão & Todd, 2002, 2003), it was originally designed to make a resolve what was perceived to be a contradiction, and it was able to show how this might be done.

Todd and Miller (1999) proposed a similar model for testing various individual strategies for searching the space of potential mates. They tested a number of simple search heuristics in one-sided search (i.e., search by either males or females in the population) and mutual search. In their model there is a learning period that corresponds to the adolescent period where the individual learns about one's own mate-value and the values of potential mates. They learn by feedback of individuals they encounter in the population. They showed that the heuristics they tested, could achieve satisfactory results while making use of very little information. These models, however, left an unrealistically high percentage of the population without mates.

Recently a number of human mate choice models have been developed by Simão & Todd and colleagues (Simão & Todd, 2001, 2002, 2003; Todd, Billari & Simão, in press). While they are in the human mate-choice domain, they are generalizable to animal domains. They are based on realistic assumptions drawn from findings in psychology and have realistic outputs, which have been validated against sociological and demographical findings. They include a courtship period during which individuals can, if they wish, switch to better partners, thereby relaxing the assumption that a single mate-choice decision “is forever”, and, as such, is irreversible. Each individual’s perception of the population’s mate-value distribution is not available a priori, but, rather, is gradually built up. Individuals can make simple, robust and efficient mate-choice decisions by exploiting the mate pool. Simulation results match empirical findings on intracouple correlations, first-marriage age patterns and a sociological theory based on the effect of sex ratios to the rate of marrying. We will return to this model below.

A final approach is one by Kenrick, Li & Butner (2000) and uses the dynamic social networks theory. Kenrick et al. examine the evolutionary psychological finding that males are inclined to take advantage of unrestricted relationships while females prefer restricted relationships. They do this by modeling the population as a two-layer variation of dynamic social influence networks. Dynamic social influence networks provide a means of propagating "influence" throughout a population. The population model that Kenrick et al. used represented individuals by their location and orientation with respect to their neighbors. Opinions of individuals were able to change over time. Their model examined how males and females were affected by each other's mating options. It was found that never did a whole community end up unrestricted when normal beginning rules were used which is consistent with real world data.

2.3 Simão & Todd's (2003) model

Arguably the most complete, realistic model to date of mate-choice has been developed by Simão & Todd and colleagues (2001, 2002, 2003; Todd, Billari, & Simão, in press). This model will be illustrated in more detail because this important model has many points in common with our own.

Their model (i.e., Simão and Todd, 2003) consists of a number of individuals whose ultimate goal is marriage. In order to achieve this, they meet new people, ask them out, court them and, finally, get married. The pattern is as follows: individuals first meet new people and get to know them. Second, they evaluate them as possible mates. Third, if an acceptable and agreeing prospect is found, they marry and leave the unmarried population. Otherwise, if not married, they get older, learn from past experience and return to mate-pool in the hopes, once again, of finding a suitable mate. During the simulation, the age of marriage, the overall number of people getting married, and the mate-values of the individuals getting married are recorded for the comparison with demographic data of first-marriage age distributions and the intracouple mate-value correlations of married couples.

The simulation consists of a population that evolves as described above and an environment in which the individuals in the population interact. The environment specifies how the potential partners are encountered. Potential mates are encountered one by one, i.e. sequentially, rather than simultaneously (or, “in parallel”). By sequentiality, the authors mean that, if an offer by a particular male to form a couple is refused by a female, then the female’s decision is (usually) irreversible, in the sense that the female cannot later return to that particular male and “change her mind” and say “yes.” In short, mate-choice with respect to a particular individual is a one-shot opportunity. This means that the decision problem involves “settling immediately on the current option,” rather than “choosing from among a number of possibilities.” Based on the idea of ecological rationality[1], Simão and Todd propose that individuals solve this sequential-choice problem with simple heuristics that exploit the structure of the environment.

In their model, mate preferences and attractiveness are assumed to add up to a single value of mate quality which is a combination of the different relevant features.

[1] *Ecological rationality* is the combination of mental mechanisms that are designed to operate under pressures of limited time, limited information and limited computational power in environments to which they are attuned (Simão & Todd, 2003).

This one-dimensional variable is called *mate quality*. Preferences are universal: everyone prefers the same type of mates (a significant difference with respect to our own model). It is assumed that there is *time pressure* to mate. Each individual tries to find a mate with a high mate quality in a reasonable amount of time. This is modeled through a variable called *Fitness* which depends on the age of the individual and the mate quality of the potential mate. Fitness is the variable that evaluates the success of decision rules and strategies. Simão and Todd view the decision to mate with someone as a process rather than a discrete yes/no event. If both partners agree, they start to court each other.

Courtship time is the length of time that two individuals are courting It functions as a period in which the individuals evaluate each other. During this courtship period, individuals are able to switch to better partners, if they are available and willing, thus the decision to court is a reversible one. Even though there is the possibility to switch, individuals don't immediately court the first potential mate they meet. Because the possibility of meeting a potential mate is lower when the individual is in a relationship. And this interaction possibility decreases as individuals get more involved with each other. This is assessed through the *aspiration level*.

The aspiration level is a threshold setting the minimum value of a potential mate for which an individual would start courting. This aspiration level is adjusted with respect to the individual's own mate-value and the time the individual has been waiting for courtship. Individuals can guess their own mate-value as a result of their experience with the opposite sex. The aspiration level decreases whenever an individual is left (i.e., rejected) by a courting individual. As the time spent waiting for courtship increases, the expectations of the individual decrease.

The main motivation of the individuals in the mate pool is to find the best mate who will accept them. Each individual has a limited lifetime and a mate-value which is an indication of his/her quality. Each individual tries to find a high quality mate who will accept them within their lifetime.

Time is modeled in a sequence of discrete steps. At every step individuals meet with other individuals of the opposite sex at a certain stochastic rate. The rate of meeting, if the individual is in a relationship, depends on the individual's involvement in their current courtship process; or if the individual is single, this rate is at maximum. Each individual keeps a list — the alternative list — of recently met potential mates. This list corresponds to the people one can keep in one's social network and subsequently ask out for a date. The number of individuals on this list is limited, and if the list is full, previous names on the list are randomly removed for each new name. If two individuals request to date each other at the same time, then they enter a courtship process. An individual can make a request to several individuals for courtship although they can only court one individual at any given time. If an individual is accepted by several mates, he/she chooses to court the individual with the highest mate quality. Each individual makes the decision to ask out an individual, if he/she is single, by maximizing the fitness function for that person, which depends on the mate quality of the potential mate and the age of the choosing individual.

Fitness is defined as follows:

$$F(q_m, t) = q_m x \frac{L - t}{L}$$

where q_m is the quality of the individual's chosen mate, t is the age of the individual at which he/she mates, and L is the total reproductive lifetime. If the individual is already courting somebody then he/she will switch only if it provides fitness gain.

The fitness of the current mate is calculated assuming an optimistic estimation of the remaining courtship time (to mate) and the fitness of the new alternative is calculated with an assumption of the total required courtship time to mate. This is formulated as below:

$$F(q_a, t + K) > F(q_d, t + \max\{0, K_i - c_t\})$$

where:

q_a is the mate quality of the alternative,

$q_{d,}$ the mate quality of the current date,

t is the current age of the individual,,

K_i is the required courtship length for it to turn into mating/marriage, and

c_t is the current courtship time.

In this model, both males and females may initiate courtship, i.e., may ask a member of the opposite sex out. A courtship process is different than mating in that during this period one still can switch to a better partner. If the individuals stay long enough in a courtship period, they mate.

Individuals set aspiration levels to begin courting a partner, as entering a relationship has costs in terms of time and opportunities of meeting new people. Individuals propose to potential mates that have a mate-value higher than the aspiration level. This aspiration level should depend on the individual's own mate-value and the ranking of that value in the population. But agents cannot know their own mate qualities and ranking with respect to other individuals ahead of time, which is why the aspiration level is set dynamically.

The individual's aspiration level corresponds to the individual's self-quality assessment. It begins at 0 and is updated as the individual is left by his/her partner.

Mate-value self-evaluation is updated as follows:

$$q^*_{inew} = q^*_{iold} \times (1-\alpha) + \omega \times q_j \times \alpha$$

q_j is the quality of the individual's departing partner,

ω is a correction factor to decrease the individual's expectations slightly below the quality of that partner (0.8 is used) and

α is the learning rate which is used to avoid the aspiration level to change values too fast.

The aspiration level should also be adjusted with respect to the availability of potential mates. The aspiration level is reduced when the waiting time is getting too long as this is a loss of reproductive lifetime. The aspiration level is lowered when a waiting time threshold t_{max} is reached. t_{max} is calculated as below:

$$t_{\max} = \tau \times \frac{L-t}{L} \times (1 - \frac{q_b}{q^*})$$

τ is a proportionality constant,

t is the current age of the individual, and

q_b is the quality of the best alternative in the alternatives list whose quality is lower than q^*.

This formula states that the time an individual is willing to wait for an individual with the desired properties decreases as the individual gets older and as the quality of the best attainable alternative is lower. If t_w reaches t_{max}, the aspiration level, q^*, is set to q_b.

The aspiration level functions similarly to the self-esteem variable in our model. It acts as a threshold. While in this model it is the main mechanism, in out model it is a secondary mechanism to parallel terraced scan. The adjustment of the aspiration level with respect to the waiting time is similar to temperature.

Temperature includes the effects of time pressure to meet.

2.4 Summary

In this chapter we have briefly examined a number of current models of mate-choice. Many of these models are based on game-theory and assume static mate-values, perfect knowledge of these mate-values by the members of the population, and unlimited time to explore the space of potential mates. One model by Simão and Todd (2003) makes far more plausible assumptions about mate-values, and time to search the space. This model incorporates a notion, the *aspiration level*, of an individual in his/her search for a mate that resembles the key notion of *choosiness* in our own model. These notions are important as they play a major role in the decision mechanisms of the individuals. However, mate-values are single-valued entities and there is no notion of preference profiles for each individual in the population. We will discuss these differences at greater length in the next chapter in which our model is described.

3. Modeling Human Mate-choice

Previously we saw that mate-choice can be analyzed as three subtasks (Miller & Todd, 1998). The first task was identifying the cues, the second cue integration and the third was mate search and making decisions. This study concentrates on the third subtask. The third stage of mate-choice can yet be subdivided to three subtasks which are the search for potential mates, assessing potential mates, and deciding on a mate. In this model we mainly emphasize the search process and the assessment of the potential mates. The search process and the process of assessment of potential mates are intertwined, and the decision to mate emerges as a result of the first two stages. That the decision to mate emerges is as an emergent phenomenon doesn't mean that it is any less important than the first two subtasks.

3.1. Key Concepts

The model design was based on several concepts from analogy-making, social psychology and evolutionary psychology.

3.1.1. Mate-value

Mate-value is the total value of the characteristics that an individual possesses in terms of the potential contribution to his or her mate's reproductive success. It was first introduced by Symons (Symons, 1987 cited in Ellis, 1992) in the area of evolutionary psychology. That an individual has a higher mate-value means that he/she is more likely to have reproductive success (Ellis, 1992). A man who is young, healthy, strong, successful, well liked, respected by his peers, willing and able to protect and provide for his children is more likely to have children and rear them in a way that will ensure their survival to reproductive age.

According to sexual selection, females and males prefer individuals with higher mate-value. The genes of females who were attracted to or preferred males that have higher reproductive capability had a higher chance of having offspring, and vice versa. Those who had offspring passed on their preferences by passing on their genes. This continued until these mechanisms became universal. In line with this argument, it can be said that sexual selection, along with natural selection, has shaped male and female behavior so as to optimize reproductive prospects (Miller, 1999; Wright, 1995).

In contrast to previous models, we assume a multi-dimensional mate-value. Instead of a single mate-value, we have a number of characteristics for each individual, and each individual has different preferences for each characteristics. Multi-dimensional mate-value means that the mate-value does not consist of a single value (which means in terms of the model a single number), rather it is a series of characteristics (which means in terms of the model a series of numbers). Crucially, this means that mate-value is not absolute across all individuals in the population. The mate-value for an individual is different for each individual in the population since everyone has different preferences for the attributes of an individual. In other words, while that tanned, square-jawed fisherman may fabulously be attractive to the women of his coastal village, the fact that he cannot read or write will make him far less appealing as a permanent mate to women in the Istanbul literati set. There are, of course, universal tendencies in what individuals prefer. The characteristic of intelligence is always ranked first, second, or third but never as the least desired characteristics. We assume type preference (i.e. desire for the best) with individual differences.

The mate-value of a potential mate (Y) for an individual (X) is calculated by multiplying X's vector of preference-weights with the vector of values for the 13 defining characteristics of Y and summing the resulting values (i.e., taking the inner product of X's preference-vector with Y's mate-value vector). The multidimensional mate-value also means that potential mates are revealed only gradually, corresponding to the notion that it takes time to get to know a potential mate. The values of all characteristics are not revealed at once.

To compare the effect of a multi-dimensional mate-value with respect to a single dimensional mate-value, we set up a variation of the model that has a single-dimensional mate-value instead of a multi dimensional mate-value.

Simão & Todd (2003) assume a single dimensional mate-value in their model.

Mate-preferences

Everyone has mate-preferences and mate-choice is the behavioral outcome of mate-preferences (Miller, 1998). Mate-choice operates by rejecting some potential mates and accepting others. Mate-preference is the ranking, with respect to willingness to mate, of the characteristic values of potential mates sampled under direct choice (Widemo & Sæther, 1999). Mate-preferences have been further subdivided into 'preference functions' and 'choosiness' in biological mate-choice studies (Jennions & Petrie, 1997). 'Preference functions' determine the order in which an individual ranks potential mates assuming all other factors being equal.

'Choosiness' is the effort or energy that an individual is prepared to invest in assessing mates, both in terms of the number of mates sampled and the amount of time spent examining each mate. (Choosiness is inversely proportional to what we have called *computational temperature*, explained below.) Distinguishing between the preference function and choosiness allows us to get different mate choice patterns by changing the choosiness variable (e.g. indirectly by changing the number of dates) without changing the preference function. This section is on the preference functions. Choosiness, due to its function in the search mechanism, is going to be discussed separately in section 3.1.2.

Overall Rank	*Characteristic Preferences*	♀ *Mean Value of Rankings*	♀ *SD Value of Rankings*	♂ *Mean Value of Rankings*	♂ *SD Value of Rankings*
1	Kindness and understanding	2.08	1.59	2.43	2.55
2	Exciting personality	3.28	2.27	3.63	2.66
3	Intelligent	3.44	1.51	3.78	2.00
4	Physical attractiveness	6.26	2.49	4.04	2.32
5	Good health	5.84	2.57	5.49	2.34
6	Adaptability	5.72	2.95	5.67	2.62
7	Creativity	7.56	3.32	8.33	2.79
8	Desire for children	8.82	2.81	8.01	2.47
9	College graduate	7.94	2.70	9.41	2.18
10	Good earning capacity	8.04	2.59	9.92	2.19
11	Good Heredity	10.34	2.07	9.71	2.62
12	Good housekeeper	10.56	2.10	10.22	2.29
13	Religious orientation	11.12	3.16	10.24	3.53

Table 1 Preference rankings concerning the potential mate (In Buss and Barnes, 1986). The ♀ Symbol stands for female, the ♂ symbol stands for male, SD is for Standard deviation, and Mean is the Mean. The study was done on 100 unmarried students between the ages of 18-23 based on the findings on 92 married couples in the age range of 18-40.

Buss and Barnes (1986) studied the mate-preferences in males and females. Their findings showed that there was a certain overall trend in the characteristics that males and females valued, yet there is individual difference. In a further study, Buss (1989) showed that preferences for at least 60% of these characteristics are universal (Buss, 1989). In a recent review, it is concluded that beside the overall trend in preference functions, individual variation in mating preferences is common and can have major consequences for models of mate choice (Jennions & Petrie, 1997).

Overall Rank	*Characteristic Preferences*	♀ *Mean Value of Weights*	♀ *SD Value of Weights*	♂ *Mean Value of Weights*	♂ *SD Value of Weights*
1	Kindness and understanding	8.4	1.2	8.1	2.0
2	Exciting personality	7.5	1.8	7.2	2.0
3	Intelligent	7.4	1.2	7.1	1.5
4	Physical attractiveness	5.2	1.9	6.9	1.8
5	Good health	5.5	2.0	5.8	1.8
6	Adaptability	5.6	2.3	5.6	2.0
7	Creativity	4.2	2.6	3.6	2.2
8	Desire for children	3.2	2.2	3.8	1.9
9	College graduate	3.9	2.1	2.8	1.7
10	Good earning capacity	3.8	2.0	2.4	1.7
11	Good Heredity	2.1	1.6	2.5	2.0
12	Good housekeeper	1.9	1.6	2.1	1.76
13	Religious orientation	1.5	2.4	2.1	2.72

Table 2. The data in Table 1 converted to preference weights in the range of 1-10.

The data above shows that there are clear sex-biased differences in the preferred characteristics. “Physical attractiveness,” for example, is more valued by males, while females value “good earning capacity” more than males. Other characteristics, such as kindness and intelligence, are equally valued for both sexes.

In the model in this study, we will adopt the preference rankings taken from Buss and Barnes (1986) (see Table 1 and Table 2). The findings of these work as important quantitatively. What is of importance to us is that there is a certain universal trend in preferences, yet, the individual differences persist. And what the highest ranked, or second highest ranked property is for us of secondary concern. It is the relation of the preference characteristics to each other that will have implications for the model. Apart of this we are assuming a linear combination of these characteristics, where the characteristics are weighted with respect to their rankings. We converted these preference rankings into preference weights from 0-10 where the highest valued characteristic has the highest weight. It is important to notice that there is a trend in the characteristics preferred i.e., preferences are not random. Then, for each individual in the population we semi-randomly selected a value from a normal distribution with the means and standard deviation given in Table 2.

As noted before there is some evidence that a number of the characteristics are ranked the same universally. Yet what is of relevance for this model is that there is universality in the tendency to rank some characteristics over others. It is this relation of the characteristics with each other that is going to have implications. Whether kindness is universally ranked as the first characteristics is not of our concern.

Moreover, it is assumed that there is no relation between one's own characteristics and his/her preferences.

3.1.2. Choosiness and the Search Strategy - Computational temperature and the parallel terraced scan

Choosiness is a property of mate-preference. It is a measure of the effort an individual is prepared to invest in mate search and assessment (Jennions & Petrie, 1997). The search strategy is how an individual finds a mate.

Numerous types of search strategies have been proposed by both empirical and theoretical studies in general mate-choice models (Jennions & Petrie, 1997). Below is a brief classification of these strategies by Jennions and Petrie (1997). This classification has been done over empirical research in animal mate choice and theoretical research with mate choice models.

(1) Random mating: Accept the first mate encountered.

(2) Fixed threshold: Sample sequentially and accept the first mate that exceeds a pre-set criterion.

(3) Sequential comparison: Sequentially compare mates until the most recently encountered is of lower quality than the previously encountered individual, then accept the previously encountered male.

(4) Adaptive search: Sample until the value of the mate encountered is greater than the value expected from continued searching. The adaptive search rule also factors in the cost associated with continued searching.

(5) Pooled comparison ('Best-of-N'): Sample N males and then accept the male with the highest value for the preferred trait(s).

(6) Optimal stopping rule: This rule is similar to tactic 4, but is more realistic in that the individuals do not know beforehand the distribution of quality of the potential mates. But individuals do know how many potential mates they will encounter.

In this model, we propose a search mechanism based on the dual notions of a "parallel terraced scan" and the "choosiness" of the individual. These mechanisms were previously used in a certain class of computer models of analogy-making (Hofstadter, 1995; Mitchell and Hofstadter, 1990; Mitchell, 1993; French, 1995).

The parallel terraced scan

The idea behind the parallel terraced scan is that of initially exploring many possibilities in parallel and gradually devoting more time and effort to those avenues that seem more promising. First, many possible solutions are explored in parallel to a shallow degree. As there is more information on what the possibilities are, the possibilities are eliminated to a smaller set and these are explored more in depth, spending more energy for each. This process continues until the most eligible possibility is found.

This is closely related to the exploration-versus-exploitation problem discussed by Holland (1975). This problem is typically illustrated by the "two-arm-bandit problem"[2] in the literature (Holland, 1975). Suppose there is a slot machine in a gambling casino with *two* payoff slots (unlike those at a casino with a single arm). Each arm has a different paying rate, let's say it pays off an amount every 5 times and the other once every 10 times, but we do not know which has a higher rate. We have a large, but finite, number of tokens and want to win back as many as possible. What we have to do first is to find out which arm gives back more. What strategy should we adopt? We can consider that some of our tokens will be used to *explore* the machine (in order to determine which arm pays more), while others will be used to *exploit* the machine (i.e., make the most money possible, based on the results of our exploration). The question then is: how many tokens should we devote to exploration, how many to exploitation? Assume we have 100 tokens. We might decide to use half of the tokens for exploration. So, we use 25 to test one arm and 25 to test the other. This "exploration" phase tells us that arm no. 1 seems to be better.

[2] A "one-armed bandit" is a colloquial term for a slot-machine in a casino in which players put coins, pull a long lever – the arm of the bandit – and hope to win more coins.

So, we put all 50 remaining tokens in the slot corresponding to that arm. But there are obvious problems with this strategy. For example, if one arm paid off every 2 times and the other every 50 times, we would quickly realize this and waste a lot of tokens exploring the arm that paid off only once every 50 times. On the other hand, if the difference in the payoffs for each arm was small, we might guess wrong after our fixed exploration phase and put all of our remaining tokens into the wrong slot. Holland (1975) has shown that there is an optimal iterative strategy in which one must put tokens in (i.e., explore) what appears to be the bad slot exponentially less often than what seems to be the good slot.

We can think of the parallel terraced scan in a mate-choice context as a machine that has many arms and we are “exploring” each of the arms by building partial representations of individuals of the opposite sex and then deciding whether or not to continue “exploring” that avenue.

At the beginning of the run, the individuals are all open-minded (Mitchell & Hostadter, 1990), and have no a priori opinion about potential mates. It does not matter whom the individual dates, because he/she is initially more interested in exploring potential mates than making a selection decision for a particular one. As the individual encounters and dates more potential mates, he or she starts to get an opinion of the values of potential mates, and starts to get choosier about whom to spend time and effort on. In other words, the encounter process goes from being initially random to more and more focused (i.e., less random) as the individual sees a number of people and begins to concentrate on those individuals that are the most interesting, spending less time on the less interesting alternatives.

Simão and Todd (2003) use a strategy where each individual sets an aspiration level with respect to his/her self-evaluation. The aspiration level is adjusted with respect to self-evaluation and the remaining reproductive lifetime of the individual. Courtship is a continuous process and individuals have the opportunity to switch to another individual depending on the new prospective partner's mate-value, the age of the individual and the length of the current relationship.

Context-dependent computational temperature ("choosiness")

To avoid any confusion related to terminology, temperature in the context of our model is a characteristic associated with *individuals*, rather than an entire population. (By contrast, we generally do not speak of individual water molecules as having a temperature. It is a value associated with collections of water molecules, for example, in a cup of coffee or in a swimming pool.) Here, temperature corresponds to an individual's willingness to explore the environment of potential mates. It is the computational inverse of choosiness. The higher an individual's temperature, the less he/she is choosy about a mate and vice-versa. Temperature falls as two individuals see each other more often and get to know one another (i.e., the desire to continue to explore the space of other potential partners falls) and increases with age and if one does not fall into a dating-relationship. Age is a particularly important factor in the determination of temperature; as time passes, the luxury to spend lots of time to find a suitable mate falls.

Context-dependent temperature is a measure of the degree of randomness used in decision making (Mitchell & Hofstadter, 1990; Mitchell, 1993; French, 1995). Temperature directly affects the decision about whom to go out with and whom to marry. When an individual's temperature is high, the individual differentiates less among the people and is open to date various potential mates (i.e., essentially random encounters, regardless of perceived mate-value). However, when the temperature is low, the differences among mates will begin to have an affect and the individual is much more likely to choose one individual over the other based, almost deterministically, if temperature is low enough, on mate-value. In other words, gradually the "encounter strategy" moves from an essentially random state to a biased random state, to a quasi-determined state.

This notion of context-dependent temperature is similar to the one in simulated annealing (Kirkpatrick, Gelatt & Vecchi, 1983) but it differs in at least one crucial aspect. In classic simulated annealing, temperature follows a predetermined "annealing" schedule where the value of temperature falls monotonically with respect to the number of iterations. By contrast, temperature in the present model reflects the individual's level of information of the mate distribution (potential mates' values and the range of those values), the state of his/her present relationship and the age of the individual. It is, in short, *an on-going feedback mechanism that determines how explorative or choosy the individual should be.*

In Simão and Todd's (2003) model the aspiration level is updated with respect to the remaining reproductive lifetime. This aims to simulate the time pressure to mate and corresponds to the effect of age on the temperature variable. The interactivity parameter that determines how many potential mates an individual encounters, like the computational temperature variable, determines the number of encounters an individual has in the present model.

3.1.3 Female Choice

Darwin (1859, 1871) observed that in nature the female ultimately makes the mating decisions. Males show off before a female in order to be chosen as her mate. Trivers (1972) later explained this as a manifestation of differential parental investment on the part of males and females. He claimed that the different amount of investment in offspring of each of the sexes produced such an asymmetry. Studies in social psychology support this asymmetry in humans in the initiation of a relationship (Buss & Schmitt, 1993; Clark, Shaver & Abrahams, 1999).

In our model we compare both cases. In the first case we assume asymmetry in the sexes and use a mechanism such that the male asks out for a date to a female and the females accept or decline the invitation. In the second case, there is sexual symmetry and both sexes can ask someone out, the individual who has been asked out can either accept or reject the offer. Simão and Todd (2003) assumed sexual symmetry, where individuals that ask each other out at the same time enter a courtship period.

3.1.4. Self-esteem

Self-esteem is how you evaluate yourself (Baumeister, 1999). It is the perception of oneself as good, bad, or mediocre. People have different levels of self-esteem for specific domains. An individual regard oneself as an excellent runner, a mediocre student, and a poor cook. Similarly, the sociometer theory suggests that self-esteem is a function of multiple indices of how a person stands in relation to those around him or her (Leary, Tambor, Terdal & Downs, 1995; Kirkpatrick & Ellis, 2001). Individuals become aware of and sensitive to their relative standing in terms of their mate-value with respect to the overall population: How attractive, in general, are they as a potential mate? Perceived mate-value has been found to be a predictor of self-esteem (Brase & Guy, 2004).

The sociometer in the sociometer theory can be explained by an analogy. It is like the gas gauge of a car (Kirkpatrick & Ellis, 2001): It does not have the function of running the car, but is an indicator of the gas level, warning the driver when the gas level becomes too low. The theory is based on evolutionary principles viewing the sociometer as a behavioral adaptation to the problem of social inclusion. Exclusion from a group would pose a significant threat to an individual's survival and would lead to his or her loss of the many benefits of living in a group in the context of the hunter-gatherer society. Continuous rejection by potential mates would pose a threat to that individual's reproductive success. A sociometer that monitors levels of acceptance and rejection from romantic partners may be useful for guiding one's mate selection strategy (Kirkpatrick & Ellis, 2001).

Self-perceived mate-value is determined by social feedback concerning one's attractiveness to the opposite sex. Key to this measure is one's previous history of successes and failures in mating, in combination with appraisals of the competition. In our model we have included a self-esteem variable for each individual. This value of this variable is adjusted by rejection and acceptance by potential mates.

Studies have shown that, regardless of people's degree of self-esteem, they want to succeed at work, to have intimate relationships, etc., but people with low self-esteem are less confident than their counterparts with high self-esteem that they will be able achieve this and are much more on guard against failure, rejection and other unpleasant outcomes (Baumeister, 1999). For example, if you ask someone out for a date and that person agrees, you may feel a gain in your self-esteem. On the other hand, if you are turned down, you are likely to loose self-esteem. So, asking someone out is risky, in terms of one's own esteem. If your self-esteem is already low, you might well not ask anyone out, to prevent the possibility of being rejected and your self-esteem falling even more.

Further, studies (e.g., Baumeister, 1999) have shown that people with high self-esteem are not necessarily more talented, intelligent, likeable and attractive, or superior. Rather, people with high self-esteem simply *believe* they are better. The difference seems to be mainly one of perception of self. Studies that ask unbiased judges to rate people's attractiveness conclude that people with high and low self-esteem are about equally attractive – but studies that ask people to rate their own physical appearance find that people with high self-esteem rate themselves to be significantly more attractive than people with low self-esteem. We view self-esteem as a measure of oneself and we included a mechanism where individuals with a low self-esteem avoid taking risks of asking out someone. What we have explicitly modeled as self-esteem (and its effect on behavior) is similar, although not identical, to the self-evaluation variable in Simão and Todd's (2003) model. How our self-esteem parameter is adjusted and its precise function will be discussed in detail section 3.2.2 below.

3.2 How the model works

In what follows we will provide a description of the operation of our mate-choice model.

The population consists of an equal number of females (*Fi*) and males (*Mi*).

The size of the population was set at 200. Individuals of both sexes have:

- 13 personal characteristics (*[C1 C2...C13]*),
- a corresponding set of 13 preference weights (*[W1 W2...W13]*), one for each characteristics of the opposite sex,
- a choosiness variable (*Choosiness*),
- an age variable (*Age*),
- a self-esteem variable (*SE*) and
- a record for each of the N (i.e., 5) most recently *encountered* individuals of the opposite sex (*[rM1 rM6 .... rM47] or [rF4 rF11 ....rF45])*
- a corresponding vector of values for the characteristics of each of these recently *encountered* persons
- a counter taking record of the encountered potential mates and how long the person will be remembered
- a counter taking record of the dated individuals and how long the person will be remembered

The information maintained by each individual is shown schematically in Figure 3.

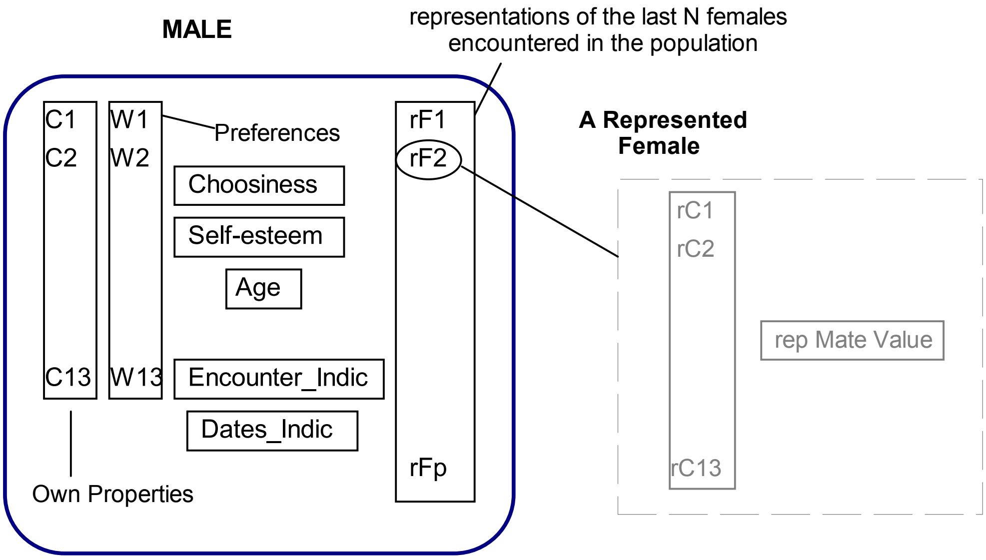

Figure 3. The sketch of a male individual.

The personal characteristics and preference weights are drawn randomly from a normal distribution around a default mean $\mu = 6$ and a standard deviation $\sigma = 2$. The mean and standard deviation values for the preference weights are generated with mean and standard deviations in accord with mate-preference findings (cf. §3.1.3.1, Table 2). The choosiness value depends on the age of the individual and his/her number of dates with the present date. As in the beginning, all of the individuals are single, the initial choosiness value is set with respect to the age in line with the choosiness formula which is described in §3.2.3. The initial age is uniformly randomly drawn from an interval from 14 to 17. The initial self-esteem value is set at 4. As the individuals start the simulation in their adolescence, they are assumed to have a slightly bad evaluation of themselves in the beginning (Baumeister, 1999).

For a given individual, the last N individuals he/she has encountered are recorded. The recorded information is their personal characteristic values (*[rC1 rC2 ..... rC13])* and their corresponding overall mate-value (*rMV*). In a departure from previous models of mate-choice, when an individual X encounters an individual Y, each individual only transmits part of his/her attribute values to the other individual. In this way, X's representation of Y (and Y's representation of X) is only gradually built up, becoming more and more accurate over time. For those characteristics about which there is not yet any information, this is filled in with the default value for that characteristic.

The model is basically designed with female choice and a multi-dimensional mate-value. To explore the effect of these two variables, two variations of the model will be tested. The first is the model with equal choice and the second is the model with a scalar mate-value (single-dimensional mate-value).

A run of the program

A single run of the simulation is equal to 200 iterations. Each iteration consists conceptually, of 3 stages — namely, the contact stage, the dating stage and, finally, the marrying stage. The first stage, the contact stage, is where the individuals encounter each other, and thereby, develop an impression of the encountered individuals. Information about a limited number of the 13 characteristics that characterize each individual is exchanged. The second stage, the dating stage, is where the individuals consider dating, ask each other out, date each other, and get to know each other a little better (i.e., exchange more information about themselves) if they date. The third stage is the marrying stage. If a dating pair marries, both individuals leave the population; if not, they continue to the next iteration where the previous stages are repeated. The program simulates the mating behavior of a

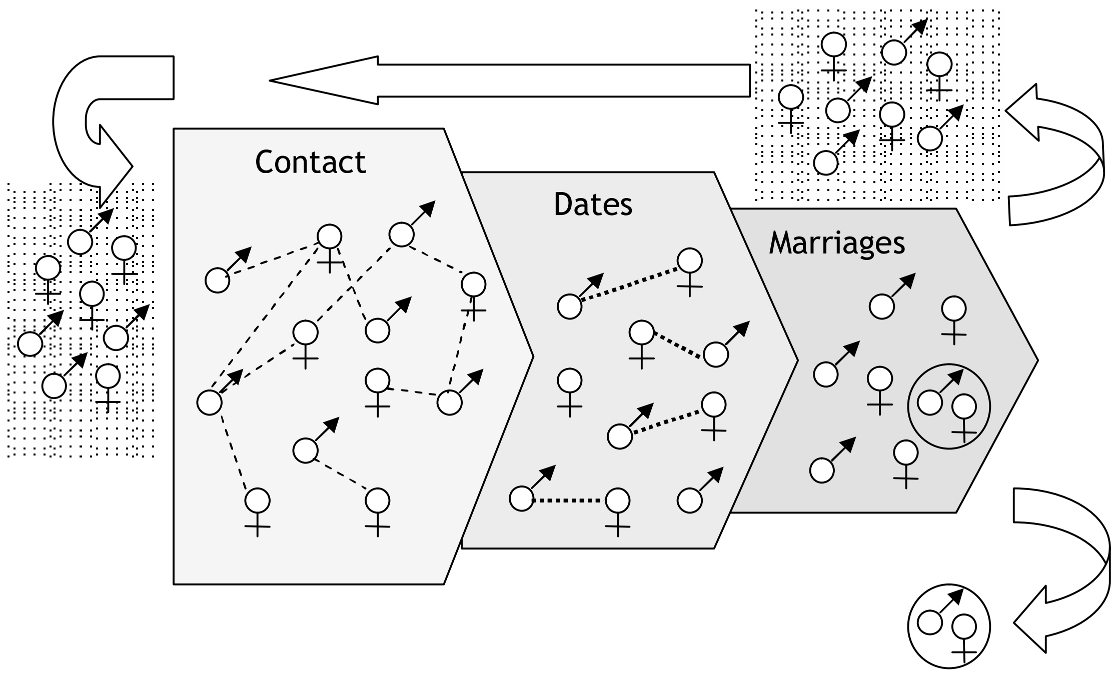

Figure 4: A visual sketch of the simulation.

population (and drop out of the pool because of age) or until they marry (at which point they also drop out of the mate-pool) where new individuals are added as individuals drop out of the pool because of age.

At the beginning of an iteration, each male will have a number of contacts with potential mates. The number of contacts is determined by the choosiness (inverse of temperature) variable of each individual. The lower the individual's choosiness, the more potential mates the individual encounters. The higher the choosiness, the fewer the number of potential mates they will encounter.

Each contact has a different duration, which is drawn from a uniform probability distribution with a lower bound of 1 and an upper bound of 4 encounters. A contact simulates a real life encounter, where the encounter could be at a friend's house, at a cafe, in a bus you take everyday, in short, any encounter with a member of the opposite sex where you have the chance of learning a little about the person. This is why, during each contact in the simulation, both males and females get some information about each other. The amount of information depends on the duration of the contact. For the simulation, an individual receiving information about another individual means receiving a number of personal characteristic values from the other individual.

However, as in real life, information transmission is not perfect. (Someone who appears very wealthy may turn out not to be all that rich.) Assume X meets Y. And Y has a value v_2 for characteristic 2 ("exciting personality"). There is a (small) Gaussian around v_2 such that the *transmitted* value, $\hat{v}_2$, is drawn from that distribution, meaning that there is some distortion of the actual value v_2 when it is acquired by X. Of course, after X has encountered Y a number of times and will have received a variety of values $\hat{v}_2$, all of which he averages. Consequently, X will

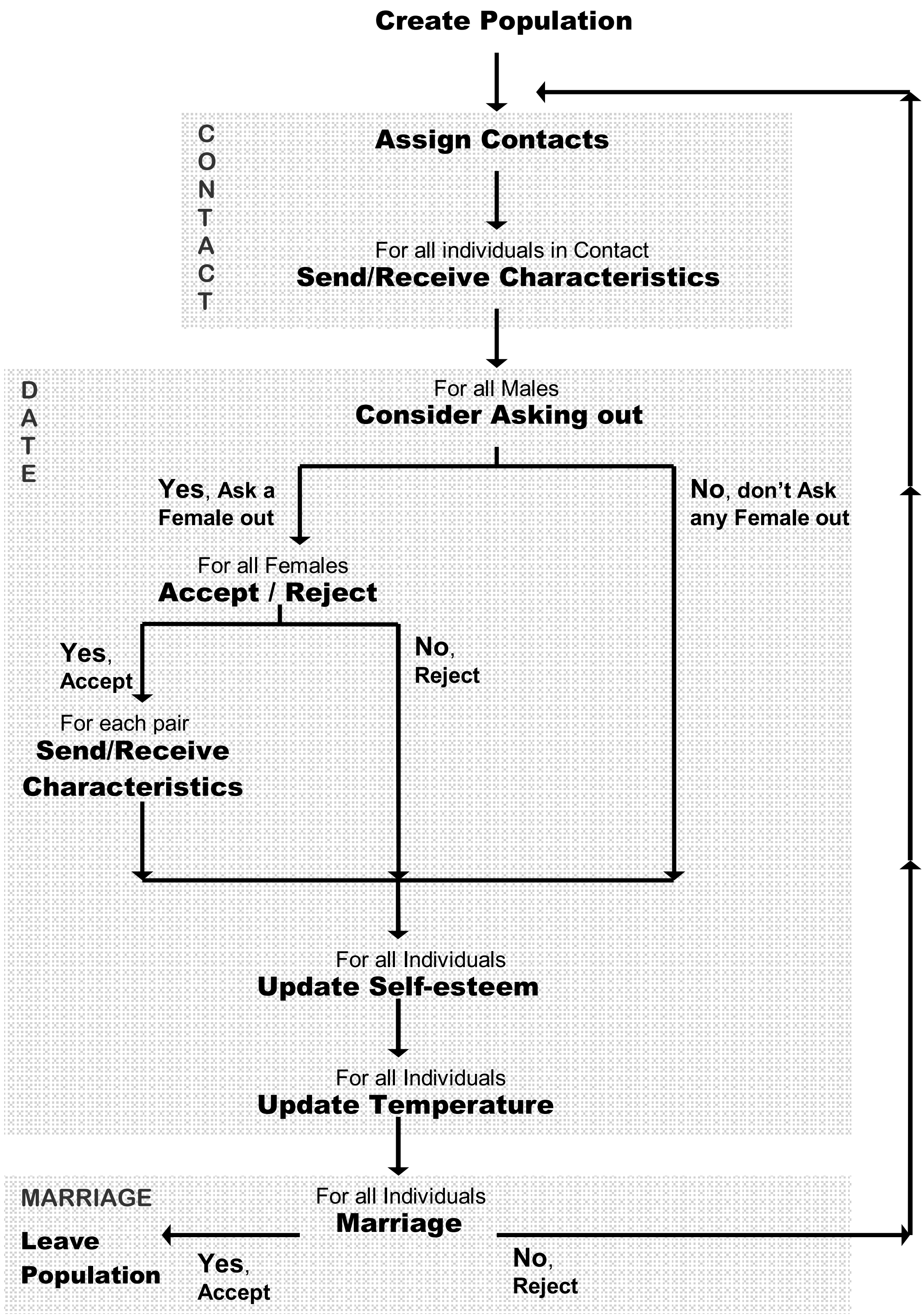

Figure 5. The algorithm.

soon have a good idea of the real value, v_2, for that characteristic in Y. We transmit $\hat{v}_2$ instead of v_2 because this better simulates our gathering of imperfect information (in this case, the true value v_2 combined with an assessment error). After each contact the representations of both individuals are updated.

Next, each male in the population considers whether or not he should ask one of the people he has encountered out for a date. He considers the women that he was in contact with for the last N iterations and makes his decision based on his temperature, his self-esteem, the mate-value of the potential mates and the overall average mate-values of the potential mates. This is where parallel terraced scan comes into play. The details are given in §3.2.1. If he decides to ask someone out, the female accepts or refuses his offer. The female makes the decision based on her own temperature, his mate-value, and the average of mate-values of the males she has recently dated. The mate-values that are used in making the decisions are the mate-values that have been updated upon the information received during the contact stage.

Depending on whether the woman accepts or refuses (or whether the male actually asked anyone out or not), the male's self-esteem variable is updated. The details are in 3.2.2. Choosiness (i.e., temperature) is also updated. Choosiness depends on the age of the individual and the consequent number of dates the individual has had with his/her present date. The details are given in sec. 3.2.3.

In the final part of an iteration, each individual checks whether he/she wants whether they marry their present date. If the temperature of both the man and the woman on the current date are below the threshold value, then they marry and leave the population. The individuals who are still single begin the contact-dating-marriage cycle again until a particular number of iterations is reached. Individuals who exceed a certain age are also taken out of the population (i.e., death in old age). New individuals are continually added to the population and ones who reach 65 are removed.

3.2.1 The Decision Criteria.

In the standard version of this model, men ask women out; women accept or decline their offers. A symmetric variation of this basic model allows for both men and women asking for dates. In what follows we will first describe the decision making processes as implemented in the standard model, then the equal choice variation. In the female choice version, each male must decide among his recent encounters, whom to ask out. Then the female either accepts or rejects the proposal. In the equal choice case, each sex must decide among their recent encounters, whom to ask out. Then, the proposed individuals either accept or reject.

The decision by the male: Asking for a date

This decision criterion is based on the parallel terraced scan. The individual is simultaneously evaluating a number of potential mates and deciding on one of them based on his temperature (i.e., choosiness) and on his current level of self-esteem. He is more likely to ask someone for a date when his temperature is high (i.e., when the value of his "choosiness" variable is low) and vice-versa. His decision, as with most decisions in this model, is made stochastically.

The male's decision is based on three factors: first the overall mate-value of the female (calculated from his perspective, i.e., based on his own preference profile), second his current self-esteem, and finally his current level of choosiness. Individual females with higher mate-values have, in general, a higher probability of being chosen (unless their temperature is very high – choosiness very low – in which case, it is essentially a uniform random choice). Self-esteem also affects the decision probability. Temperature makes the process of choosing a female either more random or more discriminating. The higher the self-esteem of the male, the more likely that he will actually act on his decision to ask one of the potential mates out.

The decision process consists of two steps.

1. First the male considers the group of females he has encountered in the last 5 contact steps. (For example, if there are on average 3 individuals encountered at each contact step, the number of individuals under consideration to be asked out would be 3*5=15 individuals). The probability that *Male i* will ask out *Female j* is calculated as follows:

$$P_{ij} = \frac{(MV_j)^{Ch_i}}{\sum_{n=1}^{R} (MV_{j_n})^{Ch_i}}$$

P_{ij} is the probability of the *Male i* to ask out *Female j*

MV_j is the mate-value of *Female j*

MV_n is the mate-value of *Female n*

j_n are the indices of the R most recently encountered females

Ch_i is the choosiness of *Male i.*

Note that when Ch_i is 1 (the default normal level of choosiness), then the probability of selecting of a potential mate Y is her mate-value divided by the sum of the mate-values of the last R females encountered by the male. On the other hand, if the value of Ch_i becomes very low ("don't care whom I ask out"), then selection is essentially independent of mate-value: any candidate is equally likely to be chosen. If, on the other hand, the value of Ch_i is high ("I only will ask out the best mate"), then the highest mate-value will completely dominate the other values and the female with this value will be chosen, essentially deterministically.

Once these probabilities are established, this leads to the selection of one woman to potentially ask out. But first we must consider his self-esteem before he actually acts on his selection.

2. Having made a choice of whom he will potentially ask out, the male must decide if he will actually go through with it. This is where self-esteem comes in. Only if the male's self-esteem is high enough, will he ask the chosen woman out. The male's self-esteem SE can range from 0 to 10 (the higher the number, the higher the male's self-esteem). A random number is chosen between 0 and 10 and if this number is less then SE, he asks the woman out. Otherwise, he does not.

The decision by the female: Accepting/rejecting an offer:

The male has asked a particular female out. She must now decide whether or not to accept. Her decision is a matter evaluating the current offer. We assume that she cannot postpone and choose among many offers. She must say yes or no to the current offer. This requires a different decision-making procedure than the "parallel" (i.e., choice among many) procedure used by the male to decide whom to ask out.

Her decision is based on the mate-value of the male and her choosiness. The decision mechanism for the female is as follows.

Each female has a record of the mate-values of her N (in this case, 5) most recent dates. If she has not yet had N dates, then she simply replaces the mate-values of "dates" with "males encountered." The mean and variance of the mate-values of these N males are used to calculate a normal distribution. This provides her with a rough estimate of the mate-values of the kind of males likely to ask her out and, most importantly, allows her to situate her new potential suitor with respect to males she has gone out with in the past. Thus, where ever her new suitor's mate-value falls on this curve, will determine the probability of her accepting a date with him. (To find the probability that she accepts, we integrate from $-\infty$ to wherever she places the mate-value of her present suitor on her mate-value curve (See Figure 6).

For the female who has just been asked out, the proposing male has a mate-value that is calculated, as always, by computing the inner product of her own preference-weight vector and his mate-value vector (i.e., the values of the 13 characteristics specified in Table 2). This value must now be adjusted based on her level of choosiness. If she is very choosy, she will adjust his MV to fall to the left of where it normally would, thus decreasing her probability of accepting his offer (see Fig. 4). If, on the other hand, she is not particularly choosy (she just wants to have a date with someone on a Friday night), she will adjust his mate-value upwards, thereby increasing the probability of her accepting his offer. The adjusted mate-value for a *Male i* who has asked out *Female j* is calculated by the following formula:

$$MV'_i = MV_i + k \times \ln(T_j + 1)$$

where:

MV_i' is the adjusted mate-value of *Male i* asking her out.

MV_i is the unadjusted mate-value of *Male i* asking her out.

Ch_j is the value of the Choosiness variable of *Female j* who has been asked out

k is a constant

The probability of accepting the offer to go out on a date by *Male i* is simply the area under the normal distribution probability density function (determined by the mate-values of her N last dates) from $-\infty$ to the adjusted MV for *Male i.* (Figure 6). Note that the female stores the real mate-value of the individual (not the adjusted mate-value) for purposes of evaluating subsequent suitors.

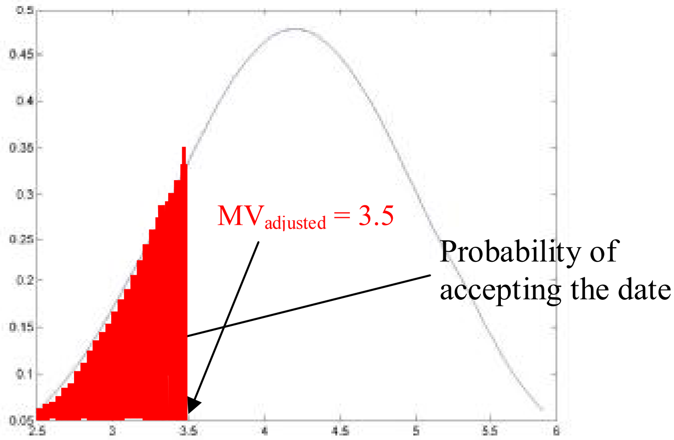

Figure 6. The probability of the female accepting the male.

Equal Choice Case for the Decision Criteria

In the equal-choice variation, members of both sexes can ask members of the opposite sex out. Each sex asks the potential mates out using the decision criteria of the male in the standard, female-choice-only case.

The equal choice case gets tricky – computationally speaking – at the step of accepting an offer. Everyone can ask someone out, and everyone can receive offers. First, each individual evaluates each propositions he or she receives by means of the same decision procedure used by females to accept or reject a male date propositions in the female-choice-only case. A problem arises, however, when an individual X accepts a date-proposition from an individual Y, but at the same time, X has also made a date-proposition to Z, who has accepted. So, does X go out with X or with Z? In the case of conflicts of this nature, the choice of an X-Y or an X-Z date is made randomly.

3.2.2 Adjusting Self-Esteem

After a male has selected the female he would potentially like to ask out, self-esteem becomes the main variable that affects the decision criteria for him to actually ask that person out (see above)." Self-esteem is a meter of self-evaluation that measures the individual's own mate-value. A male's self-esteem is adjusted by his success or failure at convincing women to go out with him. Previous experience will have an effect on future decisions to ask out other females.

In the standard form of the model where there is female choice, the woman's self-esteem plays no role. However in the equal choice variation of the model, the female has a self-esteem variable too. This self-esteem affects the decision criteria in the same way as in the female choice case and it is adjusted similarly.

Self-esteem (SE_m) ranges in value from 0 to 10. It is initialized at 5. Changes in self-esteem are calculated as follows. In this mechanism it is assumed that self-esteem is quantitatively in the same range as the mate-value.

1. If a male with a low self-esteem value asks out a female with higher mate-value ($MV_f > SE_m$)

If female accepts, then SE_m increases by a rate of 0.1 (i.e. multiplied by 1.1)

If female refuses, then SE_m decreases by a rate of 0.1 (i.e. multiplied by 0.9)

2. If a male with higher mate-value asks out female with lower mate-value ($MV_f < SE_m$)

If female accepts, then SE_m is the same

If female refuses, then SE_m decreases[3] by a rate of 0.2 (i.e. multiplied by 0.8)

If the person has not asked out anybody in that iteration and if his/her self-esteem is below the midvalue, his/her self-esteem value increases.

3. If an individual has not asked out

If male has a low self-esteem (lower than midvalue, i.e., 5), then it is increased by 0.015 (multiplied by 1.015)

3.2.3 Choosiness – Computational Temperature

Each agent has a *context-dependent computational temperature*. Temperature corresponds to the inverse of an individual's choosiness in mate-search and mate-selection. The lower the temperature, the higher the individual's choosiness and vice-versa.

---

[3] Note: We made the decrease in SE here greater than in case 1 because we reasoned that if a "low-self-esteem" female turns down a "high-status" male for a date, this is a significantly greater blow to the male's self-esteem than if a "high-status" female turns down a "low-self-esteem" male, something that the male might expect to happen.

Temperature is affected essentially by two factors — namely:

- Age of the individual
- Length of current relationship (i.e., number of dates with the same individual)

The decision to date becomes more deterministic as Temperature approaches 0 (i.e., choosiness goes up), and more uniformly randomly distributed as Temperature increases (i.e., as Choosiness decreases). This has been discussed in §3.2.1. The effect of the length of the current relationship increases as the individual gets older. The individual settles his decision faster than he/she would 10 years ago. The formulas for Temperature are as follows:

$$\gamma_i = \alpha_1 \times D \times (Age_i - \beta)$$

$$T_i = \frac{\dfrac{\alpha_2}{D \times ((Age_i - \beta) - \alpha_3)} + \alpha_4 \times \log(\alpha_5 \times D \times (Age_i - \beta))}{(N_i + \alpha_6)^{\gamma_i}}$$

where:

Age_i is the age of individual i,

T_i is the temperature of individual i,

N_i is the number of dates of individual i with his/her present date

γ_i is the exponent of the number of dates variable for individual i,

β shifts the curve in the horizontal direction,

D widens and narrows the curve,

$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$ are constants

The temperature function is shown in two formulas. The first formula determines how the number of dates affects the temperature formula. Notice that the first formula depends on the Age variable. As the individual gets older the number of dates will have a greater effect on the decrease on the temperature formula. In the second formula, the first term of the numerator (which includes the cubed age variable) corresponds to the first part of the T formula. The T variable starts off very high and decreases quickly. The second part of the numerator corresponds to the gradual rise after this decrease. The denominator actualized the effect of the N_i variable. Changing the parameters for the first term in the numerator corresponds to changing the rate of exploration individuals engage in, in their first part of their life. Changing the second term in the numerator corresponds to how fast individuals begin to increase their exploration activity as they approach the end of their reproductive lifetime. The temperature curve for a zero number of dates can be seen in Figure 7.

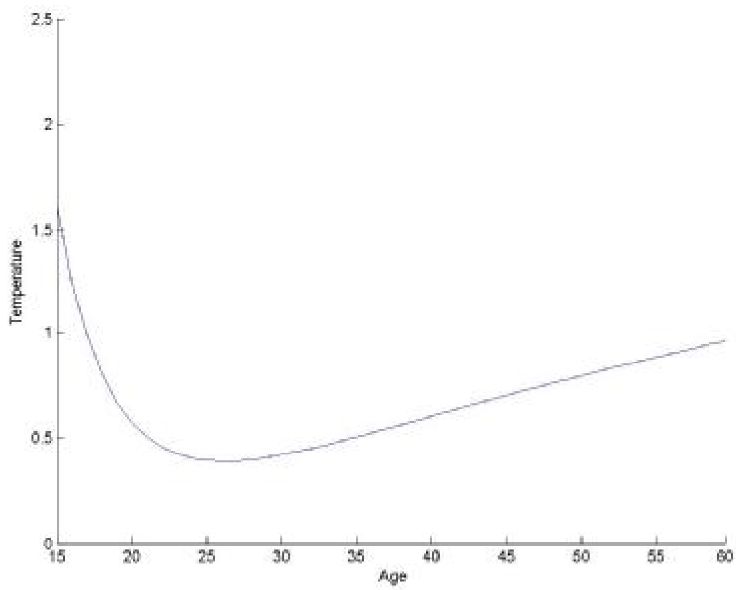

Figure 7. The temperature curve for $N_i=0$.

Below in Figure 8, is the graph of the T function with respect to the age of the individual and the number of consequent dates an individual had with his/her present date.

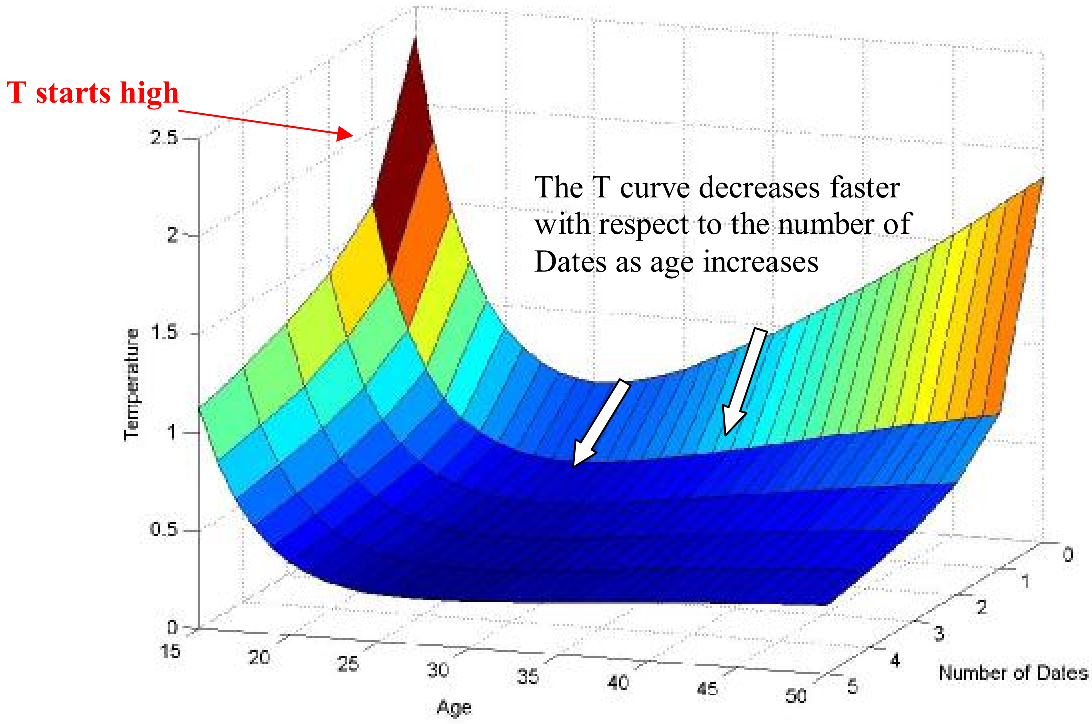

Figure 8. The Temperature vs. Age vs. Number of Dates graph.

The Temperature curve starts of very high and decreases fast. This corresponds to the first years of life when individuals are inclined to explore the potential mates rather than settle down in a relationship. There is another period of increase around age 20. This increase is a gradual increase that continues to the end of the life span. The individuals tend to be explorative as the lifetime for mating decreases. The number of dates has a higher influence on the T curve as individual gets older. This can be seen on Figure 8 from the changing steepness of the curve. The individuals marry when both of them have a sufficiently low temperature.

4. Results and Discussion

4.1. Technical Details

The model has been implemented in Matlab. A snapshot of the graphical user interface (GUI) is as below.

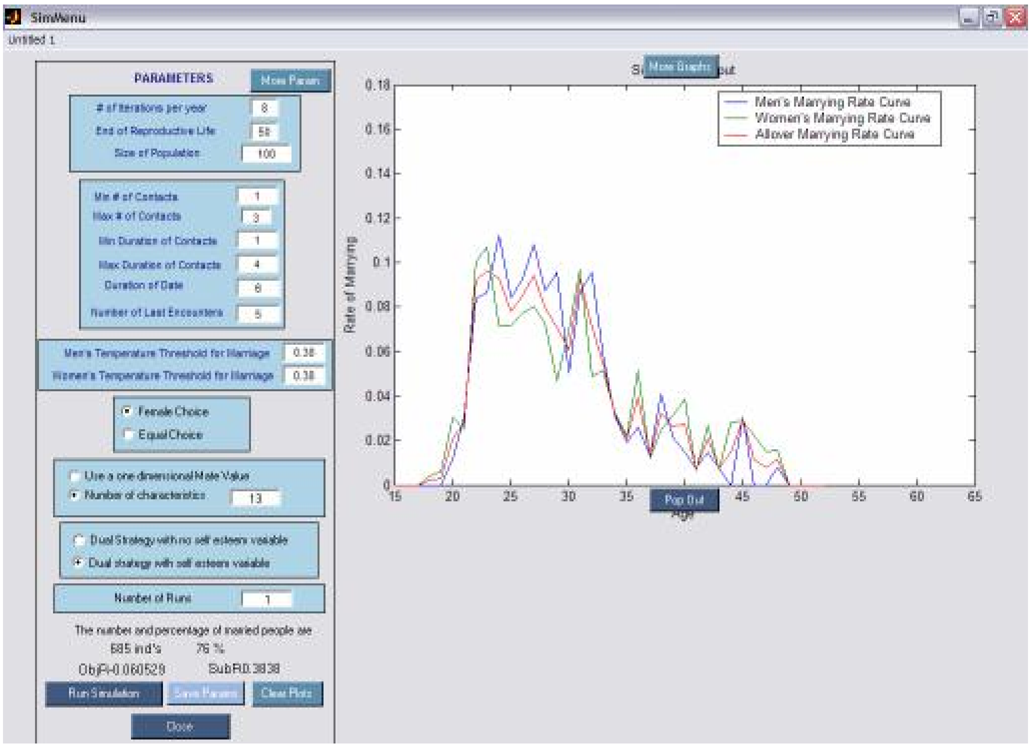

Figure 9. The snapshot of the GUI of the model.

4.2. Parameters

Some of the most important parameters with respect to their effect on the output are shown below in Table 3. In addition to these parameters, there are 5 parameters to adjust the self-esteem.

Parameters	*Description*	*Value*
Size_of_Population	Number of individuals in the population	100
Total_Inds_Generated	The total number of individuals generated throughout the simulation	1000
Female_Male_Ratio	Female to male ratio	1
Initial_Age_for_First	The range of age of the first individuals created	15-46
Initial_Age	The range of the individuals created during the simulation	15-17
Age_of_Death	The range of age that the individuals leave the population (apart from marriage)	48-52
Size_of_Characteristics	Number of characteristic of each individual	13 or 1
Male_Characteristics	Characteristic values for males	0-10 – Normal Dist Mean=6 Std=1
Female_Characteristics	Characteristic values for females	0-10 – Normal Dist Mean=6 Std=1
Male_Preference_Values	Preference weights for each male individual (see Table 4)	0-10 – (if scalar) Normal Dist Mean=5 Std=1, o/w see 0Table 4
Female_Preference_Values	Preference weights for each female individual (see Table 4)	0-10 – (if scalar) Normal Dist Mean=5 Std=1, o/w see Table 4
Distortion	The distortion while sending and receiving characteristics	0.1
Number_of_Last_Encounters	The number of last encounters that individuals remember the people they met	5
Minimum_No_of_Contacts/ Maximum_No_of_Contacts	Minimum/maximum of contacts that an male individual can get involved in the contact step	1 / 3
Minimum_Contact_Duration/ Maximum_Contact_Duration	Minimum/maximum duration of a contact that can happen	1 / 4
Date_Duration	Duration of a date	6
Temperature Parameters **β, D, α_1, α_2, α_3, α_4, α_5, α_6,**	The parameters for the Temperature curve	-
Duration_of_year	Number of cycle that a year lasts	8

Table 3. Parameter variables and the values to be used in the model.

Characteristics	*Female Mean Value (and SD Value) of Weights*	*Male Mean Value (and SD Value) of Weights*
Kindness and understanding	8.4 (1.2)	8.1 (2.0)
Exciting personality	7.5 (1.8)	7.2 (2.1)
Intelligent	7.4 (1.2)	7.1 (1.5)
Physical attractiveness	5.2 (1.9)	6.9 (1.8)
Good health	5.5 (2.0)	5.8 (1.9)
Adaptability	5.6 (2.3)	5.7 (2.0)
Creativity	4.2 (2.6)	3.4 (2.2)
Desire for children	3.2 (2.2)	3.8 (1.9)
College graduate	3.9 (2.1)	2.8 (1.7)
Good earning capacity	3.8 (2.0)	2.4 (1.7)
Good Heredity	2.1 (1.6)	2.5 (2.0)
Good housekeeper	1.9 (1.6)	2.1 (1.8)
Religious orientation	1.5 (2.4)	2.1 (2.7)

Table 4. The preference weights used in the model (same as Table 1).

One year is equal to four iterations of the simulation. One hundred individuals, of which fifty are female and the other fifty are male, are generated. The characteristics of each individual are generated from a normal distribution with a mean value of 6 and a standard deviation of one. The first one hundred individuals are generated with an age randomly distributed from a uniform distribution in the interval of 15-46. Individuals die at an age, again, assigned from a uniform distribution in the interval of 48-52. New individuals are generated with an age from a uniform distribution in the interval of 15-17 as individuals in the simulation die. The simulation population is constant at one hundred. The simulation continues until one thousand individuals are generated overall.

The data were generated from ten runs of each variation. The curves were smoothed with two-point smoothing using a two-point running average.

4.3 Testing model predictions

4.3.1 Distribution of ages at marriage

The population level outcomes are compared to first age marriage rate demographic findings of Norway in the year of 1978 and 1998. The method used in Simão & Todd (in press) is used for the evaluation of the simulation. The simulation output has been compared qualitatively to marriage rate data.

Demographic studies by Coale (1971) show that the distribution of age at first marriage follows a right-skewed bell curve in many cultures. (Coale, 1971; Todd, Billari & Simão, in press) The curve follows a bell shape with a long tail. The hazard rate of marriage curve used by Todd, Billari & Simão (in press) is used in this study. This rate is calculated by dividing the number of individuals that married at the age of x by the number of unmarried individuals at the age of x-1. The curve is the distribution of age at first marriage for Norway in the year 1978 and 1998.

4.3.1.1 The curve

The demographic data for the age at first marriage rate curve for the Norway population in 1978 is shown in Figure 10.

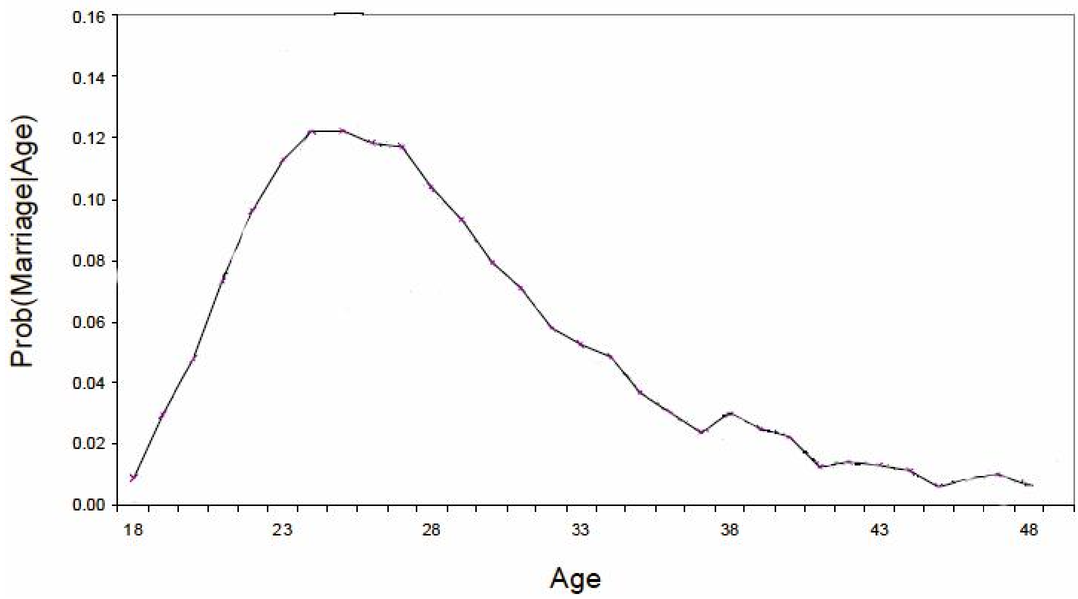

Figure 10. Marriage rate of Norway in 1978. Source: Todd, Billari, Simão (in press).

The parameters used for generating the corresponding curve are in Table 5.

Parameters	***Value***
Size_of_Characteristics	13
Type of Choice	Female Choice
Duration_of_year	8 runs
Temperature.α_1	0.017
Temperature.α_2	700
Temperature.α_3	8
Temperature.α_4	1.6
Temperature.α_5	0.051
Temperature.α_6	1.3
Temperature.β	-12
Temperature.D	0.55
Male Temperature Threshold	0.38
Female Temperature Threshold	0.38

Table 5. The parameter settings used for the generation of the standard curve.

The distribution of age at first marriage curve generated by the simulation is as below:

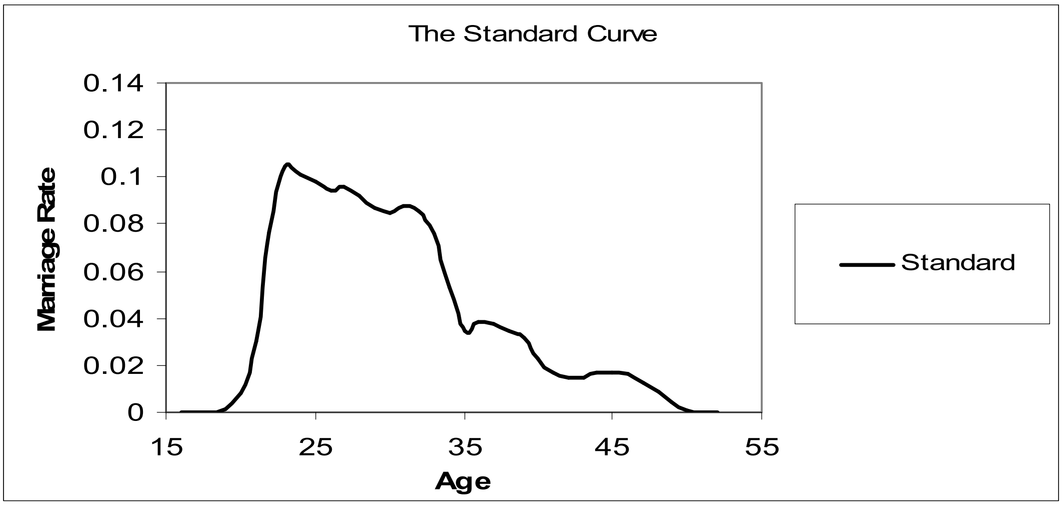

Figure 11. The marriage curve generated by the simulation.

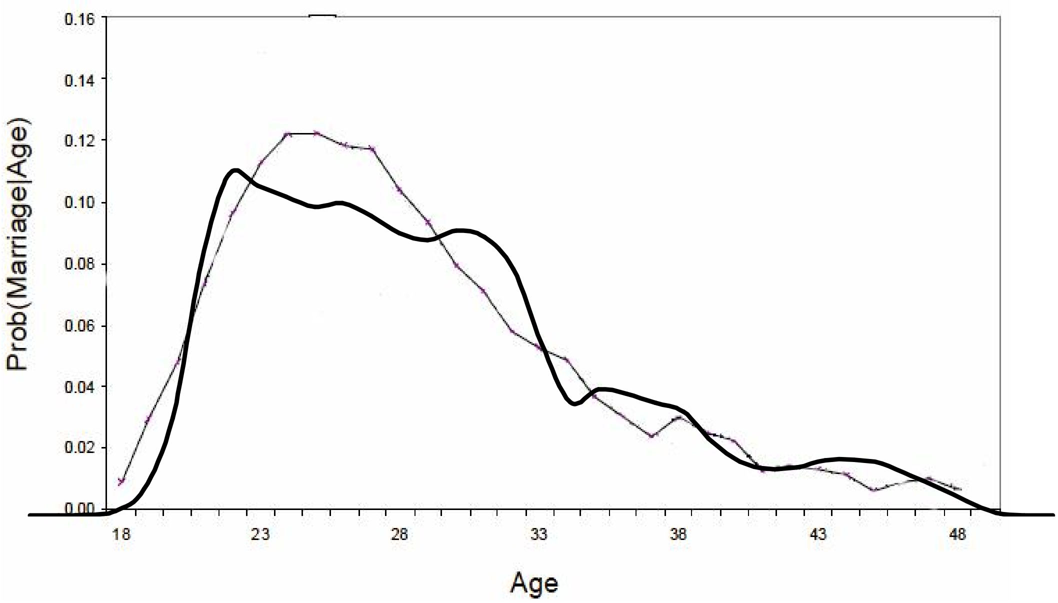

Figure 12. The marriage curve generated by the simulation and the Norway data.

When compared to the demographic data it can clearly be seen that the generated data qualitatively matches the actual data. See Figure 11 and Figure 12. The curve initially rises rapidly then gradually falls as the individuals get older. The simulation matches qualitatively the population-level patterns of mating. Simão and Todd (2002) reached a similar result. Their result can be seen in Figure 13.

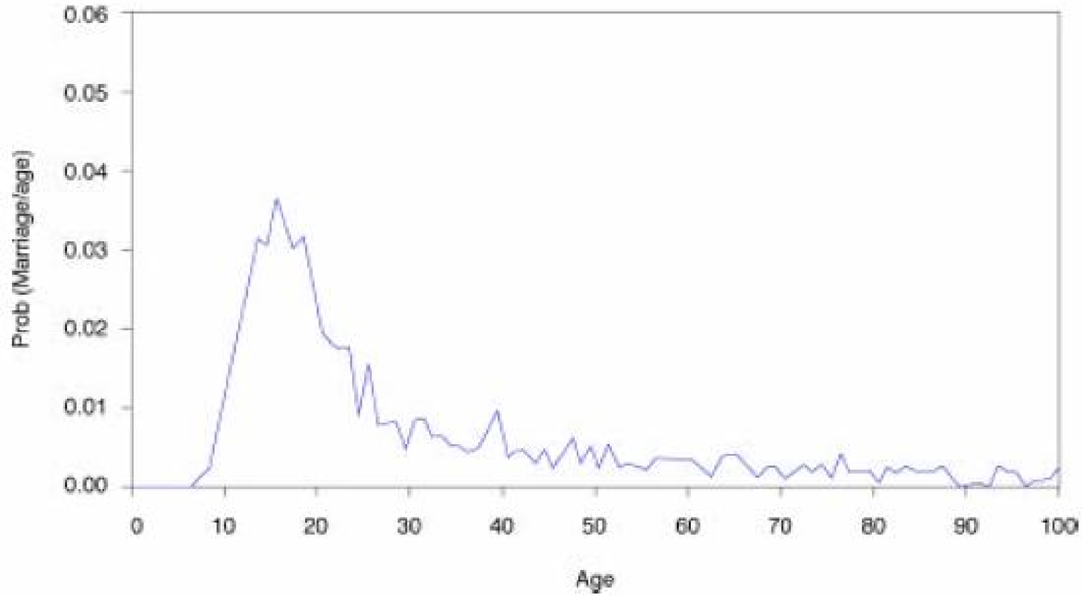

Figure 13. The generated data by Simão & Todd (in press).

4.3.1.2 The Sex Difference

Another property of the marriage curve is the shift of age of first marriages for the different sexes. See Figure 14.

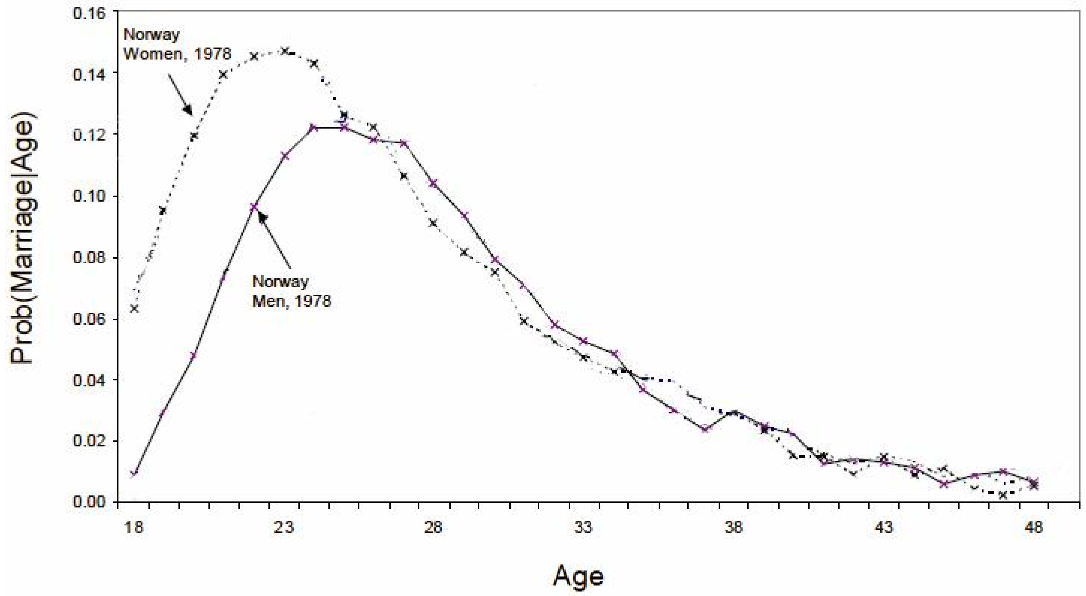

Figure 14. Marriage rate for women and men separately in Norway in 1978. Source: Todd, Billari, Simão (in press).

The same shift can be observed in the simulation output when the parameters of temperature are changed in accord. The D parameter of Temperature is set to -15 for the Males. The temperature curve for the male is shifted to the right, and the same shift occurs in the marriage curve. See Figure 14.

Figure 15. The marriage curves for females and males generated by the simulation.

The model allows us to manipulate parameters that could not be manipulated in former models of mate-choice. By changing the parameters of the temperature (i.e. "choosiness") curve, the shift of the curve for the two different sexes is simulated. The female curve is shifted leftwards with respect to the male curve. The reason for this is the shorter reproductive lifespan of the females with respect to males. By changing the parameters of the temperature curve as to simulate the higher time pressure for females, the marriage curve for females is shifted leftwards with respect to the male marriage curve. This shift hasn't been simulated by other mate choice models.

4.3.1.3 Norwegian Shift

By changing the parameters of T to reflect less pressure to be married early, the model reproduces the flattening of the Norway marriage-rate curves between 1978 and 1998 (See Figure 16). Note that no other structural changes in the model were needed to simulate the flattening of the curves that is observed in the empirical data. The curve was generated by lowering the marriage threshold for both the male and female to 0.32.

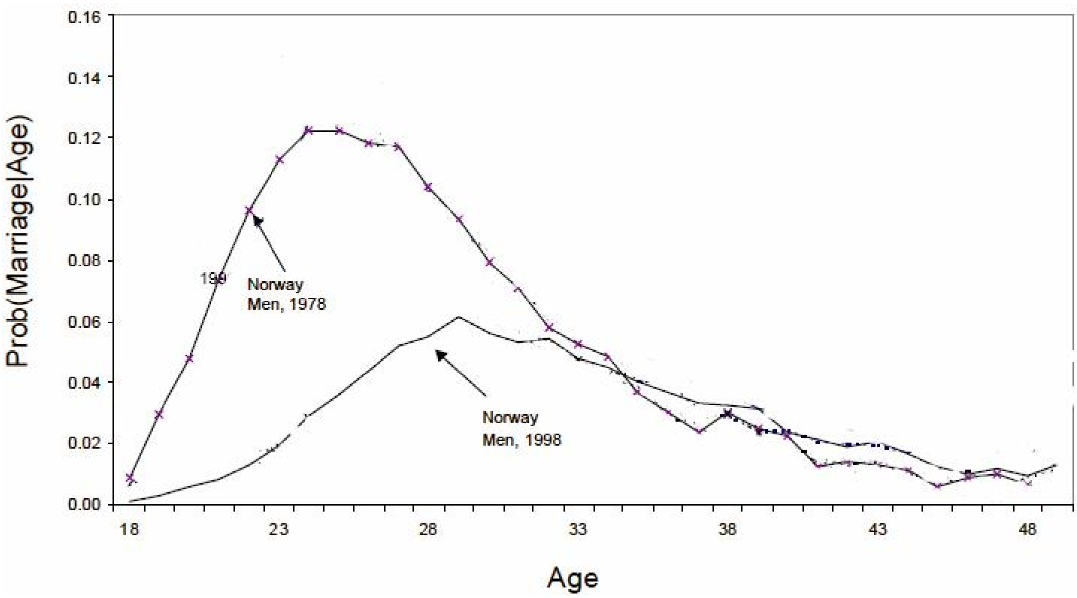

Figure 16. Marriage rate for 1978 and 1998 seperately in Norway. Source: Todd, Billari, Simão (in press).

The temperature threshold of marriage has been lowered to simulate this output.

The temperature curve captures the driving forces to marry for an individual. These could be either internal and external driving forces or a mixture of both. An example for an internal driving force would be the level of hormones which motivate sexual human behavior. An example for external driving forces would be the pressure

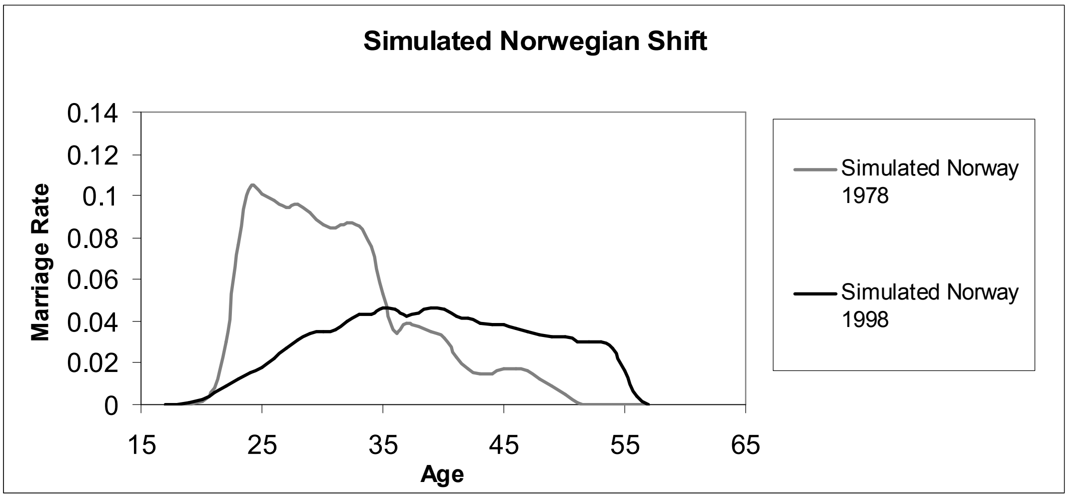

Figure 17. The two marriage curves generated by the simulation.

of society or simply the customs of the society. Though marriage is universal, there are variations of its actualization. While the Norwegian marriage curve in 1978 matches the more typical marriage patterns of other societies, the Norwegian marriage curve in 1998 is different in pattern. The reason for this is most likely the changing customs of the society. Our model is capable of producing a similar type of shift of the curve by changing the parameters of the temperature formula which is supposed to capture the new driving forces for marriage of the recent Norwegian society.

4.3.1.4 The Turkish Data

The Turkish data is in a different format than the data used for comparison. See Figure 18. However the universally observed curve tailing off to the right can still be observed.

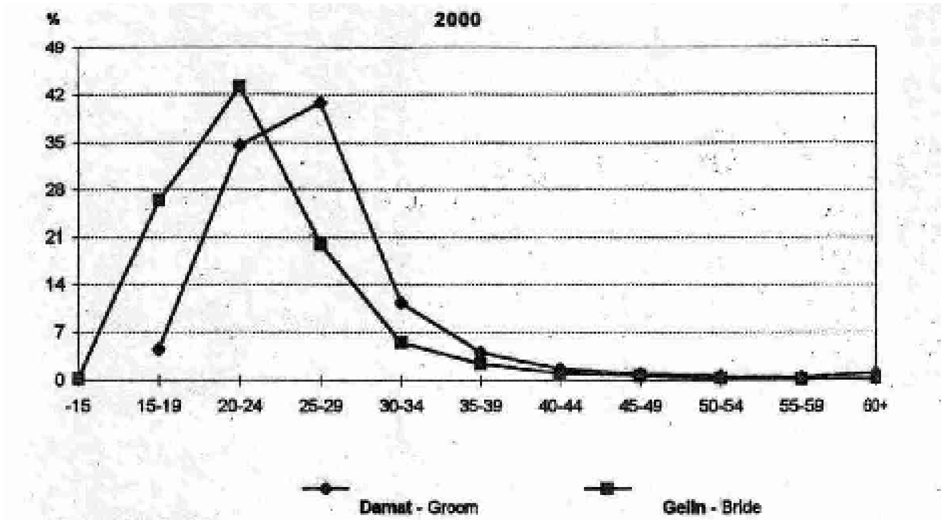

Figure 18. The Turkish data. (DİE, 2000)

4. 4 Comparison of the model with different parameter settings

To investigate the effect of the variables of the model, it has been tested with several variations of the standard model to analyze the performance of the model variables and properties. These variables have been chosen because they are the main

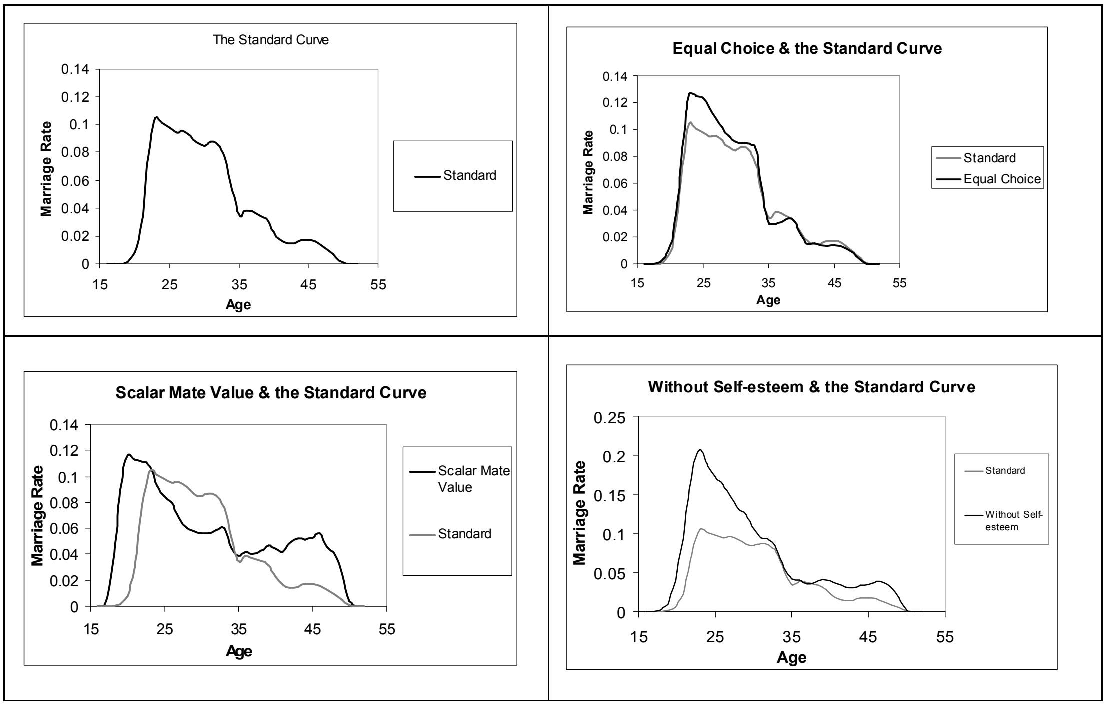

Figure 19. The marriage curves shown here are: i) the Standard Model, ii) the equal-choice variation, iii) the scalar mate-value variation, and iv) the variation without self-esteem.

value, equal choice and the no self-esteem variations. For an overview, the outputs can be seen together in a table in Figure 19.

4.4.1 Female Choice vs. Equal Choice

In the standard model, female choice was assumed based on theoretical evidence of evolutionary psychology and empirical findings on date initiation. A variation of the model with equal choice where both males and females ask out potential mates and are either accepted or rejected by those potential mates was set up. The parameters of the Standard Model were used.

This variation generated a higher curve of marrying rate (See Figure 20). The reason for this is that the search and exploration rate of the mate space has doubled as both sexes search. The marrying rate curve increases, yet, is not doubled as the individuals who are searching are at the same time the ones that are asked out. Thus, even though the search rate is doubled; the number of couples formed is increased but is not doubled.

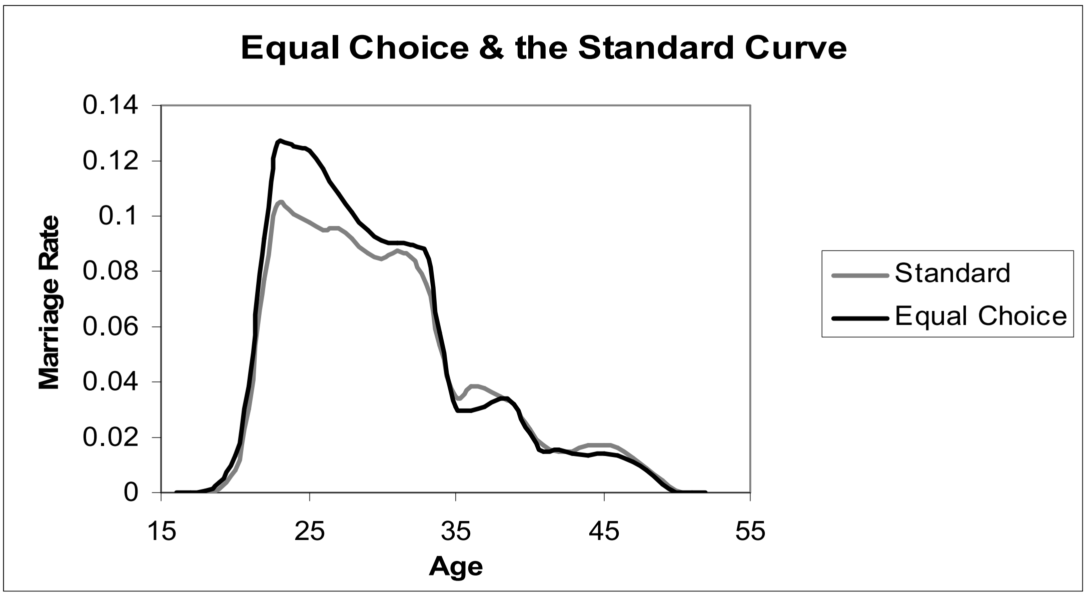

Figure 20. The curve generated with equal mate choice instead of female mate choice.

4.4.2 Multi-dimensional Mate-value vs. One-dimensional Mate-value

To find out the effects of a multi-dimensional mate-value with respect to a scalar mate-value another variation was set up. The curve can be seen in Figure 21. The marriage rate falls significantly for the scalar mate-value. The parameters of the Standard Model were used.

This variation produced a higher marriage rate curve shifted to the right indicating a higher rate of marriage and earlier marriage. This seems to suggest a faster and more efficient marriage mechanism. These results though are misleading. The computational temperature and parallel terraced scan mechanism is based on partial mate-value information. It is a way of efficiently making use of information where the search space is unknown and only partial information of individuals is available. When the mechanism is used with a scalar mate-value, it is operating with complete mate-value information. The individual receives complete information about the other individual at the first encounter. This is implausible with respect to humans.

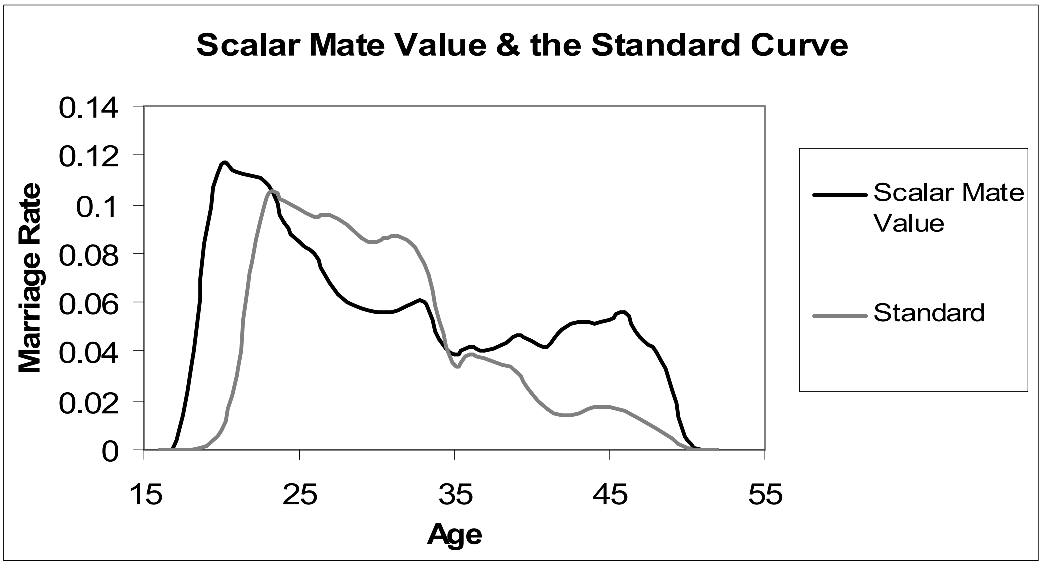

Figure 21. The curve generated with a scalar mate-value instead of a multi-dimensional mate-value.

4.4.3 Mate Search Strategy with Self-Esteem vs. Mate Search Strategy without Self-Esteem

The curve generated of the model without self-esteem can be seen in Figure 22. The parameters of the Standard Model were used. The curve rises very fast resulting in a high early marrying rate. The generated marriage rate is very high in this case. Self-esteem operates as a threshold avoiding individuals with a low self-esteem to ask out a potential mate. When it is removed the marriage rate increases, yet the intracouple correlation decreases resulting in less similar couples (see 4.4.4 for details).

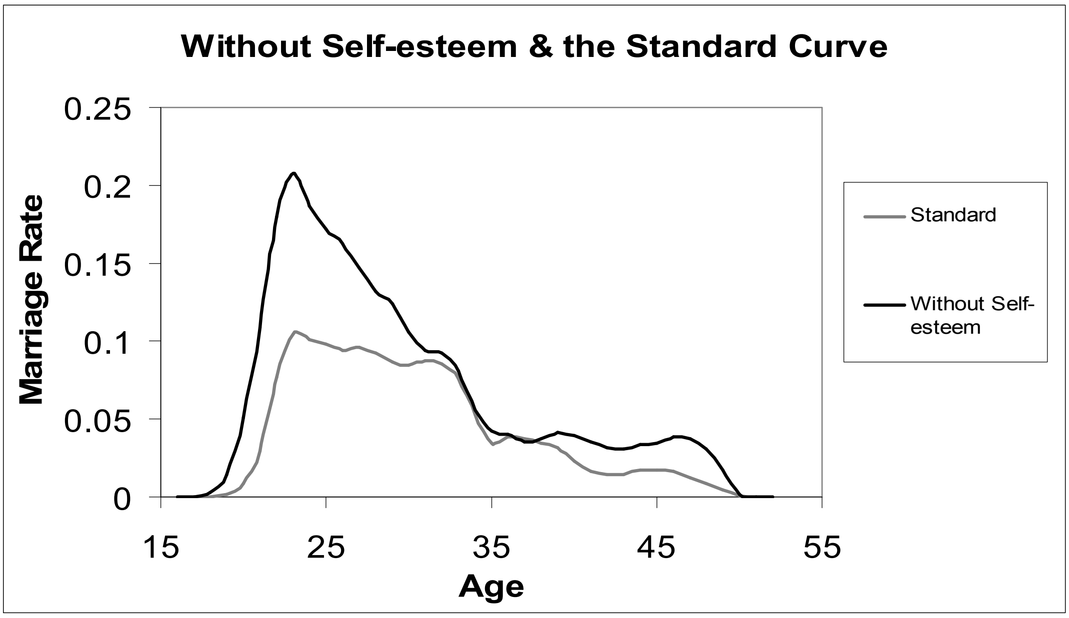

Figure 22. The curve generated without the self-esteem variable.

4.4.4 The correlations for each variation

Studies among couples show that there is similarity among couples. Below in Table 6 are the mean and standard deviation of the intracouple correlations based on the subjective mate-values for the curves above and relevant results from Simão & Todd's (2002) simulation and an empirical review by Kalick & Hamilton (1986):

Type of Parametric Variation	**Mean**	**Standard Deviation**
The Standard Model	0.28	0.08
Equal Choice Variation	0.32	0.05
Scalar Mate-value Variation	0.61	0.04
Without Self-esteem Variation	0.19	0.04
Simulated Correlations by Simão & Todd (2002)	*0.35 – 0.95 (changes with respect to parameter values)*	
Empirical Findings for physical attractiveness Kalick & Hamilton (1986)	*0.38 – 0.63 (changes with respect to research and level of commitment)*	

Table 6. The intracouple correlation values based on subjective mate-values.

These results were produced over 10 runs.

There is positive intracouple correlation (i.e. similarity) among couples of the simulated society. The equal choice variation generates a slightly higher intracouple correlation. The scalar mate-value variation generates a very high intracouple correlation; this is due to the availability of complete information at the first encounter of individuals. The intracouple correlation for the without-self-esteem variation is lower than all the other models.

Furthermore, in Table 6 the results can be compared to Simão & Todd's (2002) simulation results and the empirical findings reviewed by Kalick & Hamilton (1986). The results of the present simulation are in the similar range as in Simão and Todd's simulation. Again the present simulation results are in the similar range of the empirical data. The data reviewed in Kalick and Hamilton is the correlation of the physical attractiveness among couples.

4.4.5 The percentage of Marriage

Studies show that percentage marriages in populations are usually around 85-95 %. Simão and Todd (2002) were able to generate such a high percentage of marriage. In this model the percentages are as below:

Type of Parametric Variation	Mean %	Standard Deviation
The Standard Model	78.4	2.3
Equal Choice Variation	82.1	1.5
Scalar Mate-value Variation	94.9	1.1
Without Self-esteem Variation	95.4	0.7

Table 7. The percent that got married.

4.5 Potential sources of criticism

The use of some of the empirical findings in building the model might be criticized. The empirical findings on preferences was based on an American group of people while the empirical data which was used for the comparison of the generated output was from a Norwegian population. This can be justified by the way the data are used. What is of our concern is the quantitative property of those data. As for the preferences, the overall tendency for ranking some characteristics over the others is important. Whether the highest ranked characteristics is intelligence or beauty is of secondary concern. As for the marriage curve, the universal shape of the curve raising fast and then tailing off is important while the exact location of the peak and the overall height of the curve is, at this level of analysis, not of our interest.

One might ask why love is never mentioned in any stage of the model. It might give the impression that love is irrelevant to mate choice. This study is on the mechanism of finding a mate and it is assumed that these mechanisms were shaped through evolutionary mechanisms which operate mostly unconsciously. Emotions, in this case love, are the executers of these evolutionary mechanisms (Wright, 1995). They take their share in our decision processes to choose the ‘correct’ mate along with the conscious decision mechanisms.

5. Conclusions

5.1 Summary of the Contributions of the Model

We have shown that this model gives plausible results that qualitatively match empirical studies on patterns of human mating as expressed in marriage-rate data.

The five major and novel contributions of this model are:

- computational temperature as a measure of choosiness: the use of a context-dependent computational temperature algorithm that closely corresponds to what is called in the literature *choosiness*.
- for each individual, a multi-dimensional vector of mate-value for a variety of characteristics, instead of a single, overall mate-value describing the individual.
- changing representations: the use of representational structures of potential mates that, like those in real humans, change over time and become more detailed and complete through repeated contact with the person
- subjective mate-values: the use of mate-values that depend on an individual preference profile of each individual. Thus an individual woman, who has a high mate-value for a particular male, may have a significantly different mate-value for another male with a different preference profile.
- self-esteem: the use of self-esteem as an internal gauge measuring the mate-value of the individual with respect to potential mates.

There are not any other current models of mate-choice that include these mechanisms. The inclusion of these mechanisms allows us to be able to have a much more psychologically plausible model. In addition, it allows us to manipulate parameters that could not be manipulated in former models of mate-choice.

5.2. Future work

The simulation can be extended in certain ways.

First, the characteristics can be assumed to have different degrees of visibility potential mates. In the model, during the encounters and dates, characteristics of the individual are sent and received to the potential mate randomly. However, some characteristics require different times to assess: physical beauty can be perceived immediately, while assessing personality requires weeks (Miller & Todd, 1998). The model can be extended so that the characteristics revealed change with respect to different consecutive numbers (i.e. stages) of dating.

Second, it is assumed that there is no relation between one's own characteristics and one's preferences. Further extension can be done, after investigating the effect of characteristics of one's preferences. Most probably some characteristics will be of type preference (preference for the best) and some of homotypic preference (like prefers like). The multi-dimensional mate-value will allow us to make this distinction.

Third, studies on self-esteem over the lifespan show that self-esteem has a certain trend through the lifespan (Trzesniewski, Donnellan & Robbins, 2003). In this model, self-esteem values fluctuate through the lifespan. Integrating the empirical finding on stable self-esteem into the model might enhance the output of the model.

REFERENCES

Aron, A. (1988). The matching hypothesis reconsidered again – comment on Kalick and Hamilton. *Journal of Personality and Social Psychology, 54(3),* 441-446.

Alpern S., Reyniers D. (1999) Strategic mating with homotypic preferences. *Journal Of Theoretical Biology*, 198 (1): 71-88 May 7.

Baumeister, R.F. (1999). Self-concept, self-esteem, and identity. In In V. Derlega, B. Winstead, & W. Jones (Eds.), *Personality: Contemporary Theory and Research* (2nd Edition, pp 246-280). Chicago, IL: Nelson-Hall.

Bechtel W, Abrahamsen A, Graham G. (1998). The life of cognitive science. In: Bechtel W, Graham G, Balota DA (eds). A Companion to Cognitive Science. Malden, Mass.: Blackwell, 2–104.

Bergstrom, C., & Real, L. (2000). Towards a theory of mutual mate choice: Lessons from two-sided matching. *Evolutionary Ecology Research*, 2(4), 493-508.

Bonabeau E, Meyer (2001) C. Swarm intelligence. A whole new way to think about business. *Harvard Business Review*, 79(5), 106-114, 165.

Brase G.L., Guy, E.C. (2004). The demographics of mate value and self-esteem. *Personality and Individual Differences,* 36(2): 471-484.

Buss, M. D., Barnes, M. (1986). Preferences in Human Mate Selection, *Journal of Personality and Social Psychology*, 50(3), 559-570.

Buss, D. M. (1989). Sex differences in human mate selection: Evolutionary hypotheses tested in 37 cultures. Behavioral and Brain Sciences, 12(1), 1-49.

Buss, D. M., Kenrick, D.T. (1998). Evolutionary social psychology. In D. T. Gilbert,

S. T. Fiske, & G. Lindzey (Eds.), The handbook of social psychology (4th ed., Vol. 2, pp. 982-1026). Boston: McGraw-Hill.

Buss, D. M., Schmitt, D. P. (1993). Sexual Strategies Theory: A contextual evolutionary analysis of human mating. *Psychological Review*, 100, 204-232.

Clark, C. L., Shaver, P. R., & Abrahams, M. F. (1999). Strategic behaviors in romantic relationship initiation. *Personality and Social Psychology Bulletin*, 25, 707-720.

Coale, A. (1971). Age Patterns of Marriage . *Population Studies* 25 (2), pp. 193-214.

Cosmides, L., & Tooby., J. (1997, January) Evolutionary Psychology: A Primer. *CogWeb: Cognitive Cultural Studies*, Last update 1998. Retrieved from http://cogweb.ucla.edu/ep/EP-primer.html on 11 Dec. 2004.

Darwin, C. (1959) *On the Origin of Species.* London: Harvard University Press.

Darwin, C. (1871). *The Descent of Man, and Selection in Relation to Sex,* John Murray.

DİE (2000) *Evlenme İstatistikleri*. Devlet İstatistik Enstitüsü.

Ellis, B. J. (1992). The evolution of sexual attraction: evaluative mechanisms in women. In Barkow, Cosmides, and Tooby (Eds.), *The adapted mind: Evolutionary psychology and the generation of culture*. New York: Oxford University Press.

French, R. M. and Cleeremans, A. (1996). Probing and Prediction: A pragmatic view of cognitive modeling.

French, R. M. (1995) . *The Subtlety of Sameness*. Cambridge, MA: The MIT Press.

Fiske, S. T. (2004). Social beings: A core motives approach to social psychology. New York: Wiley.

Gilbert, N., Troitzsch, K, (1999) *Simulation for the Social Scientist,* Open Univ Press.

Gilbert, G.N. (1997). A simulation of the structure of academic science. *Sociological Research Online* <http://www.socresonline.org.uk/2/2/3.html>, 2(2), pp1-2.

Hofstadter and the Fluid Analogies Research Group (1995). *Fluid Concepts and Creative Analogies.* New York: Basic Books.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press.

Jennions M.D., Petrie M. (1997). Variation in mate choice and mating preferences: a review of causes and consequences. *Biological Reviews of the Cambridge Philosophical Society*, 72, 283-327.

Johnstone, R. (1997). The tactics of mutual mate choice and competitive search. *Behavioral Ecology and Sociobiology,* 40(1), 51-59.

Kalick, S. M., Hamilton, T. E. (1986). The matching hypothesis reexamined, *Journal of Personality and Social Psychology, 51(4),* 673-682.

Kenrick, D.T. (1994). Evolutionary social psychology: From sexual selection to social cognition. In M.P. Zanna (Ed.) *Advances in Experimental Social Psychology*, 26. (pp. 75-121). San Diego, CA: Academic Press.

Kenrick, D.T., Li, N.L., & Butner, J. (2003). Dynamical evolutionary psychology: Individual decision rules and emergent social norms. *Psychological Review*, 110(1), 3-28.

Kirkpatrick, L. A., & Ellis, B. J. (2001). An evolutionary-psychological approach to self-esteem: Multiple domains and multiple functions. In G. J. O. Fletcher & M. S. Clark (Eds.), *Blackwell handbook of social psychology: Interpersonal processes* (pp. 411–436). Oxford, England: Blackwell Publishers.

Kirkpatrick, S, Gelatt, C.D., Vecchi, M.P. (1983). Optimization by Simulated Annealing. *Science* 220, 671.

Leary, M. R., Tambor, E. S., Terdal, S. K., & Downs, D. L. (1995). Self-esteem as an interpersonal monitor: The sociometer hypothesis. *Journal of Personality and Social Psychology,* 68, 518–530.

Levine, D. K. (2001). Game Theory. *Encyclopedia of Cognitive Science*, NaturePublishing Group: Basingstoke.

Luttbeg, B. (1996) A comparative Bayes tactic for mate assessment and choice. *Behavioral. Ecology*. 7, 451-460.

Macy, M.W., Willer, R. (2002). From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Annual Review of Sociology*, Vol. 28, 2002

Mazalov, V., Perrin, N., Dombrovsky, Y. (1996) Adaptive Search and Information Updating In Sequential Mate Choice. *American Naturalist* 148 (1): 123-137 July.

Miller, G. F. (1998). How mate choice shaped human nature: A review of sexual selection and human evolution. In C. Crawford & D. Krebs (Eds.), Handbook of evolutionary psychology: Ideas, issues, and applications (pp. 87-130). Lawrence Erlbaum.

Miller, G. F. (2000). *The mating mind: How sexual choice shaped the evolution of human nature.* New York: Doubleday.

Miller, G. F., Todd, P. M. (1998). Mate choice turns cognitive. *Trends in Cognitive Sciences*, 2, 190-198.

Mitchell, M. and Hofstadter, D. R. (1990). The emergence of understanding in a computer model of concepts and analogy-making. *Physica D*, 42, 322-334.

Mitchell, M. (1993). *Analogy-making as Perception.* Cambridge: The MIT Press.

McNamara, J., Collins, E. (1990). The job search problem as an employer candidate game. *Journal of Applied Probability, 27(4),* 815-827.

Parker (1983). Mate quality and mating decisions. In P. Bateson (Ed.), *Mate Choice* (pp. 141-166). New York: Cambridge University Press.

Russel, S., Norvig, P. (2003). *Artificial Intelligence: A modern Approach*. Prentice Hall International Editions.

Simão,J., Todd, P. M. (2001). A Model of Human Mate Choice with Courtship that Predicts Population Patterns, in "*Advances in Artificial Life, 6th. European Conference, ECAL 2001, Prague, Czech Republic*", Jozef Kelemen, Petr Sosík (Eds.), Springer, Lecture Notes in Artificial Intelligence 2159.

Simão,J., Todd, P. M. (2002). Modeling Mate Choice in Monogamous Mating Systems with Courtship, *Journal of Adaptive Behavior,* 10(2).

Simão,J., Todd, P. M. (2003). Emergent Patterns of Mate Choice in Human Populations, special issue of *Journal of Artificial Life,* 9(4).

Sperber, D. (1999) Culture, cognition and evolution. In *MIT encyclopedia of cognitive sciences*, R. A. Wilson, R. A. and Keil, F. C., (Eds.). Cambridge, MA: MIT Press. pp. cxi-cxxxii.

Symons, D. (1987) Can Darwin's view of life shed light on human sexuality? In J.H. Geer and W.T. O'Donohue (Eds.) *Theories of human sexuality*. New York: Plenum.

Todd, P.M., Billari, F.C., and Simão, J. (in press). Modeling the emergence of social marriage patterns produced by individual mate-search heuristics. *Demography*.

Todd, P.M., Miller, G.F. (1999). From pride and prejudice to persuasion: Satisficing in mate search. In G. Gigerenzer, P.M. Todd and the ABC Research Group, Simple Heuristics that make us smart (pp. 287-308). New York: Oxford University Press.

Tooby, J. & Cosmides, L. (1992) The psychological foundations of culture. In *The adapted mind: Evolutionary psychology and the generation of culture* (ed. J. Barkow, L. Cosmides, & J. Tooby), pp. 19-136. NY: Oxford University Press.

Trivers, R. L. (1972) Parental investment and sexual selection. In: Campbell B (ed.) *Sexual Selection and the Descent of Man 1871-1971*. Aldine, Chicago, IL.

Trzesniewski, K. H., Donnellan, M. B., & Robins, R. W. (2003). Stability of self-esteem across the lifespan. Journal of Personality and Social Psychology, 84, 205-220.

Widemo, F., S. A. Sæther. (1999). Beauty is in the eye of the beholder: causes and consequences of variation in mating preferences. *Trends in Ecology and Evolution*,14: 26-31.

Wright,R. (1995) *The Moral Animal.* Pantheon Books (and London, Abacus, 1995).

APPENDIX

Appendix. The code of the simulation software.

The code consists of several files including one or several functions.

The file containing the main function is simmainGUI.m . The following files are the important functions used in simmainGUI.m or the functions used by the functions in simmainGUI.m. The last file is the SimMenu.m file which contains the GUI functions and its subfunctions, and the parameter settings.

file simmainGUI.m

```
%function names begin with a lower case letter, (e.g. update_Representation() )
%Variable names begin with a upper case letter (e.g. Durations)
%Variable names in a struct data structure begin with a lower case letter
%(e.g. Population.man -man is a variable)

%simmain is the main function, this is where the simulation runs
function Output=simmainGUI(handles)
%SIMMAIN    Run the simulation.
%   SIMMAIN runs the simulation

% Start the stopwatch timer
tic;
disp('Program started.')

%Parameters is a struct that contains all the parameters for the simulation
%
%
%          |-- Par_simmainGUI -- Max_Iterations
%          |
%          |                      |-- Size_of_Population
%          |                      |
%          |                      |-- Female_Male_Ratio
%          |                      |
%          |                      |-- Number_of_Last_Encounters
%          |                      |
%          |-- Par_initiate_Population--|-- MateValueDimensions
%          |                      |
%          |                      |-- Size_of_Characteristics
%          |                      |
%          |                      |-- Male_Preference_Values
%          |                      |
%          |                      |-- Female_Preference_Values
%          |
%          |                   |-- Minimum_Number_of_Contacts
%          |                   |
%          |                   |-- Maximum_Number_of_Contacts
%          |-- Par_assign_Contacts--|
%          |                   |-- Miniumum_Contact_Duration
%Paramaters--|                      |
%          |                   |-- Maxiumum_Contact_Duration
%          |
%          |-- Par_contact-- Number_of_Last_Encounters
%          |
%          |-- Par_date--Date_Duration
```

```
%           |
%           |-- Par_get_older -- Duration_of_year
%           |
%           |-- Par_simMenu -- LifespanAfterAdolescence
%           |
%           |                    |-- femaleID
%           |-- Par_plot_one_ind --|
%                                |-- maleID
%
%Initiate simulation by creating the population
%Population is a struct which consists of the man and woman structs.
%
%           |-- men
%           |(subpopulation struct)
%Population--|
%           |
%           |-- women
%            (subpopulation struct)
%
%The index of each individual in the men and women array is the id of the
%individual
%
% Each subpopulation struct is as follows
%
%              |-- sex: [50x6 char]
%              |
%              |-- characteristics: [50x m double]
%              |
%              |-- preferences: [50x m double]
%              |
%              |-- temperature: [50x1 double]
%              |
%              |-- age: [50x1 double]
%              |
% Subpopulation--|-- maritial_Status: [50x6 char]
%              |
%              |-- represented_Subpopulation: [50x1 struct]
%              |
%              |-- experience: [50x1 double]
%              |
%              |-- last_Encounters: [50xn double]
%              |
%              |-- last_Date: [50x2 double]
%
% Each represented_Subpopulation is as follows
%
%                       |-- represented_Characteristics
%                       |
%                       |-- represented_mateValues
% represented_Subpopulation--|
%                       |-- dates
%                       |
%                       |-- encounter_Indicator
%
%declare the global variables
global COUNTER;
global MARRIED_COUPLES_COUNTER;
global MALE_ID_COUNTER;
global FEMALE_ID_COUNTER;
global SINGLE_DEAD_MEN_COUNTER;
global SINGLE_DEAD_WOMEN_COUNTER;
global MARRIED_DEAD_MEN_COUNTER;
global MARRIED_DEAD_WOMEN_COUNTER;

Parameters=handles.Parameters;
```

```
%Convert the items of the Parameters struct to variables for efficiency
Max_Iterations= 1000;
Size_of_Population=Parameters.Par_initiate_Population.Size_of_Population;
Female_Male_Ratio=Parameters.Par_initiate_Population.Female_Male_Ratio;
Male_Population=Size_of_Population*(1/(Female_Male_Ratio+1));
Female_Population=Size_of_Population-Male_Population;
Min_Age=Parameters.Par_initiate_Population.Minimum_Init_Age;
Max_Age=handles.Parameters.Par_SimMenu.Age_of_Death+2;
Age_of_Death=Parameters.Par_SimMenu.Age_of_Death;
Choice=Parameters.Par_initiate_Population.FemaleChoice;
%If it is equal to 1 there is female choice if it is equal to 0 it is equal
%choice.

continueFlag=1;

MALE_ID_COUNTER=0;
FEMALE_ID_COUNTER=0;
COUNTER=0;
MARRIED_COUPLES_COUNTER=0;
SINGLE_DEAD_MEN_COUNTER=0;
SINGLE_DEAD_WOMEN_COUNTER=0;
MARRIED_DEAD_MEN_COUNTER=0;
MARRIED_DEAD_WOMEN_COUNTER=0;

Max_No_Individuals_Generated=1000;
One_Couple.men=0;
total_number_of_Married=0;
total_number_of_unMarried_Ind=Size_of_Population;
One_year=Parameters.Par_get_older.Duration_of_year;
%Preallocate
m_index=1;
rel_index=1;
All_Married_Couples=zeros(round(Max_No_Individuals_Generated/2),2);
MVsofPairs=zeros(size(All_Married_Couples));
RelMVsofPairs=zeros(size(All_Married_Couples));

Marrying_Age.males=zeros((Max_Age-Min_Age+1),1);
Marrying_Age.females=zeros((Max_Age-Min_Age+1),1);
Total_Males_Married_at_Age=zeros((Max_Age-Min_Age+1),1);
Total_Females_Married_at_Age=zeros((Max_Age-Min_Age+1),1);
All_Total_Married_at_Age=zeros((Max_Age-Min_Age+1),1);
Hazard_Rate.females=zeros((Max_Age-Min_Age+1),1);
Hazard_Rate.males=zeros((Max_Age-Min_Age+1),1);
Hazard_Rate.total=zeros((Max_Age-Min_Age+1),1);

%Initialize Random Seed
rand('state', Parameters.mate_choice_seed)

%Create the population with parameters Parameters_initiate_Population
[Population,Represented_Populations]=initiate_Population(Parameters.Par_initiate_Population,Para
meters.Par_get_older.Duration_of_year, Parameters.Par_self_esteem);

%create directory for records
%if there is a directory called temp-sim erase content, else create a new one
if (isdir('temp-sim'))
   delete(fullfile(pwd,filesep,'temp-sim',filesep,'*.*'));
else
   mkdir('temp-sim');
end

Married_Number_Iteration=zeros(round(Max_Age-Min_Age),1);
Marrying_Rate=zeros(round(Max_Age-Min_Age)+1,1);
Dates_Counter=zeros(Max_Iterations,1);
Matches_Counter=zeros(Max_Iterations,1);
marry_flag=0;
%consolidate workspace memory
```

```
cwd=pwd;
cd(tempdir);
pack;
cd(cwd);

while((FEMALE_ID_COUNTER+MALE_ID_COUNTER+1)<(Max_No_Individuals_Generated+2))
  %For each male, assign female contacts
  Contacts=assign_Contacts(Population.men, Population.women,
    Parameters.Par_assign_Contacts,Age_of_Death);

  %Put the assigned pairs into contact, characteristics are exchanged
  Represented_Populations=contact(Population, Represented_Populations, Contacts,
Parameters.Par_initiate_Population,Parameters.Par_exchange_Characteristics);

  %Female choice, else equal choice
  if(Choice==1)
    %For each man, consider asking out women for a date
    Matches=consider_Date(Population.men, Represented_Populations.mens,
        Parameters.Par_self_esteem, Population.women.maritial_Status,Choice);
    Matches_Counter(COUNTER+1)=size(Matches.men,1);
    %For each woman to be asked out by a man, ask out woman
    Dates=ask_Out(Population, Represented_Populations.womens, Matches,
        Parameters.Par_ask_Out,Parameters.Par_self_esteem,Choice);
  elseif(Choice==0)
    Matches=propose(Population, Represented_Populations, Parameters.Par_self_esteem,
      Parameters.Par_ask_Out);
    Dates=ask_Out_equal_choice(Population, Represented_Populations, Matches,
      Parameters.Par_ask_Out, Parameters.Par_self_esteem,Choice);
    Matches_Counter(COUNTER+1)=size(Matches.Matches_Men_Asking_Out.men,1)+size(Matches.Mat
      ches_Women_Asking_Out.men,1);
  end

  %For each pair, if the woman accepts, exchange characteristics
  %% Each date is considered to have a constant duration Date_Duration
  [Population,Represented_Populations]=date(Population, Represented_Populations, Dates,
    Parameters.Par_date, Parameters.Par_initiate_Population.Size_of_Characteristics,
    Parameters.Par_exchange_Characteristics,
    Parameters.Par_initiate_Population.Number_of_Last_Encounters);
  [Dates_Counter(COUNTER+1) dummy]=size(Dates.men);

  %Update the self-esteem values for men, and temperature values for each
  %individual
  Population=update_Self_Esteem(Population, Represented_Populations, Matches, Dates,
    Parameters.Par_initiate_Population.Male_Preference_Values,
    Parameters.Par_update_Self_Esteem, Parameters.Par_self_esteem,Choice);
  Population=update_Temperature(Population,Represented_Populations,  Dates,
    Parameters.Par_get_older.Duration_of_year, Parameters.Par_self_esteem);

  %the population gets older by one year every X iterations
  %Population gets older, individuals die and new generations are
  %generated
  %Check whether there is any pair that wants to marry
  [Population, Marrying_Age, Married_Couples]=want_to_Marry(Population, Dates,
    Marrying_Age,Parameters.Par_want_to_Marry);
  [Population Represented_Populations]=get_older(Population, Represented_Populations,
    Parameters.Par_get_older.Duration_of_year,
    Parameters.Par_initiate_Population,Parameters.Par_self_esteem, Age_of_Death);

  %Keep a record of the indices of married couples
  Number_Married = size(Married_Couples.men,1);
  if(ne(Married_Couples.men,0))
    All_Married_Couples((m_index):(m_index+Number_Married-
      1),1)=Population.men.ID(Married_Couples.men(:));
    All_Married_Couples((m_index):(m_index+Number_Married-
      1),2)=Population.women.ID(Married_Couples.women(:));
```

```
      MVsofPairs((m_index):(m_index+Number_Married-
        1),1)=Population.men.own_mate_value(Married_Couples.men(:));
      MVsofPairs((m_index):(m_index+Number_Married-
        1),2)=Population.women.own_mate_value(Married_Couples.women(:));
      for(k=1:Number_Married)
          RelMVsofPairs(rel_index,1)=Represented_Populations.womens(Married_Couples.women(k)).r
            epresented_mateValues(Married_Couples.men(k));
          RelMVsofPairs(rel_index,2)=Represented_Populations.mens(Married_Couples.men(k)).repres
            ented_mateValues(Married_Couples.women(k));
          rel_index=rel_index+1;
      end
      m_index=m_index+Number_Married;
   end
   %Calculate the percentage of married to unmarried in present iteration
   % if it's the end of year, calculate the marrying rate
   if (isempty(Married_Couples.men)|(Married_Couples.men==0))
      Married_Number_Iteration(COUNTER+1)=0;
   else
      [Married_Number_Iteration(COUNTER+1) dummy]=size(Married_Couples.men);
      %Assumes that male and female population is equal
      if(MARRIED_COUPLES_COUNTER==(Size_of_Population/2))
         continueFlag=0;
      end
      %Take a couple to plot
      if(marry_flag==0)
         marry_flag=1;
      end
   end
   if(Choice==1)
      record_data(Population, Matches, Dates, Married_Couples);
   else
      Matches_both.men=[Matches.Matches_Men_Asking_Out.men;
Matches.Matches_Women_Asking_Out.men];
      Matches_both.women=[Matches.Matches_Men_Asking_Out.women;
        Matches.Matches_Women_Asking_Out.women];
      record_data(Population, Matches_both, Dates, Married_Couples);
   end
   COUNTER=COUNTER+1;

   if(ismember(5,Population.men.ID))
      %open the temperature file and print the temperature values the men
      fid = fopen('temp-sim\temperature_oneMen.txt','at');
      fprintf(fid,'%5.2f\t',Population.men.temperature(find(Population.men.ID==5)));
      fprintf(fid,'\n');
      fclose(fid);
   end
   if(ismember(5,Population.women.ID))
      %open the temperature file and print the temperature values the men
      fid = fopen('temp-sim\temperature_oneWomen.txt','at');
      fprintf(fid,'%5.2f\t',Population.women.temperature(find(Population.women.ID==5)));
      fprintf(fid,'\n');
      fclose(fid);
   end
   if(ismember(5,Population.men.ID))
      %open the temperature file and print the temperature values the men
      fid = fopen('temp-sim\selfEsteem_oneMan.txt','at');
      fprintf(fid,'%5.2f\t',Population.men.self_esteem(find(Population.women.ID==5)));
      fprintf(fid,'\n');
      fclose(fid);
   end
end
%end of while loop

Nonzero_Indices=find(All_Married_Couples(:,1));
All_Married_Couples=All_Married_Couples(Nonzero_Indices,:);
MVsofPairs=MVsofPairs(Nonzero_Indices,:);
```

```
RelMVsofPairs=RelMVsofPairs(Nonzero_Indices,:);
%Find the MVs of couples
clear('continueFlag','COUNTER','total_number_of_unMarried_Ind','One_year','marry_flag');

%the number of unmarried people at each age
Total_Males_Married_at_Age(2:end)=cumsum(Marrying_Age.males(1:(end-1))); %from Minimum Age
to Max Age
Total_Females_Married_at_Age(2:end)=cumsum(Marrying_Age.females(1:(end-1)));
Total_Males_Married_at_Age(1)=0;
Total_Females_Married_at_Age(1)=0;
All_Total_Married_at_Age=Total_Males_Married_at_Age+Total_Females_Married_at_Age;

Unmarried_Males_at_Age=MALE_ID_COUNTER-Total_Males_Married_at_Age;
Unmarried_Females_at_Age=FEMALE_ID_COUNTER-Total_Females_Married_at_Age;
All_Unmarried_at_Age=(FEMALE_ID_COUNTER+MALE_ID_COUNTER)-
(Total_Females_Married_at_Age+Total_Males_Married_at_Age);

Marrying_Hazard_Rate.males=Marrying_Age.males./Unmarried_Males_at_Age;
Marrying_Hazard_Rate.females=Marrying_Age.females./Unmarried_Females_at_Age;
Marrying_Hazard_Rate.total=(Marrying_Age.males+Marrying_Age.females)./All_Unmarried_at_Age;

Duration_of_One_Run=Max_Iterations/Parameters.Par_get_older.Duration_of_year;
clear('Max_Iterations');

%record the seed and the Marrying Rate
fid = fopen('random_seed.txt', 'a');
fprintf(fid, '%d \n', Parameters.mate_choice_seed);
fclose(fid);

%Randomly choose one married Couple
if(not(isempty(Married_Couples.men)))
   Range=size(Married_Couples,1);
   Random_Index=ceil(Range*rand);
   Random_Index=truncate(Random_Index,1,Range,'both');
   One_Couple.man=Married_Couples.men(Random_Index);
   One_Couple.woman=Married_Couples.women(Random_Index);
else
   One_Couple.man=1;
   One_Couple.woman=1;
end
handles=plot_output(Married_Number_Iteration, Dates_Counter, Matches_Counter,handles);
handles=plot_one_ind(Population,One_Couple, handles);
handles=plot_rate(Marrying_Hazard_Rate, Min_Age, Max_Age, handles);
handles=plot_T(Population, Duration_of_One_Run,handles);
handles=plot_T_of_Married(Population,One_Couple, Duration_of_One_Run,handles);
Output.total_number_of_Married=(MARRIED_DEAD_MEN_COUNTER+MARRIED_DEAD_WOMEN_COUNTE
R);
%The percentage of married to the whole population is calculated (for the dead people)
Output.married_Percentage=round(((MARRIED_DEAD_MEN_COUNTER+MARRIED_DEAD_WOMEN_COUNT
ER)/(SINGLE_DEAD_MEN_COUNTER+SINGLE_DEAD_WOMEN_COUNTER+MARRIED_DEAD_MEN_COUNTER
+MARRIED_DEAD_WOMEN_COUNTER))*100);
[Corr, p]=corrcoef(MVsofPairs(:,1),MVsofPairs(:,2));
[RelCorr, Relp]=corrcoef(RelMVsofPairs(:,1),RelMVsofPairs(:,2));
Output.intracouple_correlation=Corr(1,2);
Output.rel_intracouple_correlation=RelCorr(1,2);
Output.handles=handles;

marrying_rate_stream = fopen('objective.corr.txt', 'a');
fprintf(marrying_rate_stream, '%3d \s', Output.intracouple_correlation);
fprintf(marrying_rate_stream, '%3d \s', p(1,2));
fprintf(marrying_rate_stream, '\n');
fclose(marrying_rate_stream);

marrying_rate_stream = fopen('subjective.corr.txt', 'a');
fprintf(marrying_rate_stream, '%3d \s', Output.rel_intracouple_correlation);
fprintf(marrying_rate_stream, '%3d \s', Relp(1,2));
```

```
fprintf(marrying_rate_stream, '\n');
fclose(marrying_rate_stream);

marrying_rate_stream = fopen('marrying_percentage.txt', 'a');
fprintf(marrying_rate_stream, '%3d \s', Output.married_Percentage);
fprintf(marrying_rate_stream, '\n');
fclose(marrying_rate_stream);

disp('One run completed.');

toc;

return
```

file initiate_Population.m

```
function [Population, Represented_Populations]=initiate_Population(Parameters,Duration_of_year,
   Par_self_esteem)
%INITIATE_POPULATION Create population
% POPULATION=INITIATE_POPULATION(SIZE_OF_POPULATION, FEMALE_MALE_RATIO) Creates a
population
% with a female male ratio of FEMALE_MALE_RATIO and size SIZE_OF_POPULATION
% FEMALE_MALE_RATIO is a real number between 0 and 1.
% SIZE_OF_POPULATION is an integer. POPULATION is a struct which consists
% of two HUMAN structs men and women
%
%INPUT PARAMETERS:
%   SIZE_OF_POPULATION: integer specifying the number of people in the population.
%   FEMALE_MALE_RATIO:  real number between 0 and 1 specifying the female to
%   male ratio.
%
%RETURN PARAMETERS:
%   POPULATION: population struct, consists of men and women SUBPOPULATION structs
%
%-----------------------------------------------------------------------

%generate the male population
%0/false is for male, and 1/true is for female
Male=false;
Female=true;
[Men, Mens_Represented_Population]=initiate_Subpopulation(Male,Parameters);
%generate the female population
[Women,Womens_Represented_Population]=initiate_Subpopulation(Female,Parameters);

Population.men=Men;
Population.women=Women;
Represented_Populations.mens=Mens_Represented_Population;
Represented_Populations.womens=Womens_Represented_Population;
%Initiate the Temperature values
Dates.men=0;
Dates.women=0;
Population=update_Temperature(Population,Represented_Populations, Dates,
   Duration_of_year,Par_self_esteem);
return
%end of initiate_Population
```

file initiate_Subpopulation.m

```
function [Subpopulation, Represented_Population]=initiate_Subpopulation(Sex,Parameters)
% Subpopulation =
%
%                     sex: [50x1 logical]
%         characteristics: [50x m double]
```

```
%                preferences: [50x m double]
%             own_mate_value: [50x 1 double]
%                temperature: [50x1 double]
%                        age: [50x1 double]
%                self_esteem: [50x1 double]
%            maritial_Status: [50x1 logical]
%                 experience: [50x1 double]
%            last_Encounters: [50xn double]
%                  last_Date: [50x2 double]
%    represented_Subpopulation:
global MALE_ID_COUNTER;
global FEMALE_ID_COUNTER;
Size_of_Population=Parameters.Size_of_Population;
Female_Male_Ratio=Parameters.Female_Male_Ratio;

Minimum_Init_Age=Parameters.Minimum_Init_Age;
Maximum_Init_Age=Parameters.Maximum_Init_Age;

Number_of_Characteristics=Parameters.Size_of_Characteristics;

%number of women in the population
Size_of_Female_Population= Size_of_Population*(Female_Male_Ratio/(Female_Male_Ratio+1));
%number of men in the population
Size_of_Male_Population=Size_of_Population*(1/(Female_Male_Ratio+1));

Preference_Weights=zeros(Number_of_Characteristics,2);
if(Sex==false) %if male
   Size_of_Subpopulation=round(Size_of_Male_Population);
   Size_of_Opposite_Sex_Population=round(Size_of_Female_Population);
   Preference_Weights=Parameters.Female_Preference_Values';
   Characteristics_Parameters=Parameters.Male_Characteristic_Values';
else %if female
   Size_of_Subpopulation=round(Size_of_Female_Population);
   Size_of_Opposite_Sex_Population=round(Size_of_Male_Population);
   Preference_Weights=Parameters.Male_Preference_Values';
   Characteristics_Parameters=Parameters.Female_Characteristic_Values';
end

%0 is for male and 1 is for female
Subpopulation.sex=repmat(Sex, Size_of_Subpopulation, 1);
Subpopulation.ID=zeros(Size_of_Subpopulation,1);
if(Sex==1)
   Counter=FEMALE_ID_COUNTER;
   FEMALE_ID_COUNTER=FEMALE_ID_COUNTER+Size_of_Female_Population;
else
   Counter=MALE_ID_COUNTER;
   MALE_ID_COUNTER=MALE_ID_COUNTER+Size_of_Male_Population;
end
for(i=1:Size_of_Subpopulation)
   Subpopulation.ID(i)=Counter+i;
end

Characteristics=zeros(Size_of_Subpopulation, Number_of_Characteristics);
Distr_Type=true; %0/false for uniform, 1/true for normal
%assign random values with mean expected_value and deviation variance
Char_Mean=Characteristics_Parameters(:,1);
Char_Variance=Characteristics_Parameters(:,2);
M_Size=Size_of_Subpopulation;
N_Size=Number_of_Characteristics;
%the assigned values should be of type type('integer' or 'real')
Char_Type=false; %0/false for real and 1/true for integer
Characteristics=random_Value_Generator(Distr_Type, Char_Mean, Char_Variance, M_Size, N_Size,
Char_Type);
Characteristics=truncate(Characteristics, 0,10,'both');
Subpopulation.characteristics=Characteristics;
```

```
Preferences=zeros(Size_of_Subpopulation, Number_of_Characteristics);
Pref_Mean= Preference_Weights(:,1);
Pref_Variance=Preference_Weights(:,2);
%the assigned values should be of type type('integer' or 'real'),the array size size,
M_Size=Size_of_Subpopulation;
N_Size=Number_of_Characteristics;
Pref_Type=false; %0/false for real, 1/integer for integer
Preferences= random_Value_Generator(Distr_Type, Pref_Mean, Pref_Variance, M_Size, N_Size,
Pref_Type);
Subpopulation.preferences=Preferences;

%Calculate the own mate values using the means of the opposite sex's preference weights
if(Number_of_Characteristics>1)
   Subpopulation.own_mate_value=calculate_MateValue(Characteristics,Preference_Weights(:,1)');
else
   Subpopulation.own_mate_value=Characteristics;
end

%This is actually some sort of a place holder.
Subpopulation.temperature= 5*ones(Size_of_Subpopulation,1);

%Age of the individuals, the age is chosen from a uniform distribution with
Lower_Bound=Minimum_Init_Age;
Upper_Bound=Maximum_Init_Age;
Difference=Upper_Bound-Lower_Bound;
Subpopulation.age=random_Value_Generator(0, Lower_Bound, Difference, M_Size, 1, 1 );
Subpopulation.initial_age=Subpopulation.age;
% first variable is Distr_Type=0(uniform) and the last variable is
% Type=1(real)
Upper_Bound=5;
Lower_Bound=9;
Self_Esteem_Mean=7;
Self_Esteem_Variance=0.5;
Subpopulation.self_esteem= random_Value_Generator(1, Self_Esteem_Mean, Self_Esteem_Variance,
M_Size, 1, 1 ); ;
% first variable is Distr_Type=1(normal) and the last variable is
% Type=1(real)
Subpopulation.self_esteem=truncate(Subpopulation.self_esteem, Lower_Bound, Upper_Bound,
'both');
%Each individual in the simulation starts single
% O is for single, 1 is for married
%Single is 0/false, Married is 1/true
Single=false;
Subpopulation.maritial_Status=repmat(Single, Size_of_Subpopulation, 1);

%The total number of dates that the individual had
Subpopulation.experience=zeros(Size_of_Subpopulation, 1);

%The first column is for the index of the last date and
%and the second for the number of dates with the last date
Subpopulation.last_Date=zeros(Size_of_Subpopulation, 2);

%Before people get into contact with each other they have default
%values for the representations of the others
%Each characteristics of the representation has the default value 4
%
% represented_Subpopulation=
%
%    struct array with fields:
%
%    represented_Characteristics
%    represented_mateValues
%    dates
%    encounter_Indicator
%    dates_Indicator
%    dates
```

```
%
Default_Characteristics_Value=Parameters.Default_Characteristics_Value;
Single_Represented_Subpopulation.represented_Characteristics=Default_Characteristics_Value*ones(
   Size_of_Opposite_Sex_Population,Number_of_Characteristics);
Single_Represented_Subpopulation.No_of_received_Characteristics=zeros(Size_of_Opposite_Sex_Pop
   ulation,Number_of_Characteristics);
%Assuming that the default values of the characteristics are equal
Single_Represented_Subpopulation.represented_mateValues=Default_Characteristics_Value*ones(Siz
   e_of_Opposite_Sex_Population,1);
Single_Represented_Subpopulation.encounter_Indicator=zeros(Size_of_Opposite_Sex_Population,1);
Single_Represented_Subpopulation.dates_Indicator=zeros(Size_of_Opposite_Sex_Population,1);
Single_Represented_Subpopulation.dates=zeros(Size_of_Opposite_Sex_Population,1);

%One subpopulation for each individual in the real population
Represented_Population=repmat(Single_Represented_Subpopulation, Size_of_Subpopulation,1);

return
%end of subpopulation function
```

file assign_Contacts.m

```
function Contacts=assign_Contacts(Men, Women, Parameters_assign_Contacts, Max_Age)
%Assign variables in structs to variables
Minimum_Number_of_Contacts=Parameters_assign_Contacts.Minimum_Number_of_Contacts;
Maximum_Number_of_Contacts=Parameters_assign_Contacts.Maximum_Number_of_Contacts;;
Difference_Number_of_Contacts = Maximum_Number_of_Contacts - Minimum_Number_of_Contacts;
Size_Men=size(Men.sex,1);
Size_Women=size(Women.sex,1);

%Find single men
Single_Men_Indices=find(Men.maritial_Status==0);
Size_Single_Men=size(Single_Men_Indices,1);

%Find single women
Single_Women_Indices=find(Women.maritial_Status==0);
Size_Single_Women=size(Single_Women_Indices,1);
%If all women or all men are married dont assign any contacts
if((Size_Single_Women==0)|(Size_Single_Men==0))
   Contacts.men_in_contact=0;
   Contacts.women_in_contact=0;
   Contacts.duration=0;
else
   Single_Men_Temperature=Men.temperature(Single_Men_Indices);
   Number_of_Contacts=zeros(Size_Single_Men, 1);
   %The number of contacts is assigned depending on the temperature,
   %The number of contacts change from Minimum_Number_of_Contacts to
Maximum_Number_of_Contacts
   %Upper Bound for temperature is
   Upper_Bound=2.5;
   Lower_Bound=1;
   Number_of_Contacts=ceil((Single_Men_Temperature)*(Minimum_Number_of_Contacts/
     Difference_Number_of_Contacts));
   Number_of_Contacts=round((((Single_Men_Temperature-Lower_Bound)./(Upper_Bound-
     Lower_Bound)).*Difference_Number_of_Contacts)+Minimum_Number_of_Contacts);
   %Replace the number of contacts that are higher than the max number of
   %contacts with the max num of contacts
   Exceeding_Indices=find(Number_of_Contacts>Maximum_Number_of_Contacts);
   Number_of_Contacts(Exceeding_Indices)=Maximum_Number_of_Contacts;

   %Replace the number of contacts that are lower than the min number of
   %contacts with the min num of contacts
   Exceeding_Indices2=find(Number_of_Contacts<Minimum_Number_of_Contacts);
   Number_of_Contacts(Exceeding_Indices2)=Minimum_Number_of_Contacts;
```

```
%Parameters for the assignment of women to men
Minimum_Woman_Index=Single_Women_Indices(1);
Maximum_Woman_Index=Single_Women_Indices(end);
Difference_Woman_Index=Maximum_Woman_Index-Minimum_Woman_Index;

Total_Contacts=sum(Number_of_Contacts);
%If there aren't any contacts
if(Total_Contacts==0)
   Men_in_Contact=0;
   Women_in_Contact=0;
   Duration=0;
else
   %Preallocation
   Men_in_Contact=zeros(Total_Contacts,1);
   Women_in_Contact=zeros(Total_Contacts,1);
   Duration=zeros(Total_Contacts,1);

   %Put indices of man into an array with respect to the number of contacts
   %they have..
   %Find the cumulative sum vector for the Numbers_of_Contacts vector
   Cum_Sum=cumsum(Number_of_Contacts);

   %Create the vector of indices
   Indices=[0; Cum_Sum];
   %Create the vector of Men_in_Contact by ordering the indices of
   %the men repeating as many as their contact number
   for i=1:Size_Single_Men
      if(ne(Indices(i),Indices(i+1)))
         Men_in_Contact((Indices(i)+1):Indices(i+1))=Single_Men_Indices(i);
      end
   end

   %assign women's indices according to the roulette-wheel selection
   %algorithm
   %The temperature is multiplied by 10 to avoid decimal numbers
   Single_Women_T_Proportion=round((Women.temperature(Single_Women_Indices).*10));
   Size_Wheel=sum(Single_Women_T_Proportion);

   %Generate the roulette-wheel
   Wheel=zeros(Size_Wheel,1);
   Cum_Sum=cumsum(Single_Women_T_Proportion);
   Indices=[0; Cum_Sum];
   %Assign a part of the wheel for a woman with respect to her temperature
   for i=1:Size_Single_Women
      if(ne(Indices(i),Indices(i+1)))
         Wheel((Indices(i)+1):Indices(i+1))=Single_Women_Indices(i);
      end
   end
   %Generate random values with the size of Size_Wheel
   Random_Values=ceil(Size_Wheel*rand(1,Total_Contacts));
   Women_in_Contact=Wheel(Random_Values);
   sort(Wheel(Random_Values));
   clear('Wheel','Random_Values');

   %Parameters for the random calculation of durations
   %The Contacts duration is equal to the characteristics exchanged by the pair
   Minimum_Contact_Duration=Parameters_assign_Contacts.Minimum_Contact_Duration;
   Maxiumum_Contact_Duration=Parameters_assign_Contacts.Maxiumum_Contact_Duration;
   Difference_Contact_Duration=Maxiumum_Contact_Duration-Minimum_Contact_Duration;
   clear('Parameters_assign_Contacts');
   % Assign duration of contacts randomly
   Duration=random_Value_Generator(0, Minimum_Contact_Duration,
       Difference_Contact_Duration, Total_Contacts, 1,1 );
   %the last argument is the number type integer=1
end
```

```
   Contacts.men_in_contact=Men_in_Contact;
   Contacts.women_in_contact=Women_in_Contact;
   Contacts.duration=Duration;
end
return
%end of assign_Contacts function
```

file contact.m

```
function Represented_Populations=contact(Population, Represented_Populations,Durations,
   Population_Parameters, Par_exchange_Characteristics)
Size_of_Characteristics=Population_Parameters.Size_of_Characteristics;
Number_of_Last_Encounters=Population_Parameters.Number_of_Last_Encounters;
Men_in_Contact=Durations.men_in_contact;
Women_in_Contact=Durations.women_in_contact;

if(Men_in_Contact==0)
else
   if (Size_of_Characteristics>1)
      [Represented_Populations.mens, Represented_Populations.womens]=exchange_Characteristics(
        Population, Represented_Populations.mens, Represented_Populations.womens,
        Durations,Size_of_Characteristics,Par_exchange_Characteristics);
   else
      [Represented_Populations.mens,
        Represented_Populations.womens]=exchange_Mate_Value(Population,
        Represented_Populations.mens, Represented_Populations.womens, Durations,
        Par_exchange_Characteristics);
   end
end
[Size_Men Dummy]=size(Population.men.sex);
[Size_Women Dummy]=size(Population.women.sex);
[Size_Durations Dummy]=size(Men_in_Contact);

%udpdate the encounter_indicators of the represented representations.
%For each represented female/male there is an indicator,
% the indicator has the value of a the number of encouners that the
% male/female can remember at the their encounter, and it decreases by one as a new contact step
has
% been done by the male/female. when the indicator is zero the male/female
% doesn't remember that represented female/male anymore

%first decrease 1 from each encounter_indicator
for i=1:Size_Men
   Represented_Populations.mens(i).encounter_Indicator=
     Represented_Populations.mens(i).encounter_Indicator-1;
   Represented_Populations.mens(i).encounter_Indicator(find(
     Represented_Populations.mens(i).encounter_Indicator<0))=0;
end
for j=1:Size_Women
   Represented_Populations.womens(j).encounter_Indicator=
     Represented_Populations.womens(j).encounter_Indicator-1;
   Represented_Populations.womens(j).encounter_Indicator(find(
     Represented_Populations.womens(j).encounter_Indicator<0))=0;
end

%refresh the encounter indicator for those who got in contact
if(Men_in_Contact==0)
else
   for k=1:Size_Durations
      Represented_Populations.mens(Men_in_Contact(k)).encounter_Indicator(
        Women_in_Contact(k))= Number_of_Last_Encounters;

        Represented_Populations.womens(Women_in_Contact(k)).encounter_Indicator(Men_in_Contac
        t(k)) =Number_of_Last_Encounters;
   end
end
```

file consider_date.m

```
function Matches=consider_Date(Men, Mens_Represented_Populations, Par_self_esteem,
Womens_Marital_Status,FemaleChoice)
Mens_Self_Esteem=Men.self_esteem;
Size_Men=size(Men.sex,1);
%Represented_size is the size of the female population
Represented_Size=size(Mens_Represented_Populations(1).represented_mateValues,1);
index=1;
Matches.women=zeros(Size_Men,1);
Matches.men=zeros(Size_Men,1);
Prob=zeros(Size_Men,1);
for i=1:Size_Men
   Total_Encounters=Mens_Represented_Populations(i).encounter_Indicator;
   Total_Encounters(find(Total_Encounters))=1;
   %check if the individual has married in the meanwhile
   Total_Encounters=xor(Total_Encounters, Womens_Marital_Status)&
     (Total_Encounters&(~Womens_Marital_Status));

   %Probability of being chosen of the woman by the man
   %P=((MVf)^(1/T))/Etha((MVf)^(1/T)))
   %if >1 then make =1
   %%Check divide by zero

   if(ne(Men.temperature(i),0))
     Divider=sum((Mens_Represented_Populations(i).represented_mateValues(
        find(Total_Encounters))). ^(1/Men.temperature(i)));
   else
      Divider=0;
      Prob=0;
   end
   if(ne(Divider,0))

        Prob=((((Mens_Represented_Populations(i).represented_mateValues(:)).^(1/Men.temperature(
        i)))) ./Divider);
%Choose only among those that have been recently encountered
      Prob=Prob.*Total_Encounters;
   else
      Prob=0;
   end
   if (Prob>1) Prob=1; end;
   if(ne(sum(Prob),0))
      Random_Value=rand(1);
      woman_index=1;
      Woman_Indices=find(Prob);
      Prob=Prob(Woman_Indices);
      Total=Prob(woman_index);
      while(Random_Value>Total)
         woman_index=woman_index+1;
         Total=Total+Prob(woman_index);
      end
      if(Par_self_esteem.mode_of_self_esteem_integration==1)
         Random_Value=rand(1);
         Dare=Random_Value<(Mens_Self_Esteem(i)/10);
      else
         Dare=1;
      end
      if (Dare==1)
         Matches.women(index)=Woman_Indices(woman_index);
         Matches.men(index)=i;
         index=index+1;
      end
   end
end
%remove the zero values
```

```
W_Indices=find(Matches.women);
Matches.women=Matches.women(W_Indices);
Matches.men=Matches.men(W_Indices);
%If there are no matches, then Matches.men and Matches.woman are 0
if (isempty(Matches.men))
   Matches.men=0;
   Matches.women=0;
end
return
%end of function consider_Date
```

file ask_Out.m

```
function Pairs=ask_Out(Population,Womens_Represented_Populations, Matches,
   Parameters,Par_self_esteem, FemaleChoice)
Number_compared=Parameters.number_of_last_dates_compared;
Female_Temperature_Constant=Par_self_esteem.female_temperature_constant;
Size_of_Matches=size(Matches.men,1);
Pairs.men=zeros(Size_of_Matches,1);
Pairs.women=zeros(Size_of_Matches,1);
Size_Women=size(Population.women.sex,1);
Size_Men=size(Population.men.sex,1);
Women=Population.women;
Men=Population.men;
index=1;

if(ne(Matches.women,0))
   for(i=1:Size_Women)
      if(ismember(i, Matches.women))
         Total_Dates=Womens_Represented_Populations(i).dates_Indicator;
         if(eq(sum(Total_Dates),0))
            Total_Dates=Womens_Represented_Populations(i).encounter_Indicator;
         end
         [Duration Date_Index]=sort(Total_Dates);
         %revert
         Size_Dates=size(Total_Dates,1);
         for m=1:Size_Dates
            Duration_Rev((Size_Dates+1)-m)=Duration(m);
            Date_Index_Rev((Size_Dates+1)-m)=Date_Index(m);
         end
         %Remove 0s
         Date_Index_Rev=Date_Index_Rev(find(Duration_Rev));
         if (size(Date_Index_Rev,2)>=Number_compared)
            Date_Indices=Date_Index_Rev(1:Number_compared);
         else
            Date_Indices=Date_Index_Rev(:);
         end
         %Calculate mean and variance for the recently dated/met people
         Mean=mean(Womens_Represented_Populations(i).represented_mateValues(Date_Indices));
         Std=std(Womens_Represented_Populations(i).represented_mateValues(Date_Indices));
         Proposing_Men=Matches.men(find(Matches.women==i));
         [Males_Mate_Value
            Males_Index]=max(Womens_Represented_Populations(i).represented_mateValues(
            Proposing_Men));

            Males_New_Mate_Value=Males_Mate_Value+Female_Temperature_Constant*log(Women.t
            emperature(i) +1);
         P = normcdf(Males_New_Mate_Value,Mean,Std);
         if(isnan(P))
            P=1;
         end
         Random_Value=rand;
         Accept=Random_Value<P ;
         if(Accept)
            Pairs.women(index)=i;
            Pairs.men(index)=Proposing_Men(Males_Index);
```

```
            index=index+1;
        end
    end
  end
end
%Remove the zeros in the arrays
W_Indices=find(Pairs.women);
Pairs.women=Pairs.women(W_Indices);
Pairs.men=Pairs.men(W_Indices);

%If there are no pairs formed assign 0 to Pairs.men and Pairs.women
if(isempty(Pairs.women))
  Pairs.men=0;
  Pairs.women=0;
end
return
%end of function ask_Out
```

file propose.m

```
function Matches=propose(Population, Represented_Populations, Par_self_esteem,Par_ask_Out)
Men=Population.men;
Women=Population.women;
Mens_Represented_Populations=Represented_Populations.mens;
Womens_Represented_Populations=Represented_Populations.womens;
Mens_Self_Esteem=Men.self_esteem;
Womens_Self_Esteem=Women.self_esteem;
Womens_Marital_Status=Women.maritial_Status;

Size_Men=size(Men.sex,1);
Size_Women=size(Women.sex,1);

index=1;
Matches_Men_Asking_Out.women=zeros(Size_Men,1);
Matches_Men_Asking_Out.men=zeros(Size_Men,1);
Prob=zeros(Size_Men,1);
 for i=1:Size_Men
    Total_Encounters=Mens_Represented_Populations(i).encounter_Indicator;
    Total_Encounters(find(Total_Encounters))=1;
    %Check the woman whether they married in the meanwhile. Remove if they
    %did
    Total_Encounters=xor(Total_Encounters, Womens_Marital_Status)& (Total_Encounters&
        (~Womens_Marital_Status));

    %Probability of being chosen of the woman by the man
    %P=((MVf)^(1/T))/Etha((MVf)^(1/T)))
    %%Check divide by zero
    if(ne(Men.temperature(i),0))

        Divider=sum((Mens_Represented_Populations(i).represented_mateValues(find(Total_Encoun
        ters))).^ (1/Men.temperature(i)));
    else
       Divider=0;
       Prob=0;
    end
    if(ne(Divider,0))

        Prob=((((Mens_Represented_Populations(i).represented_mateValues(:)).^(1/Men.temperatu
        re(i)))) ./Divider);
       %Choose only among those that have been recently encountered
       Prob=Prob.*Total_Encounters;
    else
       Prob=0;
    end
    %if >1 then make =1
    if (Prob>1) Prob=1; end;
```

```
    if(ne(sum(Prob),0))
       Random_Value=rand(1);
       woman_index=1;
       Woman_Indices=find(Prob);
       Prob=Prob(Woman_Indices);
       Total=Prob(woman_index);
       while(Random_Value>Total)
          woman_index=woman_index+1;
          Total=Total+Prob(woman_index);
       end
       if(Par_self_esteem.mode_of_self_esteem_integration==1)
          Random_Value=rand(1);
          Dare=Random_Value<(Mens_Self_Esteem(i)/10);
       else
          Dare=1;
       end
       if (Dare==1)
          Matches_Men_Asking_Out.women(index)=Woman_Indices(woman_index);
          Matches_Men_Asking_Out.men(index)=i;
          index=index+1;
       end
    end
 end
Matches_Women_Asking_Out.women=zeros(Size_Men,1);
Matches_Women_Asking_Out.men=zeros(Size_Men,1);
index=1;
Prob=zeros(Size_Men,1);
for i=1:Size_Women
    Total_Encounters=Womens_Represented_Populations(i).encounter_Indicator;
    Total_Encounters(find(Total_Encounters))=1;
    %Check if the man has married in the meanwhile, remove if he did
    Total_Encounters=xor(Total_Encounters, Womens_Marital_Status)&(Total_Encounters&
        (~Womens_Marital_Status));

    %Probability of being chosen of the man by the woman
    %P=((MVf)^(1/T))/Etha((MVf)^(1/T)))
    %%Check divide by zero
    if(ne(Wosmen.temperature(i),0))

         Divider=sum((Womens_Represented_Populations(i).represented_mateValues(find(Total_Enc
         ounters))) .^(1/Women.temperature(i)));
    else
       Divider=0;
       Prob=0;
    end
    if(ne(Divider,0))

         Prob=((((Womens_Represented_Populations(i).represented_mateValues(:)).^(1/Women.tem
         perature(i)))) ./Divider);
       %Choose only among those that have been recently encountered
       Prob=Prob.*Total_Encounters;
    else
       Prob=0;
    end
    %if >1 then make =1
    if (Prob>1) Prob=1; end;
    if(ne(sum(Prob),0))
       Random_Value=rand(1);
       man_index=1;
       Man_Indices=find(Prob);
       Prob=Prob(Man_Indices);
       Total=Prob(man_index);
       while(Random_Value>Total)
          man_index=man_index+1;
          Total=Total+Prob(man_index);
       end
```

```
        if(Par_self_esteem.mode_of_self_esteem_integration==1)
          Random_Value=rand(1);
          Dare=Random_Value<(Womens_Self_Esteem(i)/10);
        else
          Dare=1;
        end
        if (Dare==1)
          Matches_Women_Asking_Out.women(index)=i;
          Matches_Women_Asking_Out.men(index)=Man_Indices(man_index);
          index=index+1;
        end
    end
 end
W_Indices=find(Matches_Men_Asking_Out.women);
Matches_Men_Asking_Out.women=Matches_Men_Asking_Out.women(W_Indices);
Matches_Men_Asking_Out.men=Matches_Men_Asking_Out.men(W_Indices);
W_Indices=find(Matches_Women_Asking_Out.women);
Matches_Women_Asking_Out.women=Matches_Women_Asking_Out.women(W_Indices);
Matches_Women_Asking_Out.men=Matches_Women_Asking_Out.men(W_Indices);
%If there are no matches, then Matches.men and Matches.woman are 0
if (isempty(Matches_Men_Asking_Out.men))
   Matches_Men_Asking_Out.men=0;
   Matches_Men_Asking_Out.women=0;
end
if (isempty(Matches_Women_Asking_Out.men))
   Matches_Women_Asking_Out.men=0;
   Matches_Women_Asking_Out.women=0;
end
Matches.Matches_Men_Asking_Out=Matches_Men_Asking_Out;
Matches.Matches_Women_Asking_Out=Matches_Women_Asking_Out;
return
%end of function propose
```

file ask_Out_equal_choice.m

```
function Pairs=ask_Out_equal_choice(Population,Represented_Populations, Matches, Parameters,
   Par_self_esteem, FemaleChoice)
Number_compared=Parameters.number_of_last_dates_compared;
Female_Temperature_Constant=Par_self_esteem.female_temperature_constant;
if(FemaleChoice==1)
   Size_of_Matches=size(Matches.men,1);
   Mens_Matches=Matches;
   Pairs.men=zeros(Size_of_Mens_Matches,1);
   Pairs.women=zeros(Size_of_Mens_Matches,1);
end
Size_Women=size(Population.women.sex,1);
Size_Men=size(Population.men.sex,1);
Women=Population.women;
Men=Population.men;
index=1;

if(FemaleChoice==0)
   Mens_Matches=Matches.Matches_Men_Asking_Out;
   Womens_Matches=Matches.Matches_Women_Asking_Out;
   Size_of_Mens_Matches=size(Mens_Matches.men,1);
   Size_of_Womens_Matches=size(Womens_Matches.men,1);
   Initial_Mens_Pairs.men=zeros(Size_of_Mens_Matches,1);
   Initial_Mens_Pairs.women=zeros(Size_of_Mens_Matches,1);
   Initial_Womens_Pairs.men=zeros(Size_of_Womens_Matches,1);
   Initial_Womens_Pairs.women=zeros(Size_of_Womens_Matches,1);
   Pairs.men=0;
   Pairs.women=0;
end
Mens_Represented_Populations=Represented_Populations.mens;
Womens_Represented_Populations=Represented_Populations.womens;
```

```
if(ne(Mens_Matches.men(1),0))
   for(i=1:Size_Women)
      if(ismember(i, Mens_Matches.women))
         Total_Dates=Womens_Represented_Populations(i).dates_Indicator;
         if(eq(sum(Total_Dates),0))
            Total_Dates=Womens_Represented_Populations(i).encounter_Indicator;
         end
         [Duration Date_Index]=sort(Total_Dates);
         %revert
         Size_Dates=size(Total_Dates,1);
         for m=1:Size_Dates
            Duration_Rev((Size_Dates+1)-m)=Duration(m);
            Date_Index_Rev((Size_Dates+1)-m)=Date_Index(m);
         end
         %Remove 0s
         Date_Index_Rev=Date_Index_Rev(find(Duration_Rev));
         if (size(Date_Index_Rev,2)>=Number_compared)
            Date_Indices=Date_Index_Rev(1:Number_compared);
         else
            Date_Indices=Date_Index_Rev(:);
         end
         %Calculate mean and variance for the recently dated/met people
         Mean=mean(Womens_Represented_Populations(i).represented_mateValues(Date_Indices));
         Std=std(Womens_Represented_Populations(i).represented_mateValues(Date_Indices));
         Proposing_Men=Mens_Matches.men(find(Mens_Matches.women==i));
         [Males_Mate_Value
            Males_Index]=max(Womens_Represented_Populations(i).represented_mateValues(
            Proposing_Men));

            Males_New_Mate_Value=Males_Mate_Value+Female_Temperature_Constant*log(Women.t
            emperature(i) +1);
         P = normcdf(Males_New_Mate_Value,Mean,Std);
         if(isnan(P))
            P=1;
         end
         Random_Value=rand;
         Accept=Random_Value<P ;
         if(Accept)
            Initial_Mens_Pairs.women(index)=i;
            Initial_Mens_Pairs.men(index)=Proposing_Men(Males_Index);
            index=index+1;
         end
      end
   end
end
%Remove the zeros in the arrays
W_Indices=find(Initial_Mens_Pairs.women);
Initial_Mens_Pairs.women=Initial_Mens_Pairs.women(W_Indices);
Initial_Mens_Pairs.men=Initial_Mens_Pairs.men(W_Indices);

if(ne(Womens_Matches.men(1),0))
   for(i=1:Size_Men)
      if(ismember(i, Womens_Matches.men))
         Total_Dates=Mens_Represented_Populations(i).dates_Indicator;
         if(eq(sum(Total_Dates),0))
            Total_Dates=Mens_Represented_Populations(i).encounter_Indicator;
         end
         [Duration Date_Index]=sort(Total_Dates);
         %revert
         Size_Dates=size(Total_Dates,1);
         for m=1:Size_Dates
            Duration_Rev((Size_Dates+1)-m)=Duration(m);
            Date_Index_Rev((Size_Dates+1)-m)=Date_Index(m);
         end
         %Remove 0s
         Date_Index_Rev=Date_Index_Rev(find(Duration_Rev));
```

```
      if (size(Date_Index_Rev,2)>=Number_compared)
        Date_Indices=Date_Index_Rev(1:Number_compared);
      else
        Date_Indices=Date_Index_Rev(:);
      end
      %Calculate mean and variance for the recently dated/met people
      Mean=mean(Mens_Represented_Populations(i).represented_mateValues(Date_Indices));
      Std=std(Mens_Represented_Populations(i).represented_mateValues(Date_Indices));
      Proposing_Women=Womens_Matches.women(find(Womens_Matches.men==i));
      [Females_Mate_Value
         Males_Index]=max(Womens_Represented_Populations(i).represented_mateValues
         (Proposing_Women));
      Females_New_Mate_Value=Females_Mate_Value+Female_Temperature_Constant*log(
         sMen.temperature(i)+1);
      P = normcdf(Females_New_Mate_Value,Mean,Std);
      if(isnan(P))
        P=1;
      end
      Random_Value=rand;
      Accept=Random_Value<P ;
      if(Accept)
        Initial_Womens_Pairs.women(index)=i;
        Initial_Womens_Pairs.men(index)=Proposing_Women(Males_Index);
        index=index+1;
      end
    end
  end
end
%Remove the zeros in the arrays
W_Indices=find(Initial_Womens_Pairs.women);
Initial_Womens_Pairs.women=Initial_Womens_Pairs.women(W_Indices);
Initial_Womens_Pairs.men=Initial_Womens_Pairs.men(W_Indices);

Size_Initial_Mens=size(Initial_Mens_Pairs.men,1);

index=1;

%First remove the people asking out and their partner
%Men asking out
for(i=1:Size_Initial_Mens)
  %Check whether the man has been asked out by a female
  Asked_Out=find(Initial_Womens_Pairs.men==Initial_Mens_Pairs.men(i));
  if (isempty(Asked_Out)&(ne(Initial_Mens_Pairs.men(i),0)))
    Pairs.men(index)=Initial_Mens_Pairs.men(i);
    Pairs.women(index)=Initial_Mens_Pairs.women(i);
    index=index+1;
  elseif(ne(Initial_Mens_Pairs.men(i),0))
    Chosen=round(rand*(size(Asked_Out,1)));
    if(Chosen==0)
      Pairs.men(index)=Initial_Mens_Pairs.men(i);
      Pairs.women(index)=Initial_Mens_Pairs.women(i);
      index=index+1;
      Initial_Womens_Pairs.men(Asked_Out)=0;
    else
      Pairs.men(index)=Initial_Womens_Pairs.men(Asked_Out(Chosen));
      Pairs.women(index)=Initial_Womens_Pairs.women(Asked_Out(Chosen));
      Asked_Out=Asked_Out(find(Asked_Out~=Chosen));
      Initial_Womens_Pairs.men(Asked_Out)=0;
       index=index+1;
    end
  end
  Remove_Other_Men=find(Initial_Mens_Pairs.women==Initial_Mens_Pairs.women(i));
  Initial_Mens_Pairs.men(Remove_Other_Men)=0;
  Initial_Mens_Pairs.men(i)=0;
end
W_Indices=find(Initial_Womens_Pairs.men);
```

```
Initial_Womens_Pairs.men=Initial_Womens_Pairs.men(W_Indices);
Initial_Womens_Pairs.women=Initial_Womens_Pairs.women(W_Indices);
Size_Initial_Womens=size(Initial_Womens_Pairs.men,1);
for(i=1:Size_Initial_Womens)
   if(isempty(find(Pairs.women==Initial_Womens_Pairs.women(i))))
   Asked_Out=find(Initial_Womens_Pairs.women==Initial_Womens_Pairs.women(i));
   if ((size(Asked_Out,1)==1)&(ne(Asked_Out(1),0)))
      Pairs.men(index)=Initial_Womens_Pairs.men(i);
      Pairs.women(index)=Initial_Womens_Pairs.women(i);
      index=index+1;
   elseif((ne(Asked_Out(1),0)))
      Chosen=round(rand*(size(Asked_Out,1)));
      Pairs.men(index)=Initial_Womens_Pairs.men(Asked_Out(Chosen));
      Pairs.women(index)=Initial_Womens_Pairs.women(Asked_Out(Chosen));
      Asked_Out=Asked_Out(find(Asked_Out~=Chosen));
      Initial_Womens_Pairs.women(Asked_Out)=0;
      index=index+1;
   end
end
end

W_Indices=find(Pairs.women);
Pairs.women=Pairs.women(W_Indices);
Pairs.men=Pairs.men(W_Indices);

%If there are no pairs formed assign 0 to Pairs.men and Pairs.women
if(isempty(Pairs.women))
   Pairs.men=0;
   Pairs.women=0;
end
Pairs.men=Pairs.men';
Pairs.women=Pairs.women';
return
%end of function ask_Out_equal_choice
```

file date.m

```
function [Population, Represented_Populations]=date(Population,
Represented_Populations,Pairs_date, Parameters_date,
Size_of_Characteristics,Par_exchange_Characteristics,Number_of_Last_Encounters)
if (any(Pairs_date.women))
   Date_Duration=Parameters_date.Date_Duration;
   Number_of_Last_Dates=Parameters_date.Last_Remembered;
   Durations.men_in_contact=Pairs_date.men;
   Durations.women_in_contact=Pairs_date.women;
   Pairs_date_Size=size(Pairs_date.men, 1);
   Durations.duration=Date_Duration*ones(Pairs_date_Size,1);
   Size_Men=size(Population.men.sex,1);
   Size_Women=size(Population.women.sex,1);
   if (Size_of_Characteristics>1)
      [Represented_Populations.mens, Represented_Populations.womens]=exchange_Characteristics(
        Population, Represented_Populations.mens, Represented_Populations.womens, Durations,
        Size_of_Characteristics,Par_exchange_Characteristics);
   else
      [Represented_Populations.mens,
        Represented_Populations.womens]=exchange_Mate_Value(Population,
        Represented_Populations.mens, Represented_Populations.womens, Durations,
        Par_exchange_Characteristics);
   end

   [Population, Represented_Populations]=update_Experience(Population, Represented_Populations,
     Durations);
   %first decrease 1 from each dates_indicator
   %then change the values that are negative to 0
```

```
    for i=1:Size_Men

Represented_Populations.mens(i).dates_Indicator=Represented_Populations.mens(i).dates_Indicator-
1;
        Represented_Populations.mens(i).dates_Indicator(find(
          Represented_Populations.mens(i).dates_Indicator<0))=0;
    end
    for j=1:Size_Women

          Represented_Populations.womens(j).dates_Indicator=Represented_Populations.womens(j).dat
          es_Indicator -1;
        Represented_Populations.womens(j).dates_Indicator(find(
          Represented_Populations.womens(j).dates_Indicator<0))=0;
    end
    % %refresh the encounter indicator for those who dated
    if(any(Pairs_date.women))
        for k=1:Pairs_date_Size
            Represented_Populations.mens(Durations.men_in_contact(k)).dates_Indicator(
              Durations.women_in_contact(k))=Number_of_Last_Dates;
            Represented_Populations.womens(Durations.women_in_contact(k)).dates_Indicator(
              Durations.men_in_contact(k))=Number_of_Last_Dates;
            Represented_Populations.mens(Durations.men_in_contact(k)).encounter_Indicator(
              Durations.women_in_contact(k))=Number_of_Last_Encounters+1;
            Represented_Populations.womens(Durations.women_in_contact(k)).encounter_Indicator(
              Durations.men_in_contact(k))=Number_of_Last_Encounters+1;
        end
     end
end
return
```

file exchange_characteristics.m

```
function [Mens_Representations, Womens_Representations]=exchange_Characteristics(Population,
   Mens_Representations, Womens_Representations, Contacts, Size_of_Char, Parameter)
%This function sends and receives characteristics of the individual in contact or
%dating
Mens_Characteristics=Population.men.characteristics;
Womens_Characteristics=Population.women.characteristics;
Mens_Preferences=Population.men.preferences;
Womens_Preferences=Population.women.preferences;

Men_in_Contact=Contacts.men_in_contact;
Women_in_Contact=Contacts.women_in_contact;
Duration=Contacts.duration;

%Number of contacts in the present iteration
[Size_of_Contacts, Dummy]=size(Men_in_Contact);
Max_in_Random=Size_of_Char-1;

for i=1:Size_of_Contacts
    Female_Index=Women_in_Contact(i);
    Male_Index=Men_in_Contact(i);

    %randomly pick characteristics
    %the number of characteristics is equal to the duration of the contact
    if(Duration(i)>1)
        Picked_Female_Characteristics_Indices=random_Value_Generator(0, 1, (Max_in_Random-1),
          Duration(i),1, 1);
        Picked_Female_Characteristics_Indices(2:end+1)=Picked_Female_Characteristics_Indices;
        %First argument is uniform=0, last argument is integer 1
        Picked_Male_Characteristics_Indices=random_Value_Generator(0, 1, Max_in_Random-1,
Duration(i),1, 1);
        Picked_Male_Characteristics_Indices(2:end+1)=Picked_Male_Characteristics_Indices;
    end
    Picked_Female_Characteristics_Indices(1)=4;
    Picked_Male_Characteristics_Indices(1)=10;
```

```
%Put the picked characteristics in an array
Picked_Female_Characteristics=Womens_Characteristics(Female_Index,
  Picked_Female_Characteristics_Indices)';
Picked_Male_Characteristics=Mens_Characteristics(Male_Index,
Picked_Male_Characteristics_Indices)';
%Generate the semi-random characteristics
%First argument normal distribution=1, last argument number type real=0
%The values are distorted by a std dev of 0.1.

Distortion_Deviation=Parameter.Distortion_St_Deviation*ones(size(Picked_Female_Characteristics));
Received_Female_Characteristics=random_Value_Generator(1, Picked_Female_Characteristics,
  Distortion_Deviation, 1, Duration(i), 0);
Received_Male_Characteristics=random_Value_Generator(1, Picked_Male_Characteristics,
  Distortion_Deviation, 1,Duration(i),0);

%update the char value for the man's female representation
Female_Char_Received_Before=zeros(1,Size_of_Char);
%Find the characteristics values is received before
% Female_Char_Received_Before is 1 if it was received before,
%stays 0 if it was not

  Female_Char_Received_Before(find(Mens_Representations(Female_Index).No_of_received_Char
  acteristics( Female_Index,:)))=1;
Represented_Female_Char=Mens_Representations(Male_Index).represented_Characteristics(
  Female_Index,:);
for(j=1: Size_of_Char)
  %How many times was it received in this step
  Female_Char_Times_Received=size(find(Picked_Female_Characteristics_Indices==j),1);
  %If this characteristics was received at least one time,
  %Increase the number of times received counter
  %And include it to the previously received characteristics by
  %averaging
  if(ne(Female_Char_Times_Received,0))

      Represented_Female_Char(j)=Represented_Female_Char(j)*Female_Char_Received_Before(
      j)+
      (sum(Received_Female_Characteristics(find(Picked_Female_Characteristics_Indices==j)))./
      Female_Char_Times_Received);

Female_Char_Received_Before(j)=Female_Char_Received_Before(j)+Female_Char_Times_Received;
  end
end
Mens_Representations(Male_Index).No_of_received_Characteristics(Female_Index,:)=
  Female_Char_Received_Before;
Mens_Representations(Male_Index).represented_Characteristics(Female_Index,:)=
  Represented_Female_Char;



%update the char value for the woman's male representation
Male_Char_Received_Before=zeros(size(Womens_Representations(Female_Index
  ).No_of_received_Characteristics( Male_Index,:)));

  Male_Char_Received_Before(find(Womens_Representations(Female_Index).No_of_received_Char
  acteristics( Male_Index,:)))=1;
Represented_Male_Char=Womens_Representations(Female_Index).represented_Characteristics(
  Male_Index,:);
for(j=1: Size_of_Char)
  Male_Char_Times_Received=size(find(Picked_Male_Characteristics_Indices==j),1);
  if(ne(Male_Char_Times_Received,0))
     Represented_Male_Char(j)=Represented_Male_Char(j)*Male_Char_Received_Before(j)+
        (sum(Received_Male_Characteristics(find(Picked_Male_Characteristics_Indices==j)))./
        Male_Char_Times_Received);
     Male_Char_Received_Before(j)=Male_Char_Received_Before(j)+Male_Char_Times_Received;
  end
```

```
   end
   Womens_Representations(Female_Index).No_of_received_Characteristics(Male_Index,:)=
     Male_Char_Received_Before;
   Womens_Representations(Female_Index).represented_Characteristics(Male_Index,:)=
     Represented_Male_Char;

   %Calculate the Mate values for each

     Womens_Representations(Female_Index).represented_mateValues(Male_Index)=calculate_MateV
     alue( Womens_Representations(Female_Index).represented_Characteristics(Male_Index,:),
     Womens_Preferences(Female_Index,:));
   Mens_Representations(Male_Index).represented_mateValues(Female_Index)=calculate_MateValue(
     Mens_Representations(Male_Index).represented_Characteristics(Female_Index,:),
     Mens_Preferences(Male_Index,:));
end
return
%end of exchange_Characteristics function
```

file calculate_mateValue.m

```
function   MateValue=calculate_MateValue(Characteristics, Preferences)
%The characteristics argument is in the form of a column-wise array
%The preferences array is in the form of a line-wise array

%The mean of values for each characteristics
%the code below is for leaving the 0 values in the Characteristics Matrixs
%out while taking the average
[No_of_Pref Dummy ]=size(Preferences);
[No_of_Char Dummy ]=size(Characteristics);
if((No_of_Pref==1)&(No_of_Char~=No_of_Pref))
   Preference_Matrix=repmat(Preferences, No_of_Char,1);
   MateValue=sum(Characteristics.*Preference_Matrix,2)./sum(Preference_Matrix,2);
else
   MateValue=sum(Characteristics.*Preferences)./sum(Preferences);
end
MateValue=truncate(MateValue,0,10,'both');
return
 %end of calculate_MateValue
```

file exchange_mateValue.m

```
function [Mens_Representations, Womens_Representations]=exchange_Characteristics(Population,
   Mens_Representations, Womens_Representations, Contacts,  Parameter)
%This function sends and receives characteristics of the individual in contact or
%dating
Mens_Mate_Values=Population.men.characteristics;
Womens_Mate_Values=Population.women.characteristics;

Men_in_Contact=Contacts.men_in_contact;
Women_in_Contact=Contacts.women_in_contact;
Duration=Contacts.duration;

%Number of contacts in the present iteration
[Size_of_Contacts, Dummy]=size(Men_in_Contact);

for i=1:Size_of_Contacts
   Female_Index=Women_in_Contact(i);
   Male_Index=Men_in_Contact(i);
   %Retrieve the actual mate values
   Females_Actual_Mate_Value=Womens_Mate_Values(Female_Index,1);
   Males_Actual_Mate_Value=Mens_Mate_Values(Male_Index, 1);

   %Generate the semi-random mate values
   %First argument normal distribution=1, last argument number type real=0
   %The values are distorted by a std dev of Distortion_St_Deviation.
   Distortion_Deviation=Parameter.Distortion_St_Deviation;
```

```
Received_Female_Characteristics=random_Value_Generator(1,
  Females_Actual_Mate_Value,Distortion_Deviation, 1 , Duration(i), 0);
Received_Male_Characteristics=random_Value_Generator(1, Males_Actual_Mate_Value,
  Distortion_Deviation, 1,Duration(i),0);

Mens_Representations(Male_Index).No_of_received_Characteristics(Female_Index,:)=
Female_Char_Received_Before;
Mens_Representations(Male_Index).represented_Characteristics(Female_Index,:)=
  Represented_Female_Char;
%update the mate value for the man's female representation
Female_MV_Received_Before=0;
%if the MV was received before then Female_MV_Received_Before is 1,
%otherwise it remains the same

  Female_MV_Received_Before(find(Mens_Representations(Female_Index).No_of_received_Charac
  teristics( Female_Index,1)))=1;

Represented_Female_MV=Mens_Representations(Male_Index).represented_Characteristics(Female_In
dex,1);

%How many times was it received in this step
Female_MV_Times_Received=Duration(i);
%Increase the number of times received counter
%And include it to the previously received characteristics by
%averaging
Represented_Female_MV=Represented_Female_MV*Female_MV_Received_Before+(sum(
  Received_Female_Characteristics)./Female_MV_Times_Received);
Female_MV_Received_Before=Female_MV_Received_Before+Female_MV_Times_Received;
Mens_Representations(Male_Index).No_of_received_Characteristics(Female_Index)=
  Female_MV_Received_Before;

Mens_Representations(Male_Index).represented_Characteristics(Female_Index)=Represented_Female
_MV;

%update the char value for the woman's male representation
Male_MV_Received_Before=0;
%if the MV was received before then Male_MV_Received_Before is 1,
%otherwise it remains the same

  Male_MV_Received_Before(find(Womens_Representations(Female_Index).No_of_received_Chara
  cteristics( Male_Index,1)))=1;

Represented_Male_MV=Womens_Representations(Female_Index).represented_Characteristics(Male_I
ndex,1);

%How many times was it received in this step
Male_MV_Times_Received=Duration(i);
%Increase the number of times received counter
%And include it to the previously received characteristics by
%averaging
Represented_Male_MV=Represented_Male_MV*Male_MV_Received_Before+(sum(
  Received_Male_Characteristics)./Male_MV_Times_Received);
Male_MV_Received_Before=Male_MV_Received_Before+Male_MV_Times_Received;

Womens_Representations(Female_Index).No_of_received_Characteristics(Male_Index)=
  Male_MV_Received_Before;

Womens_Representations(Female_Index).represented_Characteristics(Male_Index)=Represented_Mal
e_MV;
Womens_Representations(Female_Index).represented_mateValues(Male_Index)=
  Womens_Representations( Female_Index).represented_Characteristics(Male_Index);

  Mens_Representations(Male_Index).represented_mateValues(Female_Index)=Mens_Representatio
  ns( Male_Index).represented_Characteristics(Female_Index);
end
```

```
return
%end of exchange_Mate_Value function
```

file update_Self_Esteem.m

```
function Population=update_Self_Esteem(Population, Represented_Populations, Matches, Dates,
Male_Preference_Values, Parameters,Par_self_esteem, Choice)
Mens_Represented_Populations=Represented_Populations.mens;
Mens_Self_Esteem=Population.men.self_esteem;
Mens_Matches=Matches;
if(Choice==0)
   Womens_Represented_Populations=Represented_Populations.womens;
   Womens_Self_Esteem=Population.women.self_esteem;
   Womens_Matches=Matches.Matches_Women_Asking_Out;
   Mens_Matches=Matches.Matches_Men_Asking_Out;
   Size_Womens_Matches=size(Womens_Matches.men,1);
end

%The vice versa holds for the female case
F_gt_M_accept=Parameters.F_gt_M_accept;
F_lt_M_accept=Parameters.F_lt_M_accept;
F_gt_M_refuse=Parameters.F_gt_M_refuse;
F_lt_M_refuse=Parameters.F_lt_M_refuse;
Non_Riskers_Imp_Value=Parameters.Non_Riskers_Imp_Value;

if(Mens_Matches.men==0)
else
   Size_Mens_Matches=size(Mens_Matches.men,1);
   Men_Accepted=zeros(Size_Mens_Matches,1);

   %Check which of the Matches are realized as Dates
   %Check whether the male was refused or accepted
   for i=1:Size_Mens_Matches
      %check if man i has dated the woman,
      %that is, the Matched female i is among the Dates of the male

Dated_Index=ismember(Dates.women(find(Dates.men==Mens_Matches.men(i))),Mens_Matches.wome
n(i));
      %if she she is among the dates, Accepted is a nonzero value
      Men_Accepted(i)=sum(Dated_Index);
      %If the man has dated this woman the Accepted is 1.
      %If Accepted is 1 the male is accepted, otherwise if it
      %is 0 then he is refused
   end
   for i=1:Size_Mens_Matches
      if(((Mens_Represented_Populations(Mens_Matches.men(i)).represented_mateValues(
        Mens_Matches.women(i)))...
            >Mens_Self_Esteem(Mens_Matches.men(i)))&(Men_Accepted(i)==0))

Mens_Self_Esteem(Mens_Matches.men(i))=Mens_Self_Esteem(Mens_Matches.men(i))*F_gt_M_refuse;
      elseif(Men_Accepted(i)==0)

Mens_Self_Esteem(Mens_Matches.men(i))=Mens_Self_Esteem(Mens_Matches.men(i))*F_lt_M_refuse;
      elseif(((Mens_Represented_Populations(Mens_Matches.men(i)).represented_mateValues(
        Mens_Matches.women(i)))>...
            Mens_Self_Esteem(Mens_Matches.men(i))))

Mens_Self_Esteem(Mens_Matches.men(i))=Mens_Self_Esteem(Mens_Matches.men(i))*F_gt_M_accept;
      else

Mens_Self_Esteem(Mens_Matches.men(i))=Mens_Self_Esteem(Mens_Matches.men(i))*F_lt_M_accept;
      end
   end
   Single_Men=find(Population.men.maritial_Status==0);
   Non_Riskers=setdiff(Single_Men,Mens_Matches.men);
   if(ne(size(Non_Riskers,2),0))
```

```
    for i=1:size(Non_Riskers,1)
      if(Mens_Self_Esteem(Non_Riskers(i))<5) % 5 because the midvalue of self esteem is 5

Mens_Self_Esteem(Non_Riskers(i))=Mens_Self_Esteem(Non_Riskers(i))*Non_Riskers_Imp_Value;

        Mens_Self_Esteem(Non_Riskers(i))=Mens_Self_Esteem(Non_Riskers(i))*Non_Riskers_Dis_Valu
        e;
      end
    end
  end
  if(Choice==0)
    Women_Accepted=zeros(Size_Womens_Matches,1);
    for i=1:Size_Womens_Matches
      %check if man i has dated the woman,
      %that is, the Matched female i is among the Dates of the male
      Dated_Index=ismember(Dates.men(find(Dates.women==Womens_Matches.women(i))),
         Womens_Matches.men(i));
      %if she she is among the dates, Accepted is a nonzero value
      Women_Accepted(i)=sum(Dated_Index);
      %If the man has dated this woman the Accepted is 1.
      %If Accepted is 1 the male is accepted, otherwise if it
      %is 0 then he is refused
    end
    if(Womens_Matches.men==0)
    else
      for i=1:Size_Womens_Matches

          if(((Womens_Represented_Populations(Womens_Matches.women(i)).represented_mate
          Values(
          Womens_Matches.men(i)))>(Womens_Self_Esteem(Womens_Matches.women(i))))…
          &(Women_Accepted(i)==0))
            Womens_Self_Esteem(Womens_Matches.women(i))=Womens_Self_Esteem(
              Womens_Matches.women(i)) *F_gt_M_refuse;
        elseif(Women_Accepted(i)==0)
            Womens_Self_Esteem(Womens_Matches.women(i))=Womens_Self_Esteem(
              Womens_Matches.women(i)) *F_lt_M_refuse;
        elseif(((Womens_Represented_Populations(Womens_Matches.women(i)).
         represented_mateValues(Womens_Matches.men(i)))>...
            Womens_Self_Esteem(Womens_Matches.women(i))))
            Womens_Self_Esteem(Womens_Matches.women(i))=Womens_Self_Esteem(
              Womens_Matches.women(i))*F_gt_M_accept;
        else
            Womens_Self_Esteem(Womens_Matches.women(i))=Womens_Self_Esteem(
              Womens_Matches.women(i))*F_lt_M_accept;
        end
      end
      Single_Women=find(Population.women.maritial_Status==0);
      Non_Riskers=setdiff(Single_Women,Womens_Matches.women);
      for i=1:size(Non_Riskers,1)
        if(Womens_Self_Esteem(Non_Riskers(i))<5) % 5 because the midvalue of self esteem is 5
          Womens_Self_Esteem(Non_Riskers(i))=Womens_Self_Esteem(Non_Riskers(i))*
            Non_Riskers_Imp_Value;
         Womens_Self_Esteem(Non_Riskers(i))=Womens_Self_Esteem(Non_Riskers(i))*
            Non_Riskers_Dis_Value;
        end
        Womens_Self_Esteem=truncate(Womens_Self_Esteem, 0,10,'both');
        Population.women.self_esteem=Womens_Self_Esteem;
      end
    end
  end
  Mens_Self_Esteem=truncate(Mens_Self_Esteem, 0,10,'both');
  Population.men.self_esteem=Mens_Self_Esteem;
end
return
```

file update_Temperature.m

```
function Population=update_Temperature(Population,Represented_Populations, Dates,
   Duration_of_year, Par_self_esteem)

Mens_Age=Population.men.age;
Womens_Age=Population.women.age;

Parameter_1=0.017;
Parameter_2 = 700;
Parameter_3 = 8;
Parameter_4 = 1.6;
Parameter_5 = 0.051;
Parameter_6 = 1.3;

Shift_param = -12;
Age_dilation = 0.55;

Size_Men=size(Mens_Age,1);
Size_Women=size(Womens_Age,1);
for(i=1:Size_Men)
    Number_of_Dates=Population.men.last_Date(i,2);
      Date_exponent_vec=Parameter_1.*Age_dilation.*(Mens_Age(i)-Shift_param) ;
      Mens_Temperature(i)= ((Parameter_2./(Age_dilation.*(Mens_Age(i)-Shift_param)-
Parameter_3).^3)+ ...
         Parameter_4.*(log(Age_dilation*Parameter_5*(Mens_Age(i) -
          Shift_param)))./(((Number_of_Dates-1)+Parameter_6).^Date_exponent_vec);
end
for(j=1:Size_Women)
    Number_of_Dates=Population.women.last_Date(j,2);
      Date_exponent_vec=Parameter_1.*Age_dilation.*(Womens_Age(j)-Shift_param) ;
      Womens_Temperature(j)= ((Parameter_2./(Age_dilation.*(Womens_Age(j)-Shift_param)-
          Parameter_3).^3)+      Parameter_4.*(log(Age_dilation*Parameter_5*(Womens_Age(j) -
          Shift_param)))./(((Number_of_Dates-1)+Parameter_6).^Date_exponent_vec);

end

%Values of T can't be less than 0
Population.men.temperature=truncate(Mens_Temperature',0,'lower');
Population.women.temperature=truncate(Womens_Temperature',0,'lower');
%Don't calculate the T for the individuals that are married.
%The T for married people will remain 0
Population.men.temperature(find(Population.men.maritial_Status==1))=0;
Population.women.temperature(find(Population.women.maritial_Status==1))=0;
return
%end of function update_Temperature
```

file want_to_Marry.m

```
function [Population, Married_at_Age,New_Married]=want_to_Marry(Population, Pairs,
   Married_at_Age, Threshold)
global MARRIED_COUPLES_COUNTER;
Male_Threshold=Threshold.Male_Marrying_Threshold;
Female_Threshold = Threshold.Female_Marrying_Threshold;
Min_Age=Threshold.Min_Age;
Male_Fixed_Ages=fix(Population.men.age);
Female_Fixed_Ages=fix(Population.women.age);
if(Pairs.men==0)
    New_Married.men=0;
    New_Married.women=0;
else
    [Size_of_Pairs Dummy]=size(Pairs.men);
    index=1;
    New_Married.men=zeros(Size_of_Pairs,1);
    New_Married.women=zeros(Size_of_Pairs,1);
    for(i=1:Size_of_Pairs)
```

```
      if((Population.men.temperature(Pairs.men(i))<=Male_Threshold)&(
        Population.women.temperature( Pairs.women(i))<=Female_Threshold)...
        &(Population.men.maritial_Status(Pairs.men(i))==0)&(Population.women.maritial_Status(Pairs
        .women(i))==0)          %Martitial status: 1 is for married, 0 is for single
          Population.men.maritial_Status(Pairs.men(i))=1;
          Population.women.maritial_Status(Pairs.women(i))=1;
          New_Married.men(index)=Pairs.men(i);
          New_Married.women(index)=Pairs.women(i);
          MARRIED_COUPLES_COUNTER=MARRIED_COUPLES_COUNTER+1;
          Male_Age_Index=Male_Fixed_Ages(Pairs.men(i))-Min_Age+1;
          Female_Age_Index=Female_Fixed_Ages(Pairs.women(i))-Min_Age+1;
          Married_at_Age.males(Male_Age_Index)=Married_at_Age.males(Male_Age_Index)+1;
          Married_at_Age.females(Female_Age_Index)=Married_at_Age.females(Female_Age_Index)+1;
          index=index+1;
      end
   end
   % remove zeros in the arrays
   New_Married.men=New_Married.men(find(New_Married.men));
   New_Married.women=New_Married.women(find(New_Married.women));
end
return
```

file get_older.m

```
function [Population, Represented_Populations]=get_older(Population, Represented_Populations,
One_Year,Parameters,Par_self_esteem, Age_of_Death)
global COUNTER;
global SINGLE_DEAD_MEN_COUNTER;
global SINGLE_DEAD_WOMEN_COUNTER;
global MARRIED_DEAD_MEN_COUNTER;
global MARRIED_DEAD_WOMEN_COUNTER;
Womens_Age=Population.women.age;
Mens_Age=Population.men.age;

Womens_Age=Womens_Age+1/One_Year;
Mens_Age=Mens_Age+1/One_Year;

Size_of_Population=Parameters.Size_of_Population;
Female_Male_Ratio=Parameters.Female_Male_Ratio;
Parameters.Maximum_Init_Age=Parameters.Minimum_Init_Age+2;
Male=false;
Female=true;
Population.women.age=Womens_Age;
Population.men.age=Mens_Age;
%The men dying age is different for each individual.
%Generate age of deaths randomly
Max_Age_Array=zeros(size(Population.women.age));

Distr_Type=true; %0/false for uniform, 1/true for normal
% %assign random values with mean expected_value and deviation variance
Mean=Age_of_Death;
Variance=1;
[M_Size, N_Size]=size(Max_Age_Array);
% %the assigned values should be of type type('integer' or 'real')
Type=true; %0/false for real and 1/true for integer
Max_Age_Array=random_Value_Generator(Distr_Type, Mean, Variance, M_Size, N_Size, Type);
Max_Age_Array=truncate(Max_Age_Array, Age_of_Death-2,Age_of_Death+2,'both');

Dead_Men=find(Mens_Age>Max_Age_Array);
Dead_Women=find(Womens_Age>Max_Age_Array);

%Count the single and married individuals
Size_Dead_Men=size(Dead_Men,1);
Size_Dead_Women=size(Dead_Women,1);
if(~isempty(Dead_Men))
   Single=find(Population.men.maritial_Status(Dead_Men)==0);
```

```
   Married=find(Population.men.maritial_Status(Dead_Men)==1);
   MARRIED_DEAD_MEN_COUNTER=MARRIED_DEAD_MEN_COUNTER+size(Married,1);
   SINGLE_DEAD_MEN_COUNTER=SINGLE_DEAD_MEN_COUNTER+size(Single,1);
end
if(~isempty(Dead_Women))
   Single=find(Population.women.maritial_Status(Dead_Women)==0);
   Married=find(Population.women.maritial_Status(Dead_Women)==1);
   MARRIED_DEAD_WOMEN_COUNTER=MARRIED_DEAD_WOMEN_COUNTER+size(Married,1);
   SINGLE_DEAD_WOMEN_COUNTER=SINGLE_DEAD_WOMEN_COUNTER+size(Single,1);
end
if(~isempty(Dead_Men))
   Size_of_New_Males=size(Dead_Men,1);
   Parameters.Size_of_Population=Size_of_New_Males+Size_of_Population*(Female_Male_Ratio/
     (Female_Male_Ratio+1));

     Parameters.Female_Male_Ratio=(Size_of_Population*(Female_Male_Ratio/(Female_Male_Ratio+
     1)))/ Size_of_New_Males;
   [Men, Mens_Represented_Populations]=initiate_Subpopulation(Male,Parameters);
   [Population.men, Represented_Populations.mens]=include_new_individuals(Population.men,
     Represented_Populations.mens, Men, Mens_Represented_Populations, Dead_Men);
end
if(~isempty(Dead_Women))
   Size_of_New_Females=size(Dead_Women,1);

Parameters.Size_of_Population=Size_of_New_Females+Size_of_Population*(1/(Female_Male_Ratio+1
));

Parameters.Female_Male_Ratio=Size_of_New_Females/(Size_of_Population*(1/(Female_Male_Ratio+
1)));
   [Women, Womens_Represented_Populations]=initiate_Subpopulation(Female,Parameters);
   [Population.women,
     Represented_Populations.womens]=include_new_individuals(Population.women,
     Represented_Populations.womens, Women, Womens_Represented_Populations, Dead_Women);
end
return
%end of function get_older

function    [Subpopulation, Represented_Subpopulations]=include_new_individuals(Subpopulation,
Represented_Subpopulations, New_Subpopulation, New_Represented_Populations, Indices)
Subpopulation.sex(Indices)=New_Subpopulation.sex(:);
Subpopulation.ID(Indices)=New_Subpopulation.ID(:);
Subpopulation.characteristics(Indices,:)=New_Subpopulation.characteristics(:,:);
Subpopulation.preferences(Indices,:)=New_Subpopulation.preferences(:,:);
Subpopulation.own_mate_value(Indices)=New_Subpopulation.own_mate_value(:);
Subpopulation.temperature(Indices)=New_Subpopulation.temperature(:);
Subpopulation.age(Indices)=New_Subpopulation.age(:);
Subpopulation.initial_age(Indices)=New_Subpopulation.initial_age(:);
Subpopulation.self_esteem(Indices)=New_Subpopulation.self_esteem(:);
Subpopulation.maritial_Status(Indices)=New_Subpopulation.maritial_Status(:);
Subpopulation.experience(Indices)=New_Subpopulation.experience(:);
Subpopulation.last_Date(Indices,:)=New_Subpopulation.last_Date(:,:);
Represented_Subpopulations(Indices)=New_Represented_Populations(:);
return
```

file SimMenu.m

```
function varargout = SimMenu(varargin)

% parameters
% SIMMENU M-file for SimMenu.fig
%      SIMMENU, by itself, creates a new SIMMENU or raises the existing
%      singleton*.
%
%      H = SIMMENU returns the handle to a new SIMMENU or the handle to
%      the existing singleton*.
%
```

```
%     SIMMENU('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in SIMMENU.M with the given input arguments.
%
%     SIMMENU('Property','Value',...) creates a new SIMMENU or raises the
%     existing singleton*.  Starting from the left, property value pairs are
%     applied to the GUI before SimMenu_OpeningFunction gets called.  An
%     unrecognized property name or invalid value makes property application
%     stop.  All inputs are passed to SimMenu_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help SimMenu

% Last Modified by GUIDE v2.5 25-Mar-2002 12:19:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @SimMenu_OpeningFcn, ...
   'gui_OutputFcn',  @SimMenu_OutputFcn, ...
   'gui_LayoutFcn',  [] , ...
   'gui_Callback',   []);
if nargin & isstr(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before SimMenu is made visible.
function SimMenu_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

%Parameters for simmain
handles.Parameters.mate_choice_seed= sum(100*clock);
handles.Parameters.Par_get_older.Duration_of_year=8;
handles.Parameters.Par_SimMenu.Age_of_Death=50;

handles.Parameters.Par_initiate_Population.Size_of_Population=100;
handles.Parameters.Par_initiate_Population.Female_Male_Ratio=1;
handles.Parameters.Par_initiate_Population.Minimum_Init_Age=15;
handles.Parameters.Par_initiate_Population.Maximum_Init_Age=46;
handles.Parameters.Par_initiate_Population.Number_of_Last_Encounters=5;
%0 if there's equal choice and 1 if there is female choice
handles.Parameters.Par_initiate_Population.FemaleChoice=1;
% 0 if it is one dimensional, 1 if it is multidimensional (13) dimension
handles.Parameters.Par_initiate_Population.MateValueDimensions=1;
if(handles.Parameters.Par_initiate_Population.MateValueDimensions==1)
   handles.Parameters.Par_initiate_Population.Size_of_Characteristics=13;
   handles.Parameters.Par_initiate_Population.Male_Preference_Values=...
      [8.13 7.21 7.09 6.89 5.78 5.64  3.59 3.84 2.76 2.37 2.53 2.14 2.12;...
         1.96 2.05 1.54 1.78 2.02 1.8  2.15 2.02 1.68 1.9 1.68 1.76 2.72];
   handles.Parameters.Par_initiate_Population.Female_Preference_Values=...
```

```
        [8.40 7.48 7.35 5.18 5.51 5.60 4.18 3.22 3.89 3.82 2.05 1.88 1.45;...
           1.22 1.75 1.16 1.92 1.98 2.27 2.55 2.16 2.08 1.99 1.59 1.62 2.43];
     %the characteristics are from 1 to 13
     %kindness and understanding(1) , Exciting personality(2), intelligence(3),
     %Physical attractiveness(4), Good health(5), adaptability(6),
     %creativity(7), Desire for children(8),  College graduate(9),
     % Good earning capacity(10), Good heredity(11), Good housekeeper(12),
     %Religious orientation(13).

     handles.Parameters.Par_initiate_Population.Male_Characteristic_Values=...
        [5 5 5 5 5 5 5 5 5 5 5 5 5;...
           2 2 2 2 2 2 2 2 2 2 2 2 2];
     handles.Parameters.Par_initiate_Population.Female_Characteristic_Values=...
        [5 5 5 5 5 5 5 5 5 5 5 5 5;...
           2 2 2 2 2 2 2 2 2 2 2 2 2];

elseif(handles.Parameters.Par_initiate_Population.MateValueDimensions==0)
     handles.Parameters.Par_initiate_Population.Size_of_Characteristics=1;
     %First row is the mean second row is the standard deviation
     handles.Parameters.Par_initiate_Population.Male_Preference_Values=[5 ; 1];
     handles.Parameters.Par_initiate_Population.Female_Preference_Values=[5 ; 1];
     handles.Parameters.Par_initiate_Population.Male_Characteristic_Values=[5; 2];
     handles.Parameters.Par_initiate_Population.Female_Characteristic_Values=[ 5; 2];
end

handles.Parameters.Par_initiate_Population.Default_Characteristics_Value=3.5;

handles.Parameters.Par_assign_Contacts.Minimum_Number_of_Contacts=1;
handles.Parameters.Par_assign_Contacts.Maximum_Number_of_Contacts=3;
handles.Parameters.Par_assign_Contacts.Minimum_Contact_Duration=1;
handles.Parameters.Par_assign_Contacts.Maxiumum_Contact_Duration=4;

handles.Parameters.Par_ask_Out.number_of_last_dates_compared=5;

handles.Parameters.Par_date.Date_Duration=6;
handles.Parameters.Par_date.Last_Remembered=5;

handles.Parameters.Par_exchange_Characteristics.Distortion_St_Deviation=0.1;

handles.Parameters.Par_update_Self_Esteem.F_gt_M_accept=1.10;
handles.Parameters.Par_update_Self_Esteem.F_lt_M_accept=1;
handles.Parameters.Par_update_Self_Esteem.F_gt_M_refuse=0.95;
handles.Parameters.Par_update_Self_Esteem.F_lt_M_refuse=0.85;
handles.Parameters.Par_update_Self_Esteem.Non_Riskers_Imp_Value=1.015;
handles.Parameters.Par_update_Self_Esteem.Non_Riskers_Dis_Value=0.99;


handles.Parameters.Par_self_esteem.coefficient=1;
handles.Parameters.Par_self_esteem.female_temperature_constant=1;
%0 is dual strategy with no self esteem, 1 is with self esteem ,
handles.Parameters.Par_self_esteem.mode_of_self_esteem_integration=1;

handles.Parameters.Par_want_to_Marry.Male_Marrying_Threshold=0.38;
handles.Parameters.Par_want_to_Marry.Female_Marrying_Threshold=0.38;
handles.Parameters.Par_want_to_Marry.Min_Age=handles.Parameters.Par_initiate_Population.Minim
um_Init_Age;

handles.Number_of_Runs=1;


%Set all the numbers in the GUI
% set(handles.MaxRunInput,'String', num2str(handles.Parameters.Par_simmainGUI.Max_Iterations));
set(handles.SizeInput,'String',
num2str(handles.Parameters.Par_initiate_Population.Size_of_Population));
% set(handles.FemaleMaleInput,'String',
num2str(handles.Parameters.Par_initiate_Population.Female_Male_Ratio));
```

```
set(handles.CharacteristicsInput,'String',
num2str(handles.Parameters.Par_initiate_Population.Size_of_Characteristics));

set(handles.MinContactInput,'String',
num2str(handles.Parameters.Par_assign_Contacts.Minimum_Number_of_Contacts));
set(handles.MaxContactInput,'String',
num2str(handles.Parameters.Par_assign_Contacts.Maximum_Number_of_Contacts));
set(handles.MinDurationInput,'String',
num2str(handles.Parameters.Par_assign_Contacts.Minimum_Contact_Duration));
set(handles.MaxDurationInput,'String',
num2str(handles.Parameters.Par_assign_Contacts.Maxiumum_Contact_Duration));

set(handles.LastEncountersInput,'String',
num2str(handles.Parameters.Par_initiate_Population.Number_of_Last_Encounters));

set(handles.DateDurationInput,'String', num2str(handles.Parameters.Par_date.Date_Duration));

set(handles.MenTemperatureThresholdInput,'String',
num2str(handles.Parameters.Par_want_to_Marry.Male_Marrying_Threshold));
set(handles.WomenTemperatureThresholdInput,'String',
num2str(handles.Parameters.Par_want_to_Marry.Female_Marrying_Threshold));

set(handles.YearIterationsInput,'String',
num2str(handles.Parameters.Par_get_older.Duration_of_year));

if (handles.Parameters.Par_self_esteem.mode_of_self_esteem_integration==0)
   set(handles.radiobutton1,'Value', 1);
else
   set(handles.radiobutton2,'Value', 1);
end
if(handles.Parameters.Par_initiate_Population.MateValueDimensions==1)
   set(handles.MultiDradiobutton,'Value', 1);
else
   set(handles.SingleDradiobutton,'Value', 1);
end
if (handles.Parameters.Par_initiate_Population.FemaleChoice==1 )
   set(handles.radiobutton12,'Value', 1);
else
   set(handles.radiobutton13,'Value', 1);
end
set(handles.radiobutton5,'Value', 1);
set(handles.NumberRunsInput,'String', num2str(handles.Number_of_Runs));
guidata(hObject, handles);
return

% --- Outputs from this function are returned to the command line.
function varargout = SimMenu_OutputFcn(hObject, eventdata, handles)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function ReproductiveInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function ReproductiveInput_Callback(hObject, eventdata, handles)
%Convert (numeric) String input to double
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'End of Reproductive Life');
if(Check_Value)
   handles.Parameters.Par_SimMenu.Age_of_Death=user_entry;
   guidata(hObject, handles);
end
```

```
% --- Executes during object creation, after setting all properties.
function SizeInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function SizeInput_Callback(hObject, eventdata, handles)
%Convert (numeric) String input to double
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Size of Population');
if(Check_Value)
   handles.Parameters.Par_initiate_Population.Size_of_Population=user_entry;
   guidata(hObject, handles);
end

%  -- Executes during object creation, after setting all properties.
function CharacteristicsInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function CharacteristicsInput_Callback(hObject, eventdata, handles)
%returns contents of FemaleMaleInput as a double
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Number of Characteristics');
if(Check_Value)
   handles.Parameters.Par_initiate_Population.Size_of_Characteristics=user_entry+1;
   guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function MinContactInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function MinContactInput_Callback(hObject, eventdata, handles)
%returns contents of MinContactInput as a double
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Min # of Contacts');
if(Check_Value)
   handles.Parameters.Par_assign_Contacts.Minimum_Number_of_Contacts=user_entry;
   guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function MaxContactInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function MaxContactInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Max # of Contacts');
if(Check_Value)
   handles.Parameters.Par_assign_Contacts.Maximum_Number_of_Contacts=user_entry;
   guidata(hObject, handles);
```

```
end

% --- Executes during object creation, after setting all properties.
function MinDurationInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function MinDurationInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Minimum Contact Duration');
if(Check_Value)
   handles.Parameters.Par_assign_Contacts.Minimum_Contact_Duration=user_entry;
   guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function MaxDurationInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function MaxDurationInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Max Contact Duration');
if(Check_Value)
   handles.Parameters.Par_assign_Contacts.Maximum_Contact_Duration=user_entry;
   guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function DateDurationInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function DateDurationInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Duration of Date');
if(Check_Value)
   handles.Parameters.Par_date.Date_Duration=user_entry;
   guidata(hObject, handles);
end

% --- Executes during object creation, after setting all properties.
function LastEncountersInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function LastEncountersInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Number of Last Encounters');
if(Check_Value)
   handles.Parameters.Par_contact.Number_of_Last_Encounters=user_entry;
   guidata(hObject, handles);
end
```

```
% --- Executes during object creation, after setting all properties.
function YearIterationsInput_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function YearIterationsInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, '# of Iterations per Year');
if(Check_Value)
    handles.Parameters.Par_get_older.Duration_of_year=user_entry;
    guidata(hObject, handles);
end

function MenTemperatureThresholdInput_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function MenTemperatureThresholdInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Men''s Temperature Threshold');
if(Check_Value)
    handles.Parameters.Par_want_to_Marry.Male_Marrying_Threshold=user_entry;
    guidata(hObject, handles);
end

function WomenTemperatureThresholdInput_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function WomenTemperatureThresholdInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Women''s Temperature Threshold');
if(Check_Value)
    handles.Parameters.Par_want_to_Marry.Female_Marrying_Threshold=user_entry;
    guidata(hObject, handles);
end

% --- Executes on button press in CloseButton.
function CloseButton_Callback(hObject, eventdata, handles)
%close all figures
close all;

% --- Executes on button press in RunButton.
function RunButton_Callback(hObject, eventdata, handles)
Number_of_Runs=handles.Number_of_Runs;
for i=1:Number_of_Runs
    ResultSim=simmainGUI(handles);
end
handles=ResultSim.handles;
guidata(hObject, handles);
handles.Married=ResultSim.total_number_of_Married;
Percentage_Married=ResultSim.married_Percentage;
intracouple_correlation=ResultSim.intracouple_correlation;
rel_intracouple_correlation=ResultSim.rel_intracouple_correlation;
guidata(hObject, handles);
```

```
set(handles.OutputText, 'Visible', 'on');
Output_String=strcat(num2str(handles.Married),' ind''s');
set(handles.OutputOutput, 'String', Output_String);
Percentage_String=strcat(num2str(Percentage_Married),' %');
set(handles.PercentageOutput, 'String', Percentage_String);
Corr_String=strcat('ObjR', num2str(intracouple_correlation));
set(handles.ObjCorrOutput, 'String', Corr_String);
Corr_String=strcat('SubR',num2str(rel_intracouple_correlation));
set(handles.SubCorrOutput, 'String', Corr_String);

function SaveParamsButton_Callback(hObject, eventdata, handles)
Parameters=handles.Parameters;
save parameters.mat Parameters;
guidata(hObject, handles);

% --- Executes on button press in ClearButton.
function ClearButton_Callback(hObject, eventdata, handles)
%Clear all graphs in the main window
axes(handles.axes3);
cla;
axes(handles.axes5);
cla;
axes(handles.axes6);
cla;
set(handles.OutputText, 'Visible', 'off');
set(handles.OutputOutput, 'String', '');
set(handles.PercentageOutput, 'String', '');
set(handles.ObjCorrOutput, 'String', '');
set(handles.SubCorrOutput, 'String', '');
%Close all figures except the main window figure
%Set the handle of the main figure invisible
set(handles.SimFigure, 'HandleVisibility','off');
%close all figures
close all;
%Set the handle of the main figure visible
set(handles.SimFigure, 'HandleVisibility','on');

function Popout3Button_Callback(hObject, eventdata, handles)
handles.popout_axes3=Popout(handles.axes3,handles.axes9);
guidata(hObject, handles);

function Popout5Button_Callback(hObject, eventdata, handles)
handles.popout_axes5=Popout(handles.axes5,handles.axes11);
guidata(hObject, handles);

function Popout6Button_Callback(hObject, eventdata, handles)
handles.popout_axes6=Popout(handles.axes6, handles.axes12);
guidata(hObject, handles);

function new_axes=Popout(Axes1,Axes2)
if(strcmp(get(Axes1,'Visible'),'on'))
   Axes=Axes1;
else
   Axes=Axes2;
end
pop=figure;     % Create a new figure
try
   new_axes = copyobj(Axes,pop);
   set(new_axes,'Position',[0.13 0.11 0.775 0.815]);
catch
   new_axes = axes;
end

function MoreButton_Callback(hObject, eventdata, handles)
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
```

```
    % toggle button is pressed
    set(handles.frame3, 'Visible', 'off');
    set(handles.frame6, 'Visible', 'off');
    set(handles.frame8, 'Visible', 'off');
    set(handles.frame10, 'Visible', 'off');
    set(handles.frame12, 'Visible', 'off');
    set(handles.frame14, 'Visible', 'off');
    set(handles.frame15, 'Visible', 'off');
    set(handles.YearIterationsText, 'Visible', 'off');
    set(handles.YearIterationsInput, 'Visible', 'off');
    set(handles.SizeText, 'Visible', 'off');
    set(handles.SizeInput, 'Visible', 'off');
    set(handles.MinContactText, 'Visible', 'off');
    set(handles.MinContactInput, 'Visible', 'off');
    set(handles.MaxContactText, 'Visible', 'off');
    set(handles.MaxContactInput, 'Visible', 'off');
    set(handles.DateDurationText, 'Visible', 'off');
    set(handles.DateDurationInput, 'Visible', 'off');
    set(handles.MinDurationText, 'Visible', 'off');
    set(handles.MinDurationInput, 'Visible', 'off');
    set(handles.MaxDurationText, 'Visible', 'off');
    set(handles.MaxDurationInput, 'Visible', 'off');
    set(handles.CharacteristicsInput, 'Visible', 'off');
    set(handles.ReproductiveInput, 'Visible', 'off');
    set(handles.ReproductiveText, 'Visible', 'off');
    set(handles.LastEncountersText, 'Visible', 'off');
    set(handles.LastEncountersInput, 'Visible', 'off');
    set(handles.MenTemperatureThresholdText, 'Visible', 'off');
    set(handles.MenTemperatureThresholdInput, 'Visible', 'off');
    set(handles.WomenTemperatureThresholdText, 'Visible', 'off');
    set(handles.WomenTemperatureThresholdInput, 'Visible', 'off');
    set(handles.radiobutton1,'Visible','off');
    set(handles.radiobutton2,'Visible','off');
    set(handles.SingleDradiobutton,'Visible','off');
    set(handles.MultiDradiobutton,'Visible','off');
    set(handles.radiobutton12,'Visible','off');
    set(handles.radiobutton13,'Visible','off');
    set(handles.NumberRunsInput, 'Visible', 'off');
    set(handles.NumberRunsText, 'Visible', 'off');
    %%%%%%%%%%%
    set(handles.Parameters2_Frame,'Visible','on');
    set(handles.frame10, 'Visible', 'on');
    set(handles.frame11, 'Visible', 'on');
    set(handles.frame12, 'Visible', 'on');
    set(handles.RandomSeedInput, 'Visible', 'on');
    set(handles.radiobutton5,'Visible','on');
    set(handles.radiobutton6,'Visible','on');
    guidata(hObject, handles);
elseif button_state == get(hObject,'Min')
    % toggle button is not pressed
    set(handles.Parameters2_Frame,'Visible','off');
    set(handles.frame10, 'Visible', 'off');
    set(handles.frame11, 'Visible', 'off');
    set(handles.frame12, 'Visible', 'off');
    set(handles.radiobutton5,'Visible','off');
    set(handles.radiobutton6,'Visible','off');
    set(handles.RandomSeedInput, 'Visible', 'off');
    set(handles.frame3, 'Visible', 'on');
    set(handles.frame6, 'Visible', 'on');
    set(handles.frame10, 'Visible', 'on');
    set(handles.frame12, 'Visible', 'on');
    set(handles.frame8, 'Visible', 'on');
    set(handles.frame14, 'Visible', 'on');
    set(handles.frame15, 'Visible', 'on');
    set(handles.YearIterationsText, 'Visible', 'on');
    set(handles.YearIterationsInput, 'Visible', 'on');
```

```
    set(handles.SizeText, 'Visible', 'on');
    set(handles.SizeInput, 'Visible', 'on');
    set(handles.MinContactText, 'Visible', 'on');
    set(handles.MinContactInput, 'Visible', 'on');
    set(handles.MaxContactText, 'Visible', 'on');
    set(handles.MaxContactInput, 'Visible', 'on');
    set(handles.DateDurationText, 'Visible', 'on');
    set(handles.DateDurationInput, 'Visible', 'on');
    set(handles.MinDurationText, 'Visible', 'on');
    set(handles.MinDurationInput, 'Visible', 'on');
    set(handles.MaxDurationText, 'Visible', 'on');
    set(handles.MaxDurationInput, 'Visible', 'on');
    set(handles.CharacteristicsInput, 'Visible', 'on');
    set(handles.ReproductiveInput, 'Visible', 'on');
    set(handles.ReproductiveText, 'Visible', 'on');
    set(handles.LastEncountersText, 'Visible', 'on');
    set(handles.LastEncountersInput, 'Visible', 'on');
    set(handles.MenTemperatureThresholdText, 'Visible', 'on');
    set(handles.MenTemperatureThresholdInput, 'Visible', 'on');
    set(handles.WomenTemperatureThresholdText, 'Visible', 'on');
    set(handles.WomenTemperatureThresholdInput, 'Visible', 'on');
    set(handles.radiobutton1,'Visible','on');
    set(handles.radiobutton2,'Visible','on');
    set(handles.SingleDradiobutton,'Visible','on');
    set(handles.MultiDradiobutton,'Visible','on');
    set(handles.radiobutton12,'Visible','on');
    set(handles.radiobutton13,'Visible','on');
    set(handles.NumberRunsText, 'Visible', 'on');
    set(handles.NumberRunsInput, 'Visible', 'on');
    guidata(hObject, handles);
end

function MoreGraphsButton_Callback(hObject, eventdata, handles)
button_state = get(hObject,'Value');
if button_state == get(hObject,'Max')
    % toggle button is pressed
    set(handles.axes3, 'Visible', 'off');
    set(handles.legend3,'Visible', 'off');
    set(allchild(handles.axes3),'Visible','off');
    %%%%
    set(handles.axes5, 'Visible', 'on');
    set(handles.legend5,'Visible', 'on');
    set(allchild(handles.axes5),'Visible','on');
    set(handles.axes6, 'Visible', 'on');
    set(handles.legend6,'Visible', 'on');
    set(allchild(handles.axes6),'Visible','on');
    guidata(hObject, handles);
elseif button_state == get(hObject,'Min')
    % toggle button is not pressed
    set(handles.axes3, 'Visible', 'on');
    set(handles.legend3,'Visible', 'on');
    set(allchild(handles.axes3),'Visible','on');
    %%%%
    set(handles.axes5, 'Visible', 'off');
    set(handles.legend5,'Visible', 'off');
    set(allchild(handles.axes5),'Visible','off');
    set(handles.axes6, 'Visible', 'off');
    set(handles.legend6,'Visible', 'off');
    set(allchild(handles.axes6),'Visible','off');
    guidata(hObject, handles);
end

function radiobutton1_Callback(hObject, eventdata, handles)
off = [handles.radiobutton2];
mutual_exclude(off);
handles.Parameters.Par_self_esteem.mode_of_self_esteem_integration=0;
```

```
guidata(hObject, handles);

function radiobutton2_Callback(hObject, eventdata, handles)
off = [handles.radiobutton1];
mutual_exclude(off);
handles.Parameters.Par_self_esteem.mode_of_self_esteem_integration=1;
guidata(hObject, handles);

function radiobutton5_Callback(hObject, eventdata, handles)
off = [handles.radiobutton6];
mutual_exclude(off);
handles.Parameters.mate_choice_seed = sum(100*clock);
set(handles.RandomSeedInput,'String','');
set(handles.RandomSeedInput,'Enable','inactive');
guidata(hObject, handles);

function radiobutton6_Callback(hObject, eventdata, handles)
off = [handles.radiobutton5];
mutual_exclude(off);
guidata(hObject, handles);

function SingleDradiobutton_Callback(hObject, eventdata, handles)
off = [handles.MultiDradiobutton];
mutual_exclude(off);
handles.Parameters.Par_initiate_Population.MateValueDimensions=0;
guidata(hObject, handles);

function MultiDradiobutton_Callback(hObject, eventdata, handles)
off = [handles.SingleDradiobutton];
mutual_exclude(off);
handles.Parameters.Par_initiate_Population.MateValueDimensions=1;
guidata(hObject, handles);

function radiobutton12_Callback(hObject, eventdata, handles)
off = [handles.radiobutton13];
mutual_exclude(off);
%0 if there's equal choice and 1 if there is female choice
handles.Parameters.Par_initiate_Population.FemaleChoice=1;
guidata(hObject, handles);

function radiobutton13_Callback(hObject, eventdata, handles)
off = [handles.radiobutton12];
mutual_exclude(off);
%0 if there's equal choice and 1 if there is female choice
handles.Parameters.Par_initiate_Population.FemaleChoice=0;
guidata(hObject, handles);

function mutual_exclude(off)
set(off,'Value',0);

%--- Executes during object creation, after setting all properties.
function RandomSeedInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
guidata(hObject, handles);

function RandomSeedInput_Callback(hObject, eventdata, handles)
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Random Seed');
if(Check_Value)
   handles.Parameters.mate_choice_seed=user_entry;
   guidata(hObject, handles);
end
```

```
% --- Executes during object creation, after setting all properties.
function NumberRunsInput_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
guidata(hObject, handles);

function NumberRunsInput_Callback(hObject, eventdata, handles)
%Convert (numeric) String input to double
user_entry = str2double(get(hObject,'String'));
Check_Value=Check_Input(user_entry, hObject, handles, 'Number of Runs');
if(Check_Value)
   handles.Number_of_Runs=user_entry;
   guidata(hObject, handles);
end

function Value=Check_Input(user_entry, hObject, handles, Input_Label)
if (isnan(user_entry))
   %     Erase the value in the edit box
   set(hObject,'String','');
   %     Display error message
   ErrorMsg=['You must enter a numeric value in ', Input_Label,'.'];
   error=errordlg(ErrorMsg,'Bad Input','modal')
   Value=0;
   %     Determine whether val is a number between MIN and MAX
elseif ((user_entry >= get(hObject,'Max')) & ...
       (user_entry <= get(hObject,'Min')))
   %     Erase the value in the edit box
   set(hObject,'String','');
   %     Display the error message
   error_String=['You have entered an invalid entry in ', Input_Label, '. The input should be in the
range of ', num2str(get(hObject,'Min')), '-', num2str(get(hObject,'Max')), '.']
   error=errordlg(error_String,'Bad Input','modal')
   Value=0;
else
   Value=1;
end
clear('Input_Label');
% end of Check_Input function
```