J Econ Interac Coord (2006) 1:171–187
DOI 10.1007/s11403-006-0014-7

REGULAR ARTICLE

Evolving densities in continuous strategy games through particle simulations

Julide Yazar

Published online: 7 September 2006

Abstract Many cases of strategic interaction between agents involve a continuous set of choices. It is natural to model these problems as continuous space games. Consequently, the population of agents playing the game will be represented with a density function defined over the continuous set of strategy choices. Simulating evolutionary dynamics on continuous strategy spaces is a challenging problem. The classic approach of discretizing the strategy space is ineffective for multidimensional strategy spaces. We present a principled approach to simulation of adaptive dynamics in continuous space games using sequential Monte Carlo methods. Sequential Monte Carlo methods use a set of weighted random samples, also named particles to represent density functions over multidimensional spaces. Sequential Monte Carlo methods provide computationally efficient ways of computing the evolution of probability density functions. We employ resampling and smoothing steps to prevent particle degeneration problem associated with particle estimates. The resulting algorithm can be interpreted as an agent based simulation with elements of natural selection, regression to mean and mutation. We illustrate the performance of the proposed simulation technique using two examples: continuous version of the repeated prisoner dilemma game and evolution of bidding functions in first-price closed-bid auctions.

Keywords Stochastic and dynamic games · Evolutionary games · Agent based simulations · Monte Carlo methods · Computational techniques

JEL Classification C73 · C15 · C63

J. Yazar (✉)
Department of Economics, Ohio Wesleyan University, Delaware, OH 43015, USA
e-mail: jjyazar@owu.edu

1 Introduction

Evolutionary games analyze the evolution of strategic choices of interacting agents over time. In particular, evolutionary dynamic models hypothesize a continuum of interacting agents and represent the evolutionary process in terms of deterministic differential equations. The role of deterministic evolutionary dynamics in approximating the stochastic processes which results from repeated random interactions of players has been studied extensively (Benaim and Weibull 2003; Boylan 1992; Corradi and Sarin 2000; Dawid 1999).

In many cases of strategic interaction between agents involve a continuous set of choices, where each agent can choose from an infinite set of strategies. Therefore it is natural to model these problems using continuous space games. Examples are quantity, price competition between firms, choice of timing, investment, effort and all mixed strategy games. Consequently the population of agents playing the game will be represented with a density function defined over the continuous set of strategy choices. Simulating evolution of this population is a challenging problem. Classical methods employ deterministic techniques such as discretization of the strategy space (Friedman and Yellin 1997) and Riemann approximation to the expectation integrals in the dynamics. However, these methods do not scale well as the dimensionality of the state space increases because the number of grid points increases exponentially with the number of dimensions. Many practical applications require models of strategic competion in multi-dimensional strategy spaces. In contrast to the discretization approaches, the error rate of the Monte Carlo integration methods are independent of the number of dimensions. For this reason several researchers resorted to genetic algorithm based approaches with heuristic modifications to simulate their theoretical models (Killingback and Doebeli 2002; Killingback et al. 1999; Wahl and Nowak 1999). In this paper we develop a principled approach to simulation of evolutionary dynamics in continuous strategy games. The method we present is based on sequential Monte Carlo methods and can be interpreted as an agent based simulation with elements of natural selection, regression to mean and mutation. The proposed algorithm solves the over-dispersion problem associated with random mutations and does not require artificial methods such as cooling schedules for mutation variance for convergence. By providing an accurate agent based simulation method for evolutionary dynamics this paper contributes to previous work by Carpenter (2002) which compare agent-based simulations with corresponding game dynamics. The techniques presented in the paper are applicable to a wide variety of dynamic evolution and learning methods.

Specifically, the probability density over the strategy space is approximated with a population of weighted set of random samples, from hereon called particles. The particle approach to density estimation provides a computationally efficient way of modeling the evolution of probability density functions in high dimensional strategy spaces. However, a straightforward implementation of the particle simulation will suffer from degeneration of the discrete random measure (i.e. all particles with the exception of a few will be assigned vanishing

weights). A resampling step and smoothing methods are employed to prevent particle degeneration problem associated with particle estimates.

The organization of the remaining sections is as follows. First, we review adaptive dynamics models in discrete and continuous spaces. Then we discuss sequential Monte Carlo methods for simulation of dynamic models with states represented by probability density functions. Then we present a method based on sequential Monte-Carlo methods for simulation of continuous space replicator dynamics. We discuss resampling and smoothing steps to avoid particle degeneration. Then the performance of the proposed method is illustrated on two examples: a continuous space version of the repeated Prisoner Dilemma game proposed by Killingback et al. (1999) and evolution of optimal bidding behavior in closed-bid first-price auctions. In conclusion, we provide a brief comparison of the resulting algorithm with the genetic programming techniques.

2 Models of adaptive dynamics

2.1 Evolutionary games on continuous strategy spaces

We consider normal form symmetric games between two agents, agent 1 and agent 2. Each agent can choose a strategy s^i from the set of strategy choices S. As a result each agent will get the payoff of $\pi(s^i, s^j)$, where s^j denotes the strategy choice of agent i's opponent. The payofff function and the strategy choices S are common knowledge. In the following time is assumed to be measured discretely ($t = 1, 2, \ldots$). At each time step, members of a large population is matched randomly to play the game. The frequency of strategy choices chosen by the population at each time step t is given $\mu(t)$, a probability density function defined over the set S. We are interested in modeling the evolution of the density function $\mu(t)$ over time.

2.2 Replicator dynamics in discrete strategy spaces

We first review replicator dynamics in discrete strategy spaces. Following the presentation in Vega-Redondo (1996), we assume the set S is a finite set with discrete set of choices $S = \{s_1, s_2, \ldots, s_n\}$. Then the population strategy profile is a vector valued function $\mu(t) = \{\mu_1(t), \mu_2(t), \ldots, \mu_n(t)\}$ with $\sum_i \mu_i(t) = 1$, where $\mu_i(t)$ gives the proportion of agents that play the strategy s_i in the population at time t. We assume that if a member of the population plays the strategy s_i then the expected number of its offsprings is proportional to its expected payoff which is given by:

$$E(s_i, \mu(t)) = \sum_{s \in S} \pi(s_i, s)\mu_i(t) \tag{1}$$

Under these assumptions the proportion of agents that are playing strategy s_i at time $t + 1$ is given by:

$$\mu_i(t+1) = \mu_i(t)\frac{E(s_i,\mu(t))}{\sum_j \mu_j(t)E(s_j,\mu(t))} = \mu_i(t)\frac{\pi_i(\mu(t))}{E(\mu(t),\mu(t))} \quad (2)$$

To obtain the well known continuous time version of the replicator dynamics, we need to define how the evolution of the population will proceed in a fractional time Δ. Assume on a time interval Δ only $O(\Delta)$ percent of the population will play the game and will have offspings based on their payoff, whereas $O(1-\Delta)$ percent of the population won't be affected.

$$\mu_i(t+\Delta) = \frac{c\Delta\mu_i(t)E(s_i,\mu(t)) + (1-c\Delta)\mu_i(t)}{\sum_j [c\Delta\mu_j(t)E(s_j,\mu(t)) + (1-c\Delta)\mu_j(t)]} \quad (3)$$

Then the continuous time version of the replicator dynamics follows from the definition of the derivative

$$\dot{\mu}_i(t) \equiv \frac{\mu_i(t+\Delta)-\mu_i(t)}{\Delta} = \mu_i(t)c\left[E(s_i,\mu(t)) - E(\mu(t),\mu(t))\right] \quad (4)$$

where c is a proportionality constant that can be absorbed in the definition of the time index. The intuition behind the replicator dynamics is that the rate change in the relative frequency of a strategy is proportional to the differential payoff of that strategy. We note that $\sum_i \dot{\mu}(t) = 0$ ensuring that $\sum_i \mu_i(t) = 1$ for all t provided that population starts with an initial state $\mu(0)$ satisfiying $\sum_i \mu_i(0) = 1$

2.3 Replicator dynamics in continuous strategy spaces

The dynamic evolution for continuous strategy space games has been considered in Bomze (1990), Friedman and Yellin (1997), Oechssler and Riedel (2001) and Vega-Redondo (1996). The set of strategy choices S is a Borel set with associated σ algebra. The population strategy profile at time t is a probability density function $\mu^t(s)$ defined on S. Let A be a subset of S, then the proportion of the population playing a strategy s in A is $\int_A \mu^t(s)\mathrm{d}s$. Again we assume agents playing the strategy s will be replaced by a number of their offsprings proportional to their expected payoff at time t, $E(s,\mu^t) \equiv \int \pi(s,s')\mu^t(s')\mathrm{d}s'$. leading to the replicator dynamics for evolution in continuous strategy games

$$\dot{\mu}^t(s) = \mu^t(s)\left[E(s,\mu^t) - E(\mu^t,\mu^t)\right] \quad (5)$$

The proportionality constant has been absorbed into the definition of the time scale. The continuous time dynamics can be simulated with a discrete time version of the dynamics:

$$\mu^{t+1}(s) = \alpha\mu^t(s)\left[1 + \Delta\left(E(s,\mu^t) - E(\mu^t,\mu^t)\right)\right] \quad (6)$$

for sufficiently small $\Delta > 0$, where α is a constant chosen to ensure that the density μ^{t+1} is proper probability measure with $\int \mu^{t+1}(s)\mathrm{d}s = 1$. Equivalently $\alpha = \left(\int \mu^t(s)\left[1 + \Delta(E(s,\mu^t) - E(\mu^t,\mu^t))\right]\mathrm{d}s\right)^{-1}$. We note that for special cases of payoff functions the continuous and discrete version of the dynamics can exhibit different dynamic behaviour even in the limit Δ converges to zero (Fudenberg and Levine 1998).

There are many other dynamic models of learning and adaptation for continuous strategy games that we could consider. For example local adjustment path (LAP) dynamics consider adaptation given by the gradient of the fitness surface $\partial E(s,\mu^t)/\partial s$ (Friedman and Yellin 1997).

$$\dot{\mu}^t(s) = -\frac{\partial \mu^t(s)}{\partial s}\frac{\partial E(s,\mu^t)}{\partial s} - \frac{\partial^2 E(s,\mu^t)}{\partial s^2}\mu^t(s) \tag{7}$$

3 Evolving densities through particle simulations

3.1 Sequential Monte Carlo methods

Sequential Monte Carlo (SMC) methods is a collection of simulation based tools which provide computationally efficient means to compute posterior distributions (Doucet et al. 2001). SMC methods differ from the classical grid-based computation of posterior distributions in their flexible implementations. Over the past 5 years there has been a surging interest in SMC methods with numerous applications in Bayesian filtering, Neural Network training, Controls, Shape Recognition. SMC methods are variations of the idea to represent the posterior distribution using weighted samples.

Monte Carlo (MC) methods are mainly used to calculate the expectation integrals using samples from the underlying distribution. SMC methods are applicable if in each iteration the posterior distribution of the next state of the system depends on expectation integrals of the distribution of the current state. Typically, it is not possible to sample directly from the distribution of the current state directly. Therefore one has to resort to importance sampling methods, where samples obtained from a "proposal" distribution are weighted to obtain an approximation to the required state distribution. A recursive implementation of the importance sampling will have problems with sample degeneration, where after a few iteration only few samples with large weights are present in the distribution estimate. Gordon et al. (1993) introduced a resampling step to avoid this problem leading to an efficient implementation of sequential Monte Carlo.

3.2 Sequential Monte Carlo simulation of replicator dynamics

Sequential Monte Carlo simulation of replicator dynamics relies on constructing a sample based approximate to the current population state μ^t

$$\hat{\mu}^t = \sum_{j=1}^{M} w_j^t \delta(s_j^t) \tag{8}$$

from a sample based estimate for $\hat{\mu}^{t-1}$, population state at $t-1$. $\delta(s_j^t)$ and w_j^t are the samples and their associated weights in the strategy space at time t. We adapt the Bootstrap filter derived in Gordon et al. (1993) to simulate the replicator dynamics:

1. Initialization $t = 0$
   For $i = 1, \ldots, N$ sample s_i^1 from initial distribution population state μ^0 and set $t = 1$.
2. Importance sampling step
   For $i = 1, \ldots, N$ evaluate the importance weights $\tilde{w}_i^t = \pi^t(s_i^t)$.
   Normalize the importance weights.
3. Selection step
   Resample with replacement N particles $\tilde{s}_i^{t+1}$ from the set s_i^t according to the importance weights $\tilde{w}_i^t$.
4. Smoothing step
   Every particle $i = 1, \ldots, N$ is moved with small random perturbation according to distribution $p^t(x|\tilde{s}_i)$ to obtain new samples s_i^{t+1}.
   Set $t = t + 1$.

First we note that Steps 2 and 3 result in similar dynamics employed by Vega-Redondo (1997) and Schenk-Hoppe (2000) where a small percentage of players are replaced with the highest performing strategies. Although here the percentage of the agents being replaced is not constant but depends on the relative performance of the strategies in consideration.

After the resampling step, a significant number of the samples could be identical. This problem becomes more dominant when the weights have a peaky structure. The last smoothing step is a remedy for this degeneration problem and it also recognizes the deficiency of the discrete samples in representing continuous valued random variables. In the smoothing step an appropriate random perturbation is added to each sample.

We note that the perturbation distribution is time dependent, ideally it should adjust the level of smoothing based on the extent of the distribution and the number of samples in the estimate. Clearly, the different dimensions require different degrees of smoothing depending on their support and variance. This perturbation approach is closely related to the kernel density smoothing, where the point-mass estimator of the distribution:

$$\hat{\mu}^t = \sum_{j=1}^{M} w_j^t \delta(s_j^t) \tag{9}$$

is replaced with a smoothed estimator of the form:

$$\hat{\mu_S}^t = \sum_{j=1}^{M} w_j^t \Omega(\mathcal{M}(s_j^t), \mathcal{S}^t) \tag{10}$$

where Ω is a unimodal symmetric density function centered at $\mathcal{M}$ with scale parameter $\mathcal{S}$. The sample depletion problem after the resampling step is remedied by sampling from $\hat{\mu_S}$, which is equivalent to sampling with replacement from the set of particles $\{s_j^t\}$ and adding a smoothing term sampled from the smoothing kernel Ω.

One popular choice for the smoothing step is the Gaussian kernel with diagonal covariance matrix. In each dimension the standard deviation is adjusted to $c(M-m)/N^{(1/d)}$ $(M-m)$ represents the support length in that dimension d is the number of dimensions of the strategy space, where c is a scaling coefficient. It sets the size of the smoothing kernel to c percent of the grid size. The grid size is computed under the assumption that the samples are uniformly distributed over the support of the population density. However, this method result in a posterior distribution that is *over-dispersed* as compared to the variance of the weighted samples at the end of the selection step. Ideally, the covariance and mean of the particles after the smoothing step should be identical to their covariance and mean before smoothing term is applied.

To overcome the over-dispersion problem, we use a shrinkage rule suggested by Liu and West (2001). Specifically for the smoothing kernel $p^t(x|\tilde{s}_i)$, we use Gaussian kernels $p^t(x|\tilde{s}_i)$ with mean $m(t) = a\tilde{s}_i + (1-a)\bar{s}$ and covariance $S(t) = \sqrt{1-a^2}\Sigma_s$, where $\bar{s}$ and Σ_s are the mean and covariance of the weighted samples obtained in Step 2. This guarantees that the samples obtained in Step 4 will have the same sample mean and covariance $\bar{s}$ and Σ_s. The smoothing step is related to the local mutation of the genetic algorithms. The mutations are in general independent noise terms and also lead to over-dispersion as discussed before. A common remedy is to a employ a cooling scheme where the variance of the mutations is reduced with the number of iterations to achieve convergence. As we will see in the simulation examples our method does not require such measures for convergence, because the smoothing rule implements the natural reduction in the variance of the population statistics.

4 Examples

4.1 Repeated Prisoner's Dilemma game

Prisoner Dilemma game is a well known discrete space game between two players that illustrates the tradeoff between cooperation and individual benefits. Each player have two strategies Defect (D) and Cooperate (C). The symmetric payoff matrix is given in Fig. 1.

Fig. 1 Classical Prisoner's Dilemma Game

	Defect	Cooperate
Defect	(2,2)	(6,1)
Cooperate	(1, 6)	(4,4)

It is clear that for Player 1, it is beneficial to choose Defect irrespective of the strategy choice of its opponent, in other words Defect is the dominant strategy for Player 1. Since the game is symmetric, Defect is also the dominant strategy for Player 2. Therefore if the players will face each other only once, the likely outcome of the game would be mutual defection, although both players would have benefited from mutual cooperation. If we use evolutionary dynamics to study the evolution of strategies in a population where agents are matched randomly against each other for one shot game, the evolutionary selection forces will favor defectors and the final outcome would not include any cooperators in the population.

However, if the players are locked in repeated games with their opponents, cooperation might evolve. In repeated games, agents can choose complex strategies where current move depends on the entire history of the moves in the game. One can restrict itself to bounded memory models where the agents are restricted to strategies which depend on the opponent's immediate previous move. A well known bounded memory strategy "Tit for Tat" can achieve (Axelrod 1984) cooperation. A player adopting "Tit for Tat" starts with Cooperation in the first round and in subsequent rounds will play the same move played by its opponent in the previous round. Axelrod's computer tournaments "Tit for Tat" performed superior to other iterated strategies and supports the argument for evolution of cooperative behavior in repeated Prisoner's Dilemma game.

In a complex environment the treatment of cooperation by the classical Prisoner's Dilemma game is too simplistic. It is likely that a continuous degree of cooperative behavior (and defection) will be observed and not just the two extremes. Several researchers proposed continuous forms of Prisoner Dilemma game to investigate the evolution of cooperation when cooperation is a continuous trait (Killingback et al. 1999; Wahl and Nowak 1999). These models enable the researchers study the evolution of cooperative behavior in a more realistic setting. Important questions to be answered are whether cooperation can evolve from an initial state of mostly defectors and whether cooperative state is stable starting from such a state. Here we follow a model of continuous Prisoner's Dilemma introduced by Killingback and Doebeli (2002).Killingback and Doebeli first describe the discrete version of the Prisoner's Dilemma game in terms of costs and benefits. Specifically, cooperation involves a cost c to donor and brings a benefit to the recipient. The defection has zero-cost to the donor

Fig. 2 Cost and Benefit parametrization of the Prisoner Dilemma Game

	Defect	Cooperate
Defect	(0,0)	(b,-c)
Cooperate	(-c, b)	(b-c,b-c)

and zero-benefit to the recipient. As a result we arrive the following game matrix given in Fig. 2. Under the assumption that benefits exceed costs, we arrive at the classical result with Defection being a dominant strategy and mutual cooperation is preferred over mutual defection. Then they introduce a continuous valued variable I (investment) measuring the degree of cooperation. Making an investment has a cost defined by $C(I)$ to the donor and brings a benefit to the recipent measured by $B(I)$. Therefore two players choosing investment levels I_1 and I_2 will get a payoff of $\pi(I_1, I_2) = B(I_2) - C(I_1)$ and $\pi(I_2, I_1) = B(I_1) - C(I_2)$ respectively. Killingback and Doebeli considers several functional forms for the functions $B(I)$ and $C(I)$. Here we consider one plausible alternative where $B(I)$ is a concave function of I and $C(I)$ is a linear function of I. The concave shape of the benefit function indicates decreasing returns to scale for the investment. We concentrate on bounded memory strategies where each players' investment level in the current round is only a function of the strategies chosen in the previous round. And following Killingback and Doebeli we further restrict the class of admissible strategies to linear function of the payoff received in the previous round. Specifically if two players adopt strategies (α_1, β_1) and (α_2, β_2) respectively, the investments will follow the following difference equations:

$$I_1^0 = \alpha_1 \qquad I_1^{t+1} = \min\{\alpha_1 + \beta_1[B(I_2^t) - C(I_1^t)], 0\} \tag{11}$$

$$I_2^0 = \alpha_2 \qquad I_2^{t+1} = \min\{\alpha_2 + \beta_2[B(I_1^t) - C(I_2^t)], 0\} \tag{12}$$

Finally to simulate an infinite horizon game, we assume at the end of each round there is a fixed probability ρ of the game finishing at that round. Therefore the games continue on the average $n = 1/(\rho)$ rounds. The payoff to each player in the repeated game is the average payoff earned during the repeated game with stochastic stopping time.

We used the Sequential Monte Carlo technique proposed in this paper as a simulation strategy for the continuous Prisoner Dilemma game. We used $M = 1,000$ particles to approximate the posterior distribution of the strategy profiles. The smoothing parameter a was chosen to be 0.95. The probability of game termination, ρ was set to 0.1, resulting in an average of 10 rounds per game. The simulations were started as initinial condition on the variables α, β highly concentrated distribution of particles with uniformly distributed over the set $[0, 0.02] \times [0, 0.02]$. The benefit and cost functions are given as:

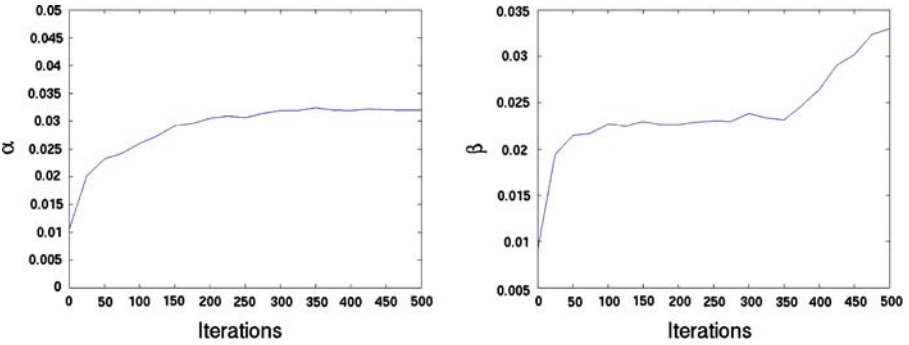

Fig. 3 Evolution of cooperation (α and β parameters)

$$B(I) = 5(1 - \exp\{-I\}) \qquad C(I) = 0.1I \tag{13}$$

The results are given in Figs. 3 and 4. In Fig. 3 we illustrate the evolution in the strategy state, we observe that the population moves slowly from a selfish state with low α, β values to a cooperative stage with high α and moderate β values. Figure 4 depicts the sample mean for α and β as a function of the iteration index. Again we observe cooperation is achieved for a high number of iterations. Finally we note that, at no time does the distribution become singular, even in the final cooperative state we observe small variations within the population.

4.2 Evolution of optimal bidding behavior in auctions

Auctions are traditional mechanisms for resolving multi agent resource allocation problems. The economic theory of auction design and analysis is well established. Klemperer (2002) gives a comprehensive review of the literature on this subject. Four basic types of auctions are widely encountered in real world auctions: the fist-price sealed-bid auction, the second-price sealed-bid auction (Vickrey auction), English auction (open cry ascending bids), the Dutch auction (open cry descending bids). Here we concentrate on evolution of optimal bidding behavior in first price sealed bid auctions. In first-price sealed-bid auction each bidder submits a single bid without seeing the others' bids. The object is sold to the bidder with the highest bid, who pays the exact amount he submitted. The English auction is strategically equivalent to first-price sealed-bid auction, where players use the same bidding functions in both auctions.

An agent's optimal bid depends on the value of the object which is being auctioned. Here we follow the basic private value model, where each bidder knows how much he values the object and this information is private. Here we assume the valution θ_i of each bidder is sampled from a distribution $f_i(\theta)$ that is common knowledge to all bidders. The players are assumed to be maximizing their expected utility defined as $\text{Prob(win)}u_i(b_i(\theta) - \theta)$, where $b_i(\cdot)$ and $u_i(\cdot)$ represents the bidding and utility function for player i, respectively.

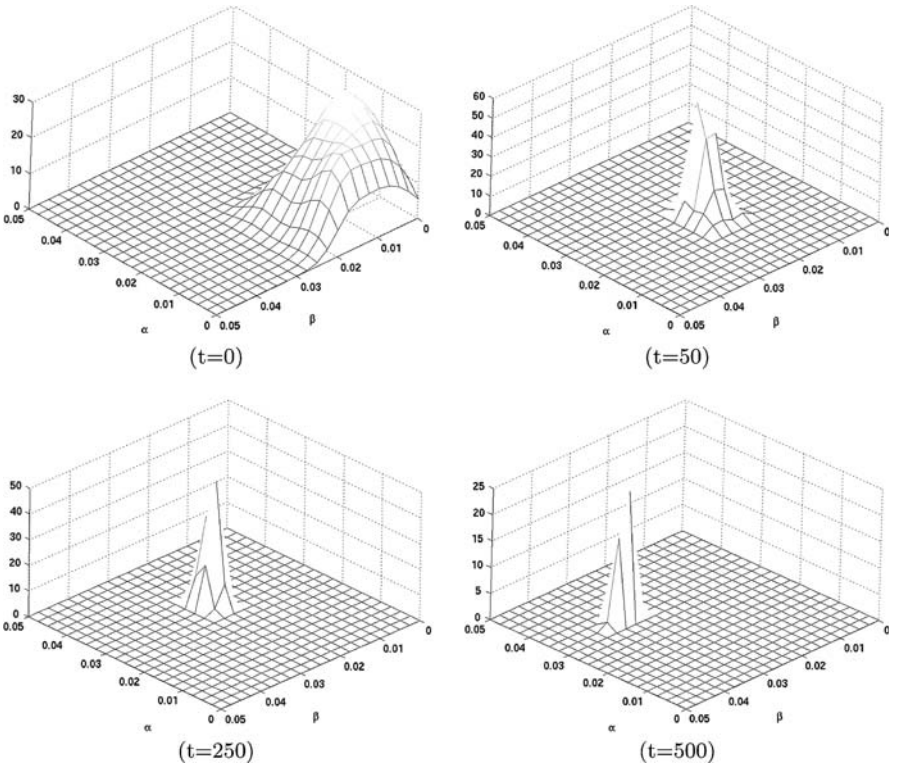

Fig. 4 Population state at different iterations

For simple auctions the optimal bidding functions can be solved analytically by determining the Bayes–Nash equilibrium of the corresponding asymmetric information game. However, analytical solutions can become intractable if model assumptions are relaxed such as asymmetric distributions for private valuations, random number of players, players with non-identical risk attitudes. Here we study the evolution of bidding functions through simulation of replicator dynamics.

To facilitate the simulation, the bidding function $b(\theta)$ is parametrized through a piecewise linear function where d bids $\{b_1, \ldots, b_d\}$ corresponding to regularly spaced values of $\theta \in [1/d, 2/d, \ldots, 1]$ is specified by agent i's strategy. (The range of values for θ has assumed to be restricted to the interval $[0, 1]$ and b_0 is assumed to be zero.) This particular form of bidding function was proposed by Byde (2003) who studied genetic programming for auction mechanism design. The resulting piecewise linear function can be expressed as:

$$b_i(\theta) = \begin{cases} 0 & \theta < 0 \\ b_k + (\theta - k/d)(b_{k+1} - b_k) & \theta \in [k/d, k + 1/d] \\ b_d & \theta > 1 \end{cases} \tag{14}$$

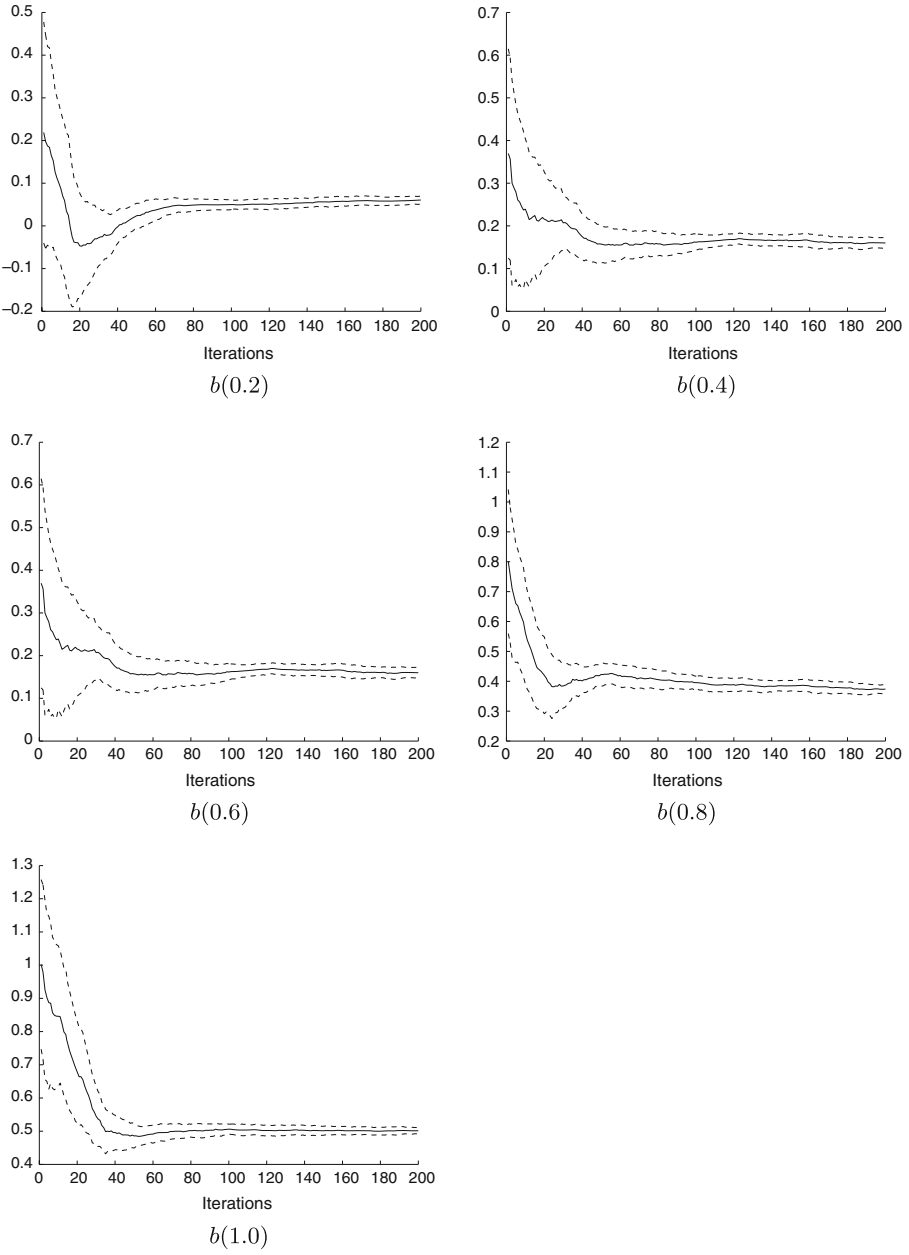

Fig. 5 Evolution of the bidding behavior in closed-bid first-price auctions. The population mean for the piecewise linear bidding function $b(\theta)$ is shown for $\theta \in \{0.2, 0.4, \ldots, 1.0\}$ (*solid*). The population standard deviation around the mean is also depicted (*dashed*)

We have chosen $d = 5$ which leads to a five dimensional continuous valued game between two agents. The valuations are drawn independently from a uniform distribution on the interval $[0, 1]$. The agents are assumed to be risk

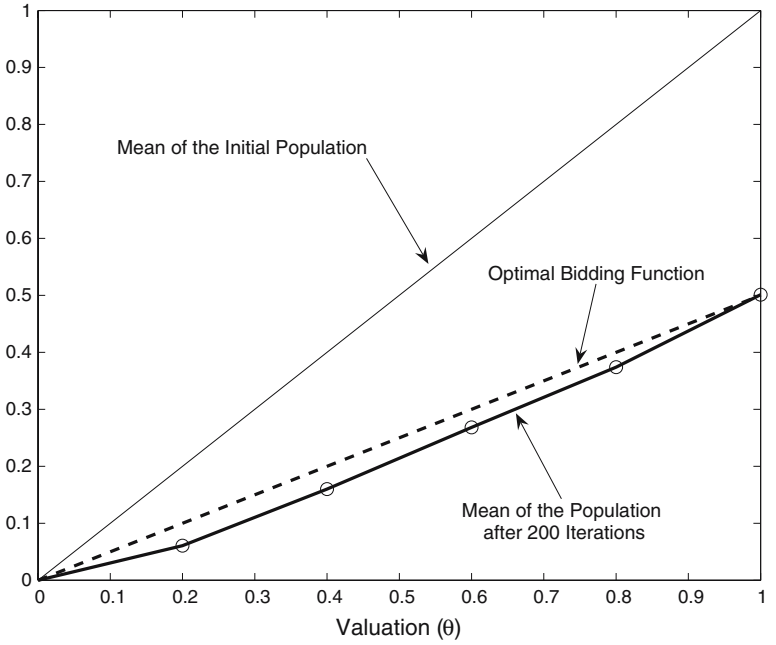

Fig. 6 Final bidding function for the mean population

neutral maximizing expected profits. Under this assumption the Bayes Nash equilibrium bidding function is to bid half of your valuation. The replicator dynamics is simulated using $M = 250$ particles. The smoothing parameter a was chosen to be 0.95.[1] The initial population has the mean bidding function $b(\theta) = \theta$ and gaussian distributed with standard deviation of 0.25 in each of the five dimensions. Fig. 5 shows the evolution of the bidding function parameters $b_1, \ldots, b_d$ through time, the solid line depicts the mean of the population whereas the dashed line shows plus-minus standard deviation in the population in that dimension. The mean bidding function of the population after 200 iterations is shown in Fig. 6 with the population standard deviation. We observe that the convergence point of the replicator dynamics is in agreement with the optimal bidding function.

5 Conclusion

We presented a SMC method for simulation of evolutionary dynamics on multi-dimensional continuous valued strategy spaces. The technique is illustrated

[1] For both examples, we have experimented with several values of the smoothing parameter. Low values of the smoothing parameter ($a < 0.9$) results in simulation of a modified dynamics because of the heavy shrinkage to the mean. High values of in the smoothing parameter ($a > 0.99$) results in trajectories that exhibit convergence with discrete jump steps and sensitivity to the initial set of particles.

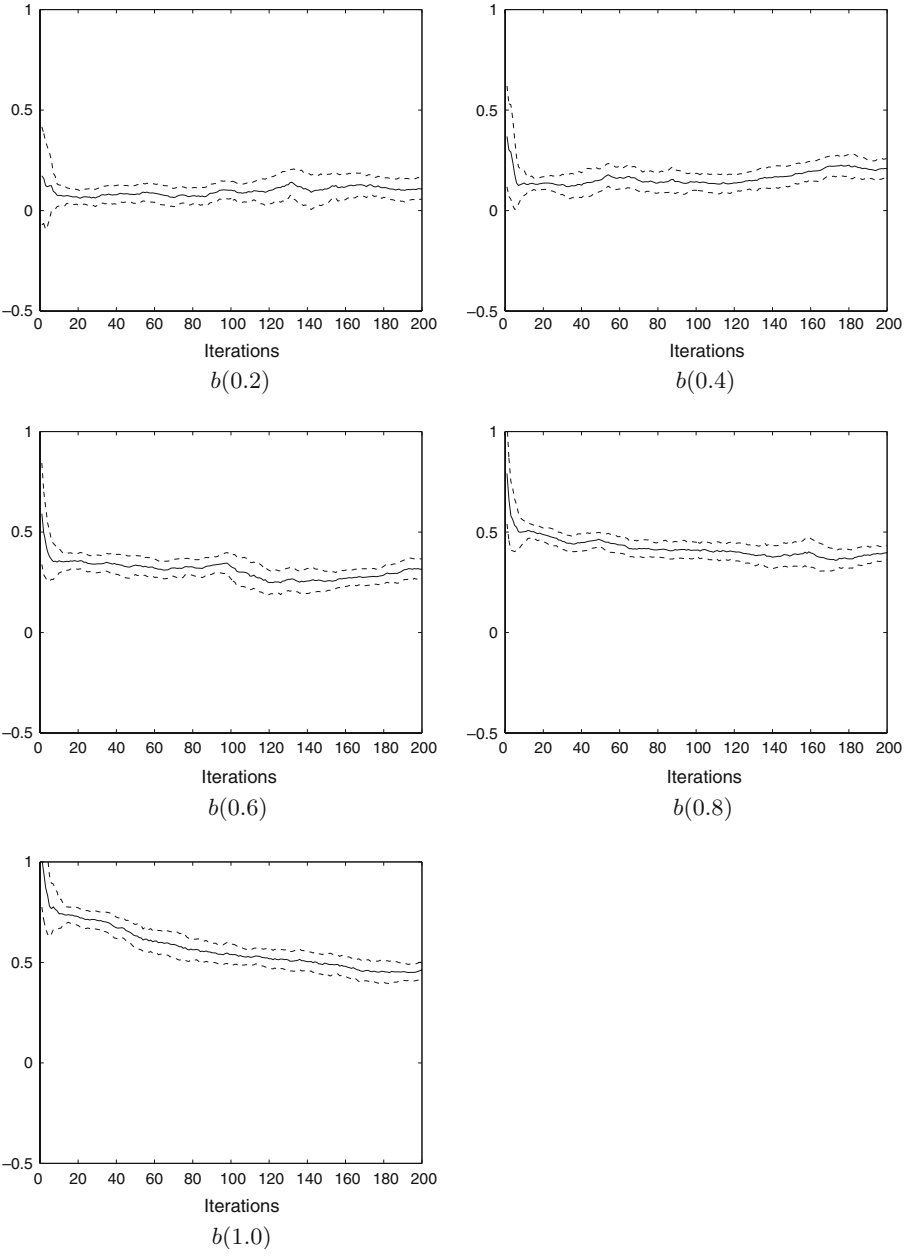

Fig. 7 Evolution of the bidding behavior with artificial mutations. The population mean (*solid*) and standard deviation (*dashed*) for the piecewise linear bidding function $b(\theta)$ is shown

on two examples: a continuous version of the repeated Prisoner Dilemma game, where cooperative behavior emerges from an initial society with selfish parameters and evolution of optimal bidding behavior in closed bid auctions.

Previously, Riechmann (2001) has shown that economic learning with a genetic algorithm can be seen as a specific form of an evolutionary game. Here we note that the SMC algorithm presented here for simulation of evolutionary dynamics is closely related to the elements of an agent based genetic algorithm. Calculation of the weights corresponds to tournament based calculation of the fitness function. Resampling with replacement step correspond to roulette wheel selection for parents. Finally, smoothing step corresponds to local mutations in the Genetic Algorithm preceded by a regress to mean step. The proposed algorithm solves the over-dispersion problem associated with random mutations by accurate implementation of the natural reduction in variance of the population statistics. And therefore it does not require artificial methods such as cooling schedules for convergence. We also note that the cross-over operation, an important ingredient of genetic algorithms is absent in the sequential Monte-Carlo simulation of evolutionary dynamics.

Acknowledgments We thank the anonymous referee and Sheri Markose for their detailed comments on the paper.

Appendix

Over-dispersion caused by artificial mutations

The resampling step of the SMC method solves the sample degeneration problems in importance sampling, where after a few iteration only few samples with large weights are present in the distribution estimate. In a dynamics model without explicit sources of variance (e.g. noise, mutations) a significant number of the particles will be identical after the resampling step. The standard approach to introduce variations in the particle set is to add small random perturbations (artificial mutations) to each sample. A straightforward implementation of the smoothing step uses independent random perturbations which increases the variance of the particle set resulting in a posterior distribution that is *over-dispersed* as compared to the variance of the weighted samples at the end of the selection step. Analogously, introducing independent random perturbations leads to simulation of a modified dynamics:

$$\mu^{t+1}(s) = \alpha \int \mu^t(s') \left[1 + \Delta \left(E(s', \mu^t) - E(\mu^t, \mu^t)\right)\right] k(s - s') \mathrm{d}s' \qquad (15)$$

where $k(\cdot)$ is the probability distribution for the random perturbations. The convolution with $k(\cdot)$ is the source of the over-dispersion problem in μ^{t+1}.

To illustrate the over-dispersion problem caused by artificial mutations we have simulated the modified dynamics given in Eq. (15) for the example in Sect. 4.2. Specifically, for the smoothing step we use independent perturbations chosen from a normal distribution with zero mean and diagonal covariance matrix $\sigma^2 I$, where $\sigma = 0.05$. Figure 7, shows the evolution of the bidding function parameters $b(0.2), \ldots, b(1.0)$ for the modified dynamics given in (15), the

solid line depicts the mean of the population whereas the dashed line shows plus–minus standard deviation in the population in that dimension. Comparing with Fig. 5 we observe that the variation in population does not diminish over time and in general larger than the variance of the introduced mutations. The limit point of the convergence is also different than $b(\theta) = \theta/2$, since the optimal bidding behavior should account for the variance of the bidding functions observed in the population.

Heuristic methods such as a cooling scheme where the variance of the mutations is reduced with the number of iterations can be adopted to achieve convergence. This will require finding out an appropriate cooling scheme for each particular problem. To overcome the over-dispersion problem we adopted a shrinkage rule suggested by Liu and West (2001) which essentially introduces correlated perturbations which preserves mean and variance of the population distribution. Examples 4.1 and 4.2 shows that our method does not require heuristics for convergence, because the smoothing rule follows the natural reduction in the variance of the population statistics.

References

Axelrod RD (1984) The evolution of cooperation. Basic Books, New York
Benaim M, Weibull J (2003) Deterministic approximation of stochastic evolution in games. Econometrica 71:873–903
Bomze I (1990) Dynamical aspects of evolutionary stability. Monatshefte Math 110:189–206
Boylan RT (1992) Laws of large numbers for dynamical systems with randomly matched agents. J Econ Theory 57:473–504
Byde A (2003) Applying evolutionary game theory to auction mechanism design. In: Proceedings of the IEEE international converence on E-commerce
Carpenter JP (2002) Evolutionary models of bargaining: comparing agent-based computational and analytical approaches to understanding convention evolution. Comput Econ 19:25–49
Corradi V, Sarin R (2000) Continuous approximation of stochastic evolutionary game dynamics. J Econ Theory 94:163–191
Dawid H (1999) Adaptive learning with genetic algorithms, analytical results and applications to economic models. Springer, Berlin Heidelberg New York
Doucet A, Freitas N, Gordon N (eds) (2001) Sequential Monte Carlo methods in practice. Springer, Berlin Heidelberg New York
Friedman D, Yellin J (1997) Evolving landscapes for population games. Draft Manuscript University of California, Sant Cruz
Fudenberg D, Levine DK (1998) The theory of learning in games. MIT Press, Cambridge
Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-gaussian bayesian state estimation. IEE Proceedings-F 140, 2
Killingback T, Doebeli M (2002) The continuous prisoner's dilemma and the evolution of cooperation through reciprocal altruism with variable investment. Am Nat 160:4
Killingback T, Doebeli M, Knowlton N (1999) Variable investment, the continuous prisoner's dilemma, and the origin of cooperation. In: Proceedings of the Royal Society of London B 263
Klemperer P (2002) Auctions: theory and practice. Princeton University Press, Englewood Cliffs
Liu J, West M (2001) Combined parameter and state estimation in simulation-based filtering. In: Doucet A, Freitas N, Gordon N (eds) Sequential Monte Carlo methods in practice, Springer, Berlin Heidelberg New York
Oechssler J, Riedel F (2001) Evolutionary dynamics on infinite strategy spaces. Econ Theory 17:1
Riechmann T (2001) Genetic algorithm learning and evolutionary games. J Econ Dyn Control 25:1019–1037

Schenk-Hoppe, KR (2000) The evolution of walrasian behavior in oligopolies. J Math Econ 33:35–55
Vega-Redondo F (1996) Evolution, games and economic behaviour. Oxford University Press, New York
Vega-Redondo F (1997) The evolution of walrasian behavior. Econometrica 65:2
Wahl LM, Nowak MA (1999) The continuous prisoner's dilemma: I. linear reactive strategies. J Theor Biol 200:323–338