Physica A 391 (2012) 2193–2196

Contents lists available at SciVerse ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Roulette-wheel selection via stochastic acceptance

Adam Lipowski[a,*], Dorota Lipowska[b]

[a] *Faculty of Physics, Adam Mickiewicz University, Poznań, Poland*

[b] *Faculty of Modern Languages and Literature, Adam Mickiewicz University, Poznań, Poland*

ARTICLE INFO

Article history:
Received 21 September 2011
Received in revised form 26 November 2011
Available online 9 December 2011

Keywords:
Roulette-wheel selection
Genetic algorithm
Complex networks

ABSTRACT

Roulette-wheel selection is a frequently used method in genetic and evolutionary algorithms or in modeling of complex networks. Existing routines select one of N individuals using search algorithms of $O(N)$ or $O(\log N)$ complexity. We present a simple roulette-wheel selection algorithm, which typically has $O(1)$ complexity and is based on stochastic acceptance instead of searching. We also discuss a hybrid version, which might be suitable for highly heterogeneous weight distributions, found, for example, in some models of complex networks. With minor modifications, the algorithm might also be used for sampling with fitness cut-off at a certain value or for sampling without replacement.

1. Introduction

Finding low-energy configurations of crystals [1], disordered magnets [2] or proteins [3], reconstructing geological structure from seismic data [4], and analyzing X-ray data [5] are just a few examples of optimization problems in physical sciences [6]. Various techniques have been developed to approach these problems and one of the most frequently used is genetic algorithms [7]. Their basic idea is to mimic the way biological evolution creates apparently better fitted species. To do so one needs to represent a pool of optimization methods (routines, functions, strategies, etc.) as a population of individuals, which, as in nature, is subjected to two, in a sense opposing, processes. On the one hand, due to mutations or crossing-over operations, the variability in the population increases. On the other hand, to guide the evolution in a desired direction, one has to trim the population with some selection mechanisms. It turns out that such a biology-inspired scheme allows us to find optimal or nearly optimal solutions of various problems in a very efficient way [8].

Selection is an important part of genetic algorithms since it affects significantly their convergence. The basic strategy follows the rule: The better fitted an individual, the larger the probability of its survival and mating. The most straightforward implementation of this rule is the so-called roulette-wheel selection [8]. This method assumes that the probability of selection is proportional to the fitness of an individual. It can be briefly described as follows. Let us consider N individuals, each characterized by its fitness $w_i > 0$ $(i = 1, 2, \ldots, N)$. The selection probability of the i-th individual is thus given as

$$p_i = \frac{w_i}{\sum_{i=1}^{N} w_i} \quad (i = 1, 2, \ldots, N). \tag{1}$$

Let us imagine a roulette wheel with sectors of size proportional to w_i $(i = 1, 2, \ldots, N)$. Selection of an individual is then equivalent to choosing randomly a point on the wheel and locating the corresponding sector (Fig. 1). When a simple search is used, such a location requires $O(N)$ operations while the binary search needs $O(\log N)$ operations.

---

* Corresponding author.
E-mail address: lipowski@amu.edu.pl (A. Lipowski).

0378-4371/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2011.12.004

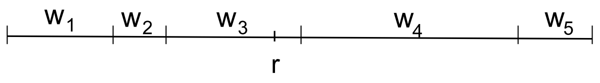

Fig. 1. In a roulette-wheel selection, one constructs a line segment of length $\sum_{i=1}^{N} w_i$ out of consecutive sectors of length w_i ($i = 1, 2, \ldots, N$), generates a random number r such that $0 < r < \sum_{i=1}^{N} w_i$, and locates the corresponding sector (w_3 in this case), thus selecting the respective individual. When a simple search of an N-element list is used for location, the procedure requires $O(N)$ operations, which can be reduced to $O(\log N)$ by means of a binary search.

There are also other methods with the selection probability depending on the fitness, such as a stochastic remainder or stochastic universal selection [9,10]. They have slightly different statistical properties and in general lead to populations of larger variability. In yet another class of selection methods, the ranking or ordering of individuals rather than their fitness plays a central role [11]. It is certainly difficult to indicate which of these methods is the best [12,13]. Although some studies suggest that fitness-based selection methods suffer from certain scaling-related problems [14], there are some works that seem to alleviate this difficulty [15].

Let us notice that the roulette-wheel selection is also used in the generation of complex networks. For example, in some models of growing networks, a newly added site is linked to one of the already existing sites with probability proportional to the degree of this old site [16–18]. This preferential-attachment algorithm is known to generate highly heterogeneous scale-free networks, which recently have been intensively studied [19]. The roulette-wheel selection is also used in certain adaptive network models [20].

Due to simplicity of implementation and straightforward interpretation, the roulette-wheel selection is frequently used in genetic algorithms. Although the binary search significantly reduces computational complexity, still faster implementations would be, in our opinion, desirable. In the present paper, we show that the roulette-wheel selection can be realized with a simple algorithm of typically $O(1)$ complexity. The proposed algorithm does not use searching but is based on a stochastic acceptance of a randomly selected individual.

2. Description and properties of the algorithm

Our algorithm consists of the following steps:

1. Select randomly one of the individuals (say, i). The selection is done with uniform probability ($1/N$), which does not depend on the individual's fitness w_i (Fig. 2).
2. With probability $w_i/w_{\max}$, where $w_{\max} = \max\{w_i\}_{i=1}^{N}$ is the maximal fitness in the population, the selection is accepted. Otherwise, the procedure is repeated from step 1 (i.e., in the case of rejection, another selection attempt is made).

Of course, the probability that the i-th individual will be selected in a single attempt equals $w_i/(Nw_{\max})$. However, since several first attempts might fail, one has to calculate the probability that the individual selected in an unspecified number of attempts will be the i-th individual. Straightforward calculations give

$$p_i' = \frac{w_i}{Nw_{\max}}(1 + q + q^2 + \cdots), \tag{2}$$

where

$$q = \frac{1}{N}\sum_{i=1}^{N}(1 - w_i/w_{\max}) = 1 - \frac{\sum_{i=1}^{N} w_i}{Nw_{\max}} = 1 - \frac{\langle w \rangle}{w_{\max}} \tag{3}$$

is the rejection probability and $\langle w \rangle = \left(\sum_{i=1}^{N} w_i\right)/N$ is the average fitness. The geometrical series (2) is convergent ($0 < q < 1$) and summing it up, we easily find that

$$p_i' = p_i, \tag{4}$$

where p_i is defined in Eq. (1). This shows that the probability distribution of our procedure is indeed the same as in the roulette-wheel selection.

Similarly, we can calculate the average number of attempts τ needed for a single selection of any individual. One obtains

$$\tau = \frac{1}{N}\sum_{i=1}^{N}\frac{w_i}{w_{\max}}[1 + 2q + 3q^2 + \cdots] = \frac{w_{\max}}{\langle w \rangle}. \tag{5}$$

Although τ, which determines the computational complexity of our algorithm, does not depend explicitly on N, the ratio $\frac{w_{\max}}{\langle w \rangle}$ might change with N, depending on the specificity of the problem. We expect, however, that in many applications, for example, those where fitness remains bounded ($w_i < B$) and $\langle w \rangle$ does not diminish to 0 for increasing N, the ratio $\frac{w_{\max}}{\langle w \rangle}$ does not increase unboundedly with N and thus a typical complexity of our algorithm should be $O(1)$.

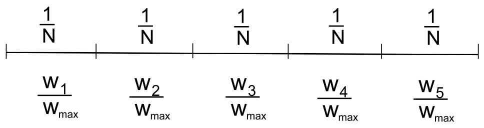

Fig. 2. In the proposed algorithm, one selects randomly (with equal probability $1/N$) an individual (say, i) and accepts such a selection with probability $w_i/w_{\max}$, where $w_{\max}$ is the maximal fitness. In case of rejection, the procedure is repeated anew (i.e., a new individual is selected repeatedly until acceptance). Although it requires extra call(s) of a random-number generator, it remains typically $O(1)$ algorithm.

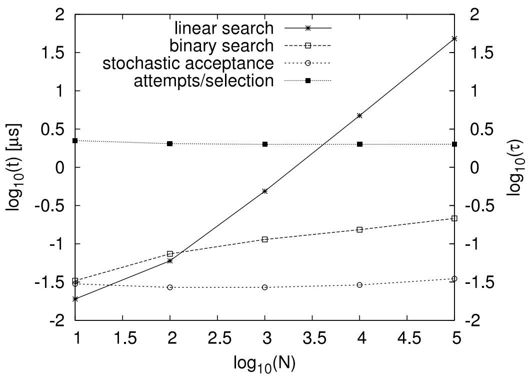

Fig. 3. The average CPU time t (in μs) of roulette-wheel selection of a single individual as a function of the population size N (log–log scale). As expected, the linear search has the complexity $O(N)$ and the binary search $O(\log(N))$. The stochastic acceptance algorithm behaves as $O(1)$ and a slight increase for the largest N is due to excessive memory requirements that exceeded the size of cache memory. Calculations were made on an Intel Xeon 2.8 GHz processor. Filled squares show the average number of attempts per selection.

One can examine a more general algorithm where acceptance is made with probability w_i/A, where $A > w_{\max}$ is a certain constant. For such an algorithm the selection probability p'_i also satisfies Eq. (4). However, a smaller acceptance probability results in a larger rejection probability and the efficiency of the algorithm is reduced (as τ increases). For $A < w_{\max}$, some of the fractions w_i/A may turn out greater than unity and the equality (4) does not hold. Thus, the choice $A = w_{\max}$ ensures optimal performance.

To examine its performance, we applied our algorithm to a population of N individuals with fitness being a uniformly distributed random number $0 < w_i < 1$. We implemented the roulette-wheel selection algorithms using linear or binary search as well as our stochastic acceptance method. We found that the distributions of selected individuals generated by these programs were within statistical errors the same. Average execution time per single selection as a function of N is shown in Fig. 3. As expected, the linear and binary search show $O(N)$ and $O(\log N)$ behavior, respectively. Our algorithm requires basically N-independent CPU time and a slight increase for large N is due to excessive memory requirements that exceeded the size of cache memory. We also measured the average number of attempts τ. In our numerical example, $w_{\max} \approx 1$ for large N, and $\langle w \rangle \sim 1/2$. Thus, according to Eq. (5), $\tau \approx 2$, and our numerical data are in excellent agreement with this result (Fig. 3).

3. Possible extensions and conclusions

As we have already mentioned in the Introduction, the roulette-wheel selection is used in some models of complex networks that are based on preferential attachment. The networks generated in this way are strongly inhomogeneous with a small fraction of sites having a very large degree. The distribution of weights, which are proportional to degrees, is thus very broad and the ratio $\frac{w_{\max}}{\langle w \rangle}$ might be quite large. Consequently, the efficiency of our algorithm might diminish. However, a simple modification of the algorithm might lead to a better performance. First, let us assume that there is one weight that is much larger than the others, say $w_1 (= w_{\max} \gg w_i,\ i = 2, 3, \ldots, N)$. In this case we might use the following hybrid algorithm, which combines search and stochastic acceptance: With probability $p_1 = w_1/\sum_{i=1}^{N} w_i$ one selects the first individual and with probability $1 - p_1$ one of the remaining $N - 1$ individuals (using roulette-wheel applied to $N - 1$ weights). In the latter step, the stochastic acceptance should be quite efficient since the largest weight w_1 was removed. The probability of selection of the individual $i(> 1)$ equals $(1-p_1)\frac{w_i}{\sum_{j=2}^{N} w_j} = \frac{w_i}{\sum_{j=1}^{N} w_j}$, thus it is indeed equal to (1). Generalization to the case where there are several much larger weights is straightforward.

To ensure a sufficiently large variability of the population, it is sometimes desirable in optimization problems to use sampling without replacement. In order to guarantee that a once selected individual is never selected again, one can simply set its fitness to 0. However, when the individual happened to have the maximal fitness, a new maximum should be found. Slightly sacrificing efficiency, one can use the old maximum to calculate the acceptance probability $w_i/w_{\max}$. Similarly, when fitness of individuals is known to be bounded ($w_i < B$), it is not needed to keep track of the current maximum, as one can use w_i/B as the acceptance probability. Although the efficiency of the algorithm is reduced, it still should remain of the $O(1)$ type [21].

The replacement of $w_{\max}$ with a certain constant $A < w_{\max}$ in our algorithm might be yet another way of increasing variability in the population. Indeed, in such a case individuals are selected according to Eq. (1) but with their fitness cut-off at A (i.e., $w_i' = \min\{w_i, A\}$). Our algorithm might also be adapted to evolving systems, such as, for example, complex adaptive systems where fitness of some or all individuals changes in time. Let us emphasize that the performance of a selection method depends on a number of factors and also on a particular type of the optimization problem [13]. A more detailed analysis of the efficiency of our algorithm in comparison with other selection methods will be presented elsewhere.

In conclusion, we have shown that Holland's original idea of using fitness-proportionate selection, i.e., the so-called roulette-wheel selection, can be formulated as an algorithm of typically $O(1)$ complexity. The numerical example shows that for sizes of populations used in genetic-algorithm applications, ranging from 10^2 to 10^4, our algorithm offers a significant CPU gain over existing routines based on a linear or binary search. The algorithm is very simple and we expect that it can be modified and used in even more sophisticated selection schemes.

References

[1] D.M. Deaven, K.M. Ho, Phys. Rev. Lett. 75 (1995) 288;
N.L. Abraham, M.I.J. Probert, Phys. Rev. B 77 (2008) 134117.
[2] K.F. Pal, Phys. A. 223 (1996) 283.
[3] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, A.J. Olson, J. Comput. Chem. 19 (1998) 1639.
[4] T. Shibutani, M. Sambridge, B. Kennett, Geophys. Res. Lett. 22 (1996) 1829.
[5] I. Golovkin, R. Mancini, S. Louis, R. Lee, L. Klein, J. Quant. Spectr. Radi. Tr. 75 (2002) 625.
[6] A.K. Hartmann, H. Rieger, Optimization Algorithms in Physics, Wiley-Vch, 2002.
[7] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan, 1975, re-issued by (MIT Press, 1992).
[8] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989.
[9] J.E. Baker, Reducing bias and inefficiency in the selection algorithm. Proceedings of the Second International Conference on Genetic Algorithms, pp. 14–21 1987.
[10] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: G.J.E Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, California, 1991, pp. 69–93.
[11] J.E. Baker, Adaptive selection methods for genetic algorithms. Proceedings of an International Conference on Genetic Algorithms and Their Applications, pp. 100–111 1985.
[12] For a recent review of various selection methods see, e.g. R. Sivaraj, T. Ravichandran, A review of selection methods in genetic algorithm, Int. J. Eng. Sci. Tech. 3 (2011) 3792.
[13] N.M. Razali, J. Gerghty, Proc. World Congress Engineering 2011 vol II WCE 2011 (London UK).
[14] P.J.B. Hancock, An empirical comparison of selection methods in evolutionary algorithms, in: Selected Papers from AISB Workshop on Evolutionary Computing, Springer–Verlag, London, UK, 1994, pp. 80–94.
[15] S. Gupta, Relative fitness scaling for improving efficiency of proportionate selection in genetic algorithms, in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, pp. 2741–2744, ACM New York, NY, USA 2009.
[16] A.L. Barabási, R. Albert, Science 286 (1999) 509.
[17] P.L. Krapivsky, S. Redner, Phys. Rev. E 63 (2001) 066123.
[18] S.N. Dorogovtsev, J.F.F. Mendes, Phys. Rev. E 62 (2000) 1842.
[19] S. Boccaletti, V. Latora, Y. Moreno, M. Chaves, D.-U. Hwang, Phys. Rep. 424 (2006) 175.
[20] D. Lipowska, A. Lipowski, Naming game on adaptive weighted networks, arXiv:1107.3263.
[21] Genetic algorithms are unlikely to operate in a regime of vanishing $\langle w \rangle$ and thus some bounds imposed on fitness should result in finite τ and $O(1)$ complexity of the algorithm.