



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

[Interaction](#)
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

[Tools](#)
[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

[Print/export](#)
[Create a book](#)
[Download as PDF](#)
[Printable version](#)

[Languages](#)
[Català](#)
[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Stochastic universal sampling

From Wikipedia, the free encyclopedia

Stochastic universal sampling (SUS) is a technique used in [genetic algorithms](#) for selecting potentially useful solutions for recombination. It was introduced by James Baker.^[1]

SUS is a development of [fitness proportionate selection](#) (FPS) which exhibits no bias and minimal spread. Where FPS chooses several solutions from the population by repeated random sampling, SUS uses a single random value to sample all of the solutions by choosing them at **evenly spaced intervals**. This gives weaker members of the population (according to their fitness) a chance to be chosen and thus reduces the unfair nature of fitness-proportional selection methods.

Other methods like [roulette wheel](#) can have bad performance when a member of the population has a really large fitness in comparison with other members. Using a comb-like ruler, SUS starts from a small random number, and chooses the next candidates from the rest of population remaining, not allowing the fittest members to saturate the candidate space.

Described as an algorithm, pseudocode for SUS looks like:

```
SUS(Population, N)
  F := total fitness of Population
  N := number of offspring to keep
  P := distance between the pointers (F/N)
  Start := random number between 0 and P
  Pointers := [Start + i*P | i in [0..(N-1)]]
  return RWS(Population, Pointers)
```

```
RWS(Population, Points)
  Keep = []
  i := 0
  for P in Points
    while fitness sum of Population[0..i] < P
      i++
    add Population[i] to Keep
  return Keep
```

Where **Population**[0..i] is the set of individuals with array-index 0 to (and including) i.

Here RWS() describes the bulk of fitness proportionate selection (also known as "roulette wheel selection") - in true fitness proportional selection the parameter **Points** is always a (sorted) list of random numbers from 0 to **F**. The algorithm above is intended to be illustrative rather than canonical.

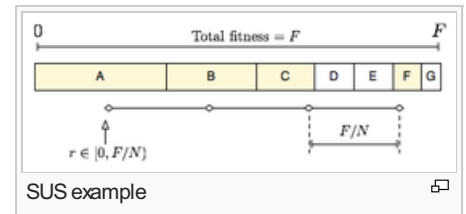
See also [\[edit\]](#)

- [Fitness proportionate selection](#)
- [Reward-based selection](#)

References [\[edit\]](#)

- ↑ Baker, James E. (1987). "Reducing Bias and Inefficiency in the Selection Algorithm". *Proceedings of the Second International Conference on Genetic Algorithms and their Application* (Hillsdale, New Jersey: L. Erlbaum Associates): 14–21.

Categories: [Genetic algorithms](#) | [Stochastic algorithms](#)



This page was last modified on 30 July 2015, at 16:59.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

