The computational complexity of ecological and evolutionary spatial dynamics

Rasmus Ibsen-Jensen [*], **Krishnendu Chatterjee** [*], and **Martin A. Nowak** [†]

[*] IST Austria, Klosterneuburg, A-3400, Austria, and [†] Program for Evolutionary Dynamics, Department of Organismic and Evolutionary Biology, Department of Mathematics, Harvard University, Cambridge, MA 02138, USA

Submitted to Proceedings of the National Academy of Sciences of the United States of America

There are deep, yet largely unexplored connections between computer science and biology. Both disciplines examine how information proliferates in time and space. Central results in computer science describe the complexity of algorithms that solve certain classes of problems. An algorithm is deemed efficient if it can solve a problem in polynomial time, which means the running time of the algorithm is a polynomial function of the length of the input. There are classes of harder problems for which the fastest possible algorithm requires exponential time. Another criterion is the space requirement of the algorithm. There is a crucial distinction between algorithms that can find a solution, verify a solution, or list several distinct solutions in given time and space. The complexity hierarchy that is generated in this way is the foundation of theoretical computer science. Precise complexity results can be notoriously difficult. The famous P=NP question is one of the hardest open problems in computer science and all of mathematics. Here we consider simple processes of ecological and evolutionary spatial dynamics. The basic question is: what is the probability that a new invader (or a new mutant) takes over a resident population? We derive precise complexity results for a variety of scenarios. We therefore show that some fundamental questions in this area cannot be answered by simple equations.

Significance

There is a deep connection between computer science and biology, as both fields study how information proliferates in time and space. In computer science, the space and time requirements of algorithms to solve certain problems generate complexity classes, which represent the foundation of theoretical computer science. The theory of evolution in structured population has provided an impressive range of results, but an understanding of the computational complexity of even simple questions is still missing. In this work we prove – unexpectedly – that some fundamental problems in ecological and evolutionary spatial dynamics can be precisely characterized by well-established complexity classes of the theory of computation. Since we show computational hardness for several basic problems, our results imply that the corresponding questions cannot be answered by simple equations. For example, there cannot be a simple formula for the fixation probability of a new mutant given frequency dependent selection in a structured population. We also show that some problems, such as calculating the molecular clock of neutral evolution in structured populations, admit efficient algorithmic solutions.

Evolutionary games on graphs | Fixation probability | Complexity classes

Evolution occurs in populations of reproducing individuals. Mutation generates distinct types. Selection favors some types over others. The mathematical formalism of evolution describes how populations change in their genetic (or phenotypic) composition over time. Many papers study evolutionary dynamics in structured populations [1, 2, 3, 4, 5, 6, 7, 8]. Spatial structure can affect the rate of neutral evolution [9]. There are results that describe which spatial structures do or do not affect the outcome of constant selection [10, 11, 12]. Constant selection refers to a situation where the competing types have constant reproductive rates independent of the composition of the population. Some population structures can be amplifiers or suppressors of constant selection [13, 6, 14] meaning that they modify the intensity of selective differences. A large literature deals with evolutionary games [15, 16, 17, 18, 19] in structured populations [1, 20, 21, 22, 23, 24, 25, 26, 27, 28]. In evolutionary games the reproductive success of an individual depends on interactions with others. Many population structures and update rules can affect the outcome of evolutionary games. For example, spatial structure can favor evolution of cooperation [1, 29].

In this paper we are interested in stochastic evolutionary dynamics in populations of finite size. A typical setting is provided by evolutionary graph theory [6, 30, 31, 32, 33, 34]. The individuals of a population occupy the vertices of a graph. The links determine who interacts with whom for receiving payoff and for reproduction. There can be a single graph for game dynamical interaction and evolutionary replacement, or the interaction and replacement graphs can be distinct [35]. Often the graph is held constant during evolutionary updating, but it is also possible to study dynamically changing graphs [36, 37, 38, 39, 40, 41, 42, 43, 44].

The study of spatial dynamics also has a long tradition in ecology [45, 46, 47, 48, 49]. Here the typical setting is that different species compete for ecological niches. Many evolutionary models are formally equivalent to ecological ones - especially if we consider only selection and not mutation. Then we can interpret the different types as individuals of different species.

This paper is structured as follows. First we give an intuitive account of the foundation of theoretical computer science. We describe classes of problems that can be solved by algorithms in certain time and space constraints. Subsequently we present two simple problems of evolutionary dynamics in spatial settings. The first problem is motivated by a very simple ecological dynamic; the second problem is the general setting of evolutionary games on graphs. In both cases, the basic question is to calculate the take over probability (or fixation probability) of a new type. That is we introduce a new type in a random position in the population and we ask what is the complexity of an algorithm that can characterize the probability that the new type takes over the population (becomes fixed). Unexpectedly we are able to prove exact complexity results (see Table 1).

Reserved for Publication Footnotes

The class PTIME (denoted as P) consists of problems whose solutions can be computed by an algorithm that uses polynomial time. This means the running time of the algorithm grows as a polynomial function of the size of the input. In computer science, PTIME represents the class of problems which can be solved efficiently.

The class NP (non-deterministic polynomial time) consists of problems, for which solutions exist that are of polynomial length, and given a candidate for a solution of polynomial length, whether the candidate is indeed a solution can be checked in polynomial time. Therefore, an NP algorithm can verify a solution in polynomial time.

In order to proceed further, we need the notion of 'reduction' between classes of problems. A reduction, from a given problem P_1 to a problem P_2, is a translation such that a solution for P_2 can provide a solution for P_1. More precisely, if there is a polynomial-time reduction from P_1 to P_2, then a polynomial-time algorithm for P_2 implies a polynomial-time algorithm for P_1.

A given problem is NP-hard if for every problem in NP there is a polynomial reduction to the given problem. A problem is NP-complete, if it is both NP-hard, as well as there is an NP algorithm for the problem.

For example, consider a Boolean formula over variables, and the question whether there exists an assignment to the variables such that the formula is true. A polynomial candidate solution is an assignment of truth values to variables, and given a candidate assignment the formula can be evaluated in polynomial time. This is the famous satisfiability, SAT, problem in computer science. The SAT problem is NP-complete.

The class P is contained in NP, and a major, long-standing open question in computer science is whether P=NP? A polynomial-time algorithm for an NP-complete (or an NP-hard) problem would imply that P=NP, resolving the long-standing open problem.

The class #P (sharpP) intuitively corresponds to counting the number of solutions. A problem is in #P if it counts the number of distinct solutions such that (i) every possible candidate for a solution is of polynomial length, and (ii) given a candidate for a solution, it can be checked in polynomial time whether the candidate is a solution. For example, given a Boolean formula, the problem whether there are at least k distinct satisfying assignments to the formula is a #P-problem. A given problem is #P-hard, if for every #P-problem there is a polynomial-time reduction to the given problem. A #P-complete problem is a problem that is both #P-hard, and there is a #P-solution. For example, counting the number of solutions in SAT is #P-complete.

The class NP is contained in #P because given the enumeration of solutions for #P, it is easy to check if there exists at least one solution. Intuitively, an NP problem asks whether there is at least one solution, whereas #P is the counting version which asks if there are least k distinct solutions (and the special case of $k = 1$ gives NP). Again a major open question is whether NP=#P? Note that a polynomial-time algorithm for a #P-complete problem would be an even bigger result as it would imply both P=NP and P=#P.

The class PSPACE consists of problems which can be solved with polynomial space. Note that a polynomial space algorithm can reuse space and can in general require exponential time. Every #P problem can be solved in PSPACE by simply enumerating each candidate for a solution and checking if it is a solution. Since we can reuse space to enumerate the candidates for solutions, the enumeration can be achieved in polynomial space. Moreover, every polynomial-time algorithm uses at most polynomial space. Hence it follows that #P is contained in PSPACE. The notion of PSPACE-hardness and PSPACE-completeness is similar to the notion of NP-hardness and NP-completeness, but with respect to the problems in PSPACE. Again a long-standing open question in computer science is whether #P=PSPACE, and a polynomial-time algorithm for a PSPACE-complete (or PSPACE-hard) problem would imply P=NP=#P=PSPACE.

We have mentioned that the major questions about the equality of the complexity classes are open problems, but the widely believed conjecture is that P is strictly contained in NP, NP is strictly contained in #P, and #P is strictly contained in PSPACE. In other words, it is widely believed that NP-complete problems cannot be solved in polynomial time, #P-complete problems are harder than NP-complete problems, and PSPACE-complete problems are harder than #P-complete problems. A pictorial illustration of the complexity classes is shown in Figure 1.

The first problem is motivated by ecological dynamics. There is an ecosystem occupied by resident species. The spatial structure of the ecosystem is given by a graph. An invading species is introduced (see Figure 2 for an illustration). We assume the invading species has a competitive advantage in the sense that once a position is occupied by the invading species the resident cannot get it back. The invading species, however, has a density constraint: if the number of invaders around a focal invader is above a threshold, h, then the invader in the focal node can not colonize another node.

We are interested in the probability that the invader starting from a random initial position will take over the entire ecosystem (and therefore drive the resident to extinction). There are two types of questions. The 'qualitative question' is whether the take over probability is greater than zero. The 'quantitative question' is concerned with computing the take over probability subject to a small error. Figure 2 gives a pictorial illustration. We prove the following results. The qualitative question is NP-complete (SI Theorem 4). The quantitative question is #P-complete (SI Theorem 8).

The second problem is concerned with evolutionary games in structured populations. There are two types, A and B, whose reproductive rates depend on local interactions. We consider the setting of games on graphs. Each vertex is occupied by one individual, which is either A or B. Interactions occur pairwise with all neighbors. The payoff matrix is given by

$$\begin{array}{cc} & \begin{array}{cc} A & B \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array} \qquad (1)$$

The entries of the payoff matrix can be positive or negative (or zero). Each individual interacts with all of its neighbors on the graph to derive a payoff sum. The payoff sum is translated into reproductive success as follows. If the payoff sum is positive, then the fecundity equals the payoff sum. If the payoff sum is negative, then the fecundity is zero. We refer to this translation as linear fitness. In any one time step, a random individual is chosen for reproduction proportional to its fecundity. The offspring, which is of the same type as the parent, is placed into an adjacent position on the graph (see Figure 3 and Figure 4 for an illustration).

We are interested in the probability that a single A individual starting in a random position on the graph generates a lineage which will take over the entire population; this probability is generally called fixation probability. As before, there are two types of questions. The 'qualitative question' is whether the fixation probability is positive. The 'quantitative question' is concerned with computing the fixation probability subject to a small error. We prove the following results. The qualitative question is NP-hard and in PSPACE. The quantitive question is #P-hard and in PSPACE. The results follow from SI Theorem 4, Theorem 8, and Theorem 15.

Note that the first problem can also be obtained as a special case of the second problem. In the payoff matrix (1) we can set, for example, $a = -1$, $b = 1$, $c = d = 0$. This 'game' has the property that type B never reproduces and type A reproduces until half its neighbors are also of type A. This parameter choice leads to the same qualitative behavior and the same complexity bounds as described in the first problem.

A generalization of games on graphs is the setting where the interaction graph and the replacement graph are distinct [35]. Thus each individual interacts with all of its neighbors on the interaction graph to receive payoff. Subsequently an individual is chosen for reproduction proportional to its fecundity. The offspring is placed randomly among all neighbors of the focal individual on the replacement graph. In this case, both the qualitative and quantitative questions become PSPACE-complete (SI Theorem 15)

We also consider a variation of the second problem. In particular we change the mapping from payoff to fecundity. We now assume that fecundity is an exponential function of payoff, and refer to it as exponential fitness (see Figure 4 for an illustration). Therefore the fecundity of an individual is always positive (even if its payoff sum is negative). In this setting the qualitative question can be decided in polynomial time. The reason is that the fixation probability is positive if the graph is connected. Thus, in order to answer the qualitative question the algorithm only needs to check whether the graph is connected; this problem is in P. However, the quantitative question has the same complexity as the previous problem (SI Theorem 16 and Theorem 17).

A special case of games on graphs is constant selection. Type A has constant fecundity a and type B has constant fecundity b independent of any interactions. The qualitative question concerning the fixation probability of A is in P. The quantitative question is in PSPACE, but any non-trivial lower bound is an open question.

In summary, we have established complexity results for some fundamental problems in ecology and evolutionary games on graphs. In particular, we have solved the open problems mentioned in the survey [50, Open Problem 2.1 and 2.2]. Our main results are summarized in Table 1. The most interesting aspects of our results are the lower bounds, which shows that in most cases there exists no efficient algorithm, under the widely believed conjecture that P is different from NP. A simple equation based solution would give an efficient algorithm, and thus our result shows that for evaluating the fixation probability in spatial settings there does not exist a simple equation based solution in general.

Finally, while we establish computational hardness for several problems, we also show that two classic problems can be solved in polynomial time (SI Section 7). First, we consider the molecular clock, which is the rate at which neutral mutations accumulate over time. The molecular clock is affected by population structure [35]. We show that the molecular clock can be computed in polynomial time because the problem reduces to solving a set of linear equalities, which can be achieved in polynomial time using Gaussian elimination. Second, we consider evolutionary games in a well-mixed population structure, where the underlying structure is the complete graph [51]. We show that the exact fixation probability can be computed in polynomial time. In this case the problem can be reduced to computing absorption probabilities in Markov chains, where each state represent the number of mutants. Hence the Markov chain is linear in the number of vertices of the graphs, and since absorption probabilities in Markov chains can be computed in polynomial time (by solving a set of linear equalities) we obtain the desired result.

1. Nowak MA, May RM (1992) Evolutionary games and spatial chaos. Nature 359:826.
2. Killingback T, Doebeli M (1996) Spatial evolutionary game theory: Hawks and doves revisited. Proc. R. Soc. B 263(1374):1135–1144.
3. Szabó G, Toke C (1998) Evolutionary prisoner's dilemma game on a square lattice. Phys. Rev. E. 58(1):69–73.
4. Szabó G, Hauert C (2002) Phase transitions and volunteering in spatial public goods games. Phys. Rev. Lett. 89:118101.
5. Hauert C, Doebeli M (2004) Spatial structure often inhibits the evolution of cooperation in the snowdrift game. Nature 428:643.
6. Lieberman E, Hauert C, Nowak MA (2005) Evolutionary dynamics on graphs. Nature 433(7023):312–316.
7. Nowak MA, Tarnita CE, Antal T (2009) Evolutionary dynamics in structured populations. Phil Trans R Soc B 365(1537):19–30.
8. Allen B, Tarnita CE (2012) Measures of success in a class of evolutionary models with fixed population size and structure. J. Math. Biol pp. 1–35.
9. Allen B et al. (2015) The molecular clock of neutral evolution can be accelerated or slowed by asymmetric spatial structure. PLoS Comput Biol 11(2):e1004108+.
10. Adlam B, Nowak MA (2014) Universality of fixation probabilities in randomly structured populations. Sci Rep 4:6692.
11. Maruyama T (1974) A markov process of gene frequency change in a geographically structured population. Genetics 76(2):367–377.
12. Barton NH (1993) The probability of fixation of a favoured allele in a subdivided population. Genetics Research 62:149–157.
13. Nowak MA, Michor F, Iwasa Y (2003) The linear process of somatic evolution. Proc Natl Acad Sci USA 100(25):14966–14969.
14. Nowak MA (2006) Evolutionary Dynamics. (Harvard University Press).
15. Smith JM (1982) Evolution and the Theory of Games. (Cambridge University Press, Cambridge, UK).
16. Hofbauer J, Sigmund K, eds. (1988) The theory of evolution and dynamical systems: Mathematical aspects of selection. (Cambridge University Press, Cambridge).
17. Hofbauer J, Sigmund K (1998) Evolutionary Games and Population Dynamics. (Cambridge University Press, Cambridge).
18. Cressman R (2003) Evolutionary dynamics and extensive form games. Economic learning and social evolution. (MIT Press, Cambridge (Mass.)).
19. Broom M, Rychtář J (2013) Game-Theoretical Models in Biology. (Chapman and Hall/CRC Math. Comput. Biol. Ser.).
20. Ellison G (1993) Learning, local interaction, and coordination. Econometrica 61(5):1047–1071.
21. Herz AV (1994) Collective phenomena in spatially extended evolutionary games. J. Theor. Biol. 169:65 – 87.
22. Nakamaru M, Nogami H, Iwasa Y (1998) Score-dependent fertility model for the evolution of cooperation in a lattice. J. Theor. Biol. 194:101 – 124.
23. Szabó G, Antal T, Szabó P, Droz M (2000) Spatial evolutionary prisoner's dilemma game with three strategies and external constraints. Phys. Rev. E 62:1095+.
24. Kerr B, Riley MA, Feldman MW, Bohannan BJM (2002) Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. Nature 418:171–174.
25. Helbing D, Yu W (2008) Migration as a Mechanism to Promote Cooperation. Advances in Complex Systems 11:641–652.
26. Tarnita CE, Ohtsuki H, Antal T, Fu F, Nowak MA (2009) Strategy selection in structured populations. J. Theor. Biol. 259:570 – 581.
27. Perc M, Szolnoki A (2010) Coevolutionary gamesa mini review. Biosystems 99:109 – 125.
28. van Veelen M, García J, Rand DG, Nowak MA (2012) Direct reciprocity in structured populations. P. Natl. Acad. Sci. USA 109:9929–9934.
29. Ohtsuki H, Hauert C, Lieberman E, Nowak MA (2006) A simple rule for the evolution of cooperation on graphs and social networks. Nature 441(7092):502–505.
30. Szabó G, Fáth G (2007) Evolutionary games on graphs. Phys. Rep. 446:97–216.
31. Yang HX, Wu ZX, Du WB (2012) Evolutionary games on scale-free networks with tunable degree distribution. EPL (Europhysics Letters) 99(1):10006.
32. Chen YT (2013) Sharp benefit-to-cost rules for the evolution of cooperation on regular graphs. Ann. Appl. Probab. 23:637–664.
33. Allen B, Nowak MA (2014) Games on graphs. EMS Surv Math Sci 1:113–151.
34. Débarre F, Hauert C, Doebeli M (2014) Social evolution in structured populations. Nat Commun 5.
35. Ohtsuki H, Pacheco JM, Nowak MA (2007) Evolutionary graph theory: breaking the symmetry between interaction and replacement. J. Theor. Biol. 246:681–694.
36. Skyrms B, Pemantle R (2000) A dynamic model of social network formation. P. Natl. Acad. Sci. USA 97(16):9340–9346.
37. Pacheco JM, Traulsen A, Nowak MA (2006) Coevolution of strategy and structure in complex networks with dynamical linking. Phys. Rev. Lett. 97:258103.
38. Fu F, Hauert C, Nowak MA, Wang L (2008) Reputation-based partner choice promotes cooperation in social networks. Phys. Rev. E 78:026117.
39. Antal T, Ohtsuki H, Wakeley J, Taylor PD, Nowak MA (2009) Evolution of cooperation by phenotypic similarity. P. Natl. Acad. Sci. USA 106(21):8597–8600.
40. Tarnita CE, Antal T, Ohtsuki H, Nowak MA (2009) Evolutionary dynamics in set structured populations. P. Natl. Acad. Sci. USA 106(21):8601–8604.

41. Szolnoki A, Perc M (2009) Resolving social dilemmas on evolving random networks. EPL (Europhysics Letters) 86(3):30007.
42. Cavaliere M, Sedwards S, Tarnita CE, Nowak MA, Csikász-Nagy A (2012) Prosperity is associated with instability in dynamical networks. Journal of Theoretical Biology 299(0):126 – 138.
43. Rand DG, Arbesman S, Christakis NA (2011) Dynamic social networks promote cooperation in experiments with humans. P. Natl. Acad. Sci. USA 108(48):19193–19198.
44. Wu B et al. (2010) Evolution of cooperation on stochastic dynamical networks. PLoS ONE 5(6):e11187.
45. Durrett R, Levin S (1994) The importance of being discrete (and spatial). Theor. Popul. Biol 46(3):363 – 394.
46. Levin SA, Paine RT (1974) Disturbance, patch formation, and community structure. P. Natl. Acad. Sci. USA 71(7):2744–2747.
47. Hassell MP, Comins HN, May RM (1994) Species coexistence and self-organizing spatial dynamics. Nature 370(6487):290–292.
48. Tilman D, Kareiva PM (1997) Spatial ecology. The role of space in population dynamics and interspecific interactions. (Princeton University Press).
49. Dieckmann U, Law R, Metz JAJ, eds. (2000) The Geometry of Ecological Interactions. (Cambridge University Press).
50. Shakarian P, Roos P, Johnson A (2012) A review of evolutionary graph theory with applications to game theory. Biosystems 107(2):66 – 80.
51. Nowak MA, Sasaki A, Taylor C, Fudenberg D (2004) Emergence of cooperation and evolutionary stability in finite populations. Nature 428(6983):646–650.

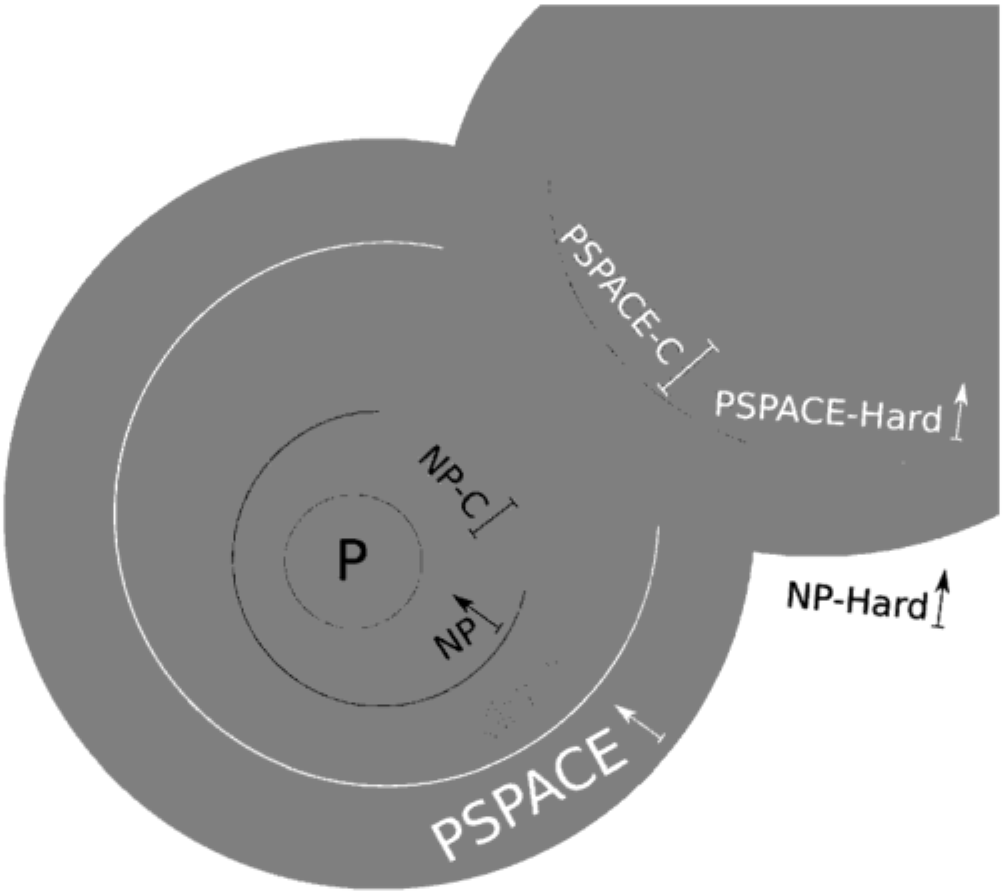

Fig. 1. A pictorial illustration of the complexity classes P, NP, #P, and PSPAPCE. The complexity class P is contained in NP, NP is contained in #P, and #P is contained in PSPACE. The widely believed conjecture is that the complexity classes are different. A problem is NP-hard if it is at least as hard as each problem in NP; and similar for #P-hardness and PSPACE-hardness. The intersection of NP and NP-hard gives the NP-complete problems, and similarly for #P-complete and PSPACE-complete problems. Hence a polynomial-time solution for a NP-hard or NP-complete problem would imply P=NP.

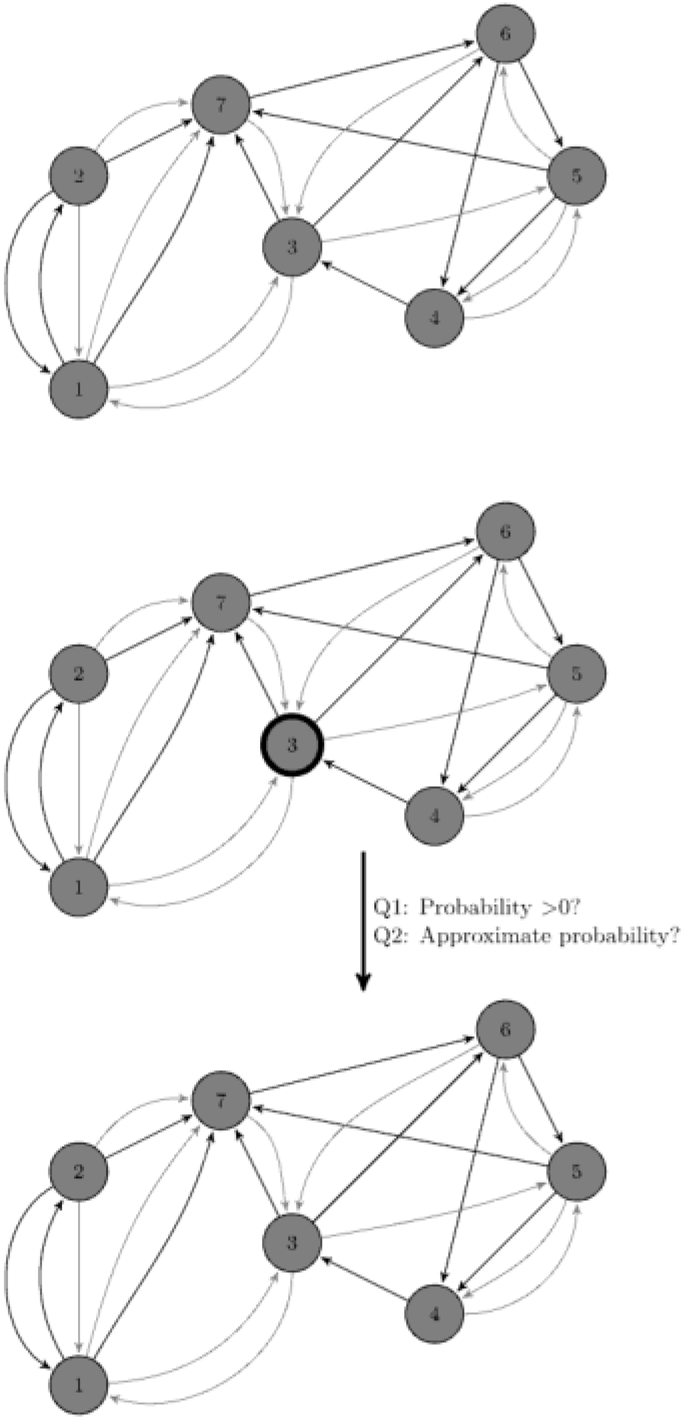

Fig. 2. Illustration of mutant introduction. The residents (type A) are colored blue and the mutants (type B) are colored red. The black edges are the edges of the interaction graph and the red are the edges of the reproduction graph. The probability to introduce a mutant in a specific vertex is always one over the number of vertices. The computational questions of interest regarding the take over probability are as follows: whether the probability is positive (qualitative question), and compute an approximation of the probability (quantitative question).

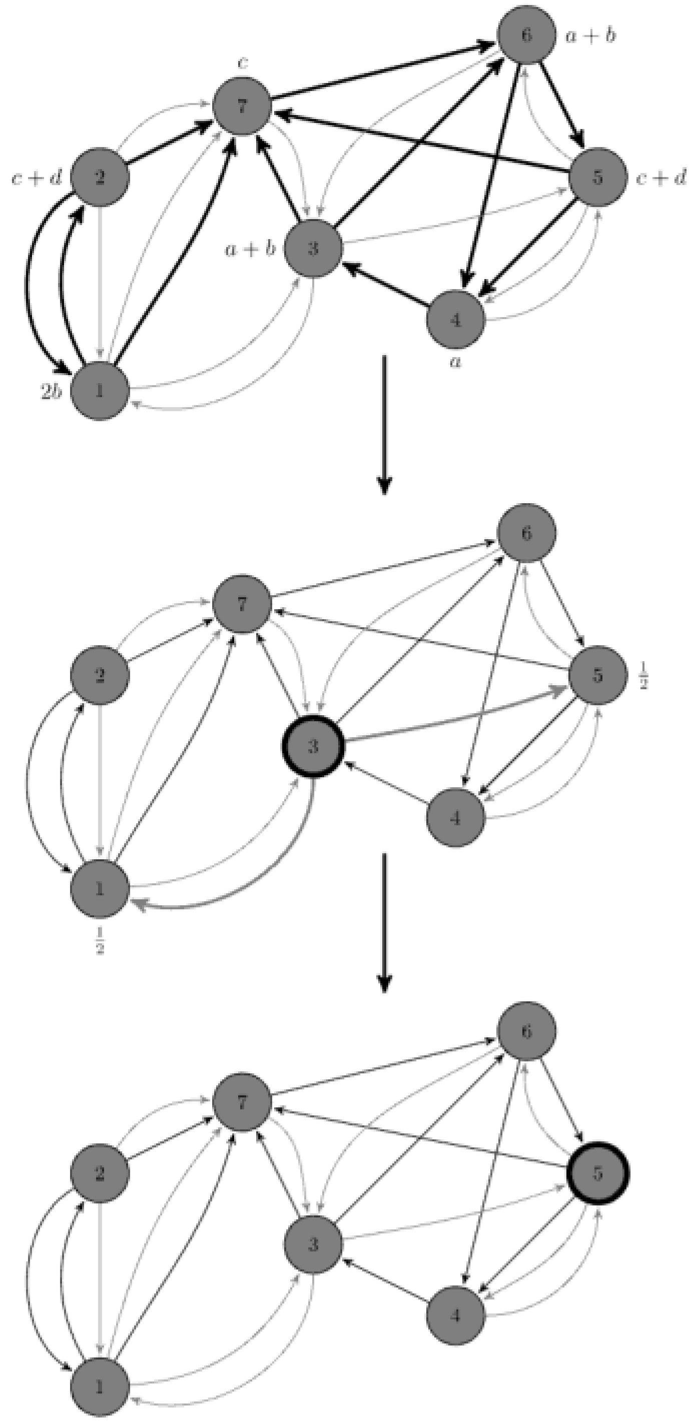

Fig. 3. Illustration of reproduction with matrix $\begin{array}{cc} & \begin{array}{cc} A & B \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array}$. The residents (type A) are colored blue and the mutants (type B) are colored red. The black edges are the edges of the interaction graph and the red are the edges of the reproduction graph. In the first figure beside each vertex the payoff of the vertex (which is the sum of the payoff of the interactions) is shown. Since the first figure shows the payoff computation, the interaction edges that are responsible for payoff calculation are boldfaced. In the second figure the vertex labeled 3 is selected for reproduction. The reproduction edges from vertex 3 are boldfaced, and each edge has probability 1/2. Finally, the successor 5 is chosen for replacement, i.e., vertex 3 reproduces to vertex 5.

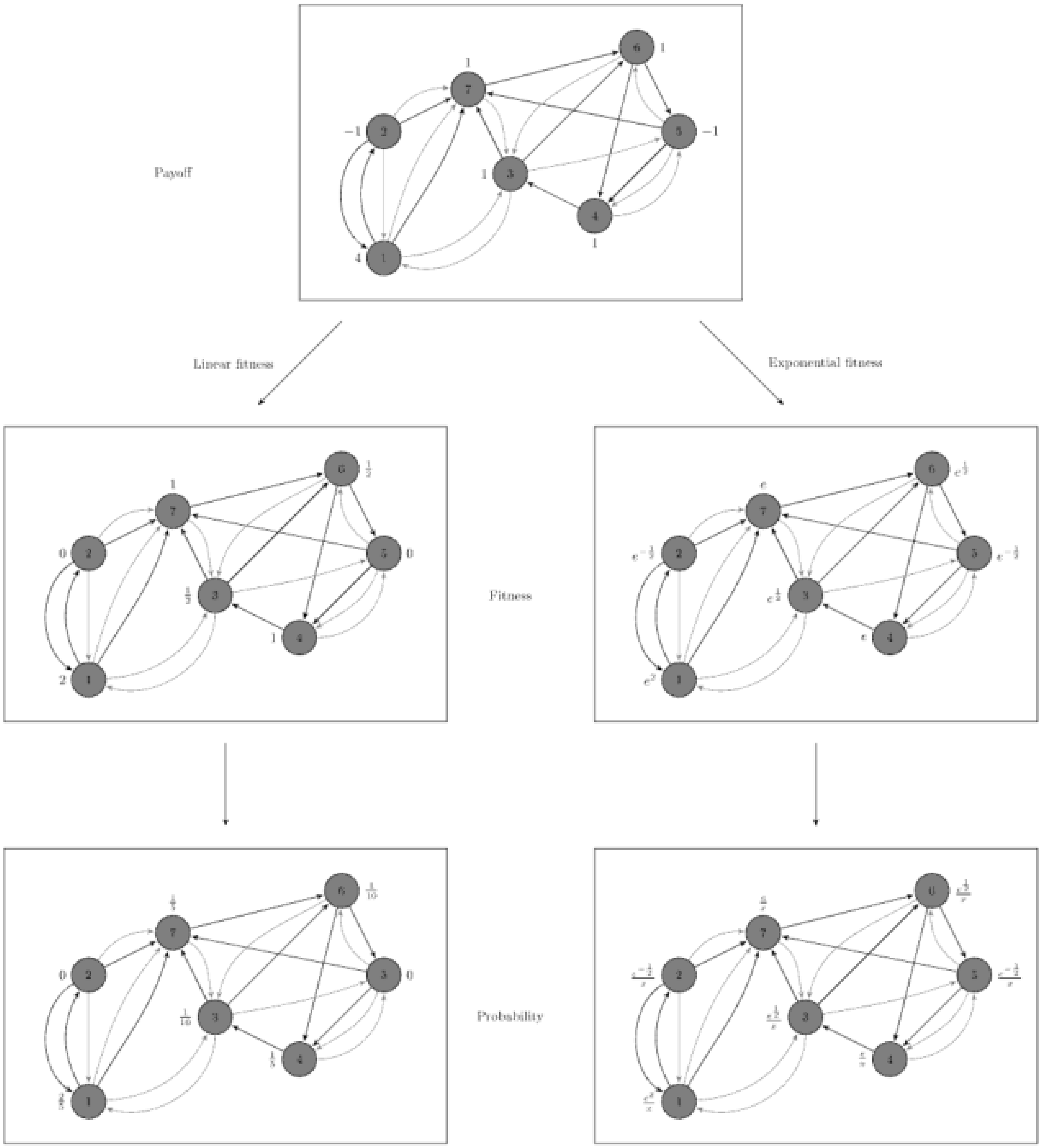

Fig. 4. Illustration of different payoffs to fitness with $\begin{array}{ccc} & A & B \\ A & -1 & 2 \\ B & 1 & -2 \end{array}$. The residents (type A) are blue and the mutants (type B) red. The black edges are the edges of the interaction graph and the red are the edges of the reproduction graph. In the figure of the first row we show the payoff for every vertex. In the next row we show the fitness which is either a linear function of the payoff but at least 0; or an exponential function of the payoff. Finally, in the third row, with each vertex we show the probability, which is the normalized fitness, that the vertex is selected for reproduction (in the last figure, the number x is the sum of the fitness, i.e., $x = e^2 + 2e + 2e^{\frac{1}{2}} + 2e^{-\frac{1}{2}}$).

Table 1. Complexity results for various models and computational questions

	Qualitative	Quantitative
Ecological Scenario	NP-complete	#P-complete
Linear fitness	PSPACE-complete	PSPACE-complete
Exponential fitness	P	PSPACE-complete