Bi-directional circular linked lists in fitness caching

Tapio Tyni and Jari Ylinen
KONE Corporation, R & D
P.O. BOX 677
05801 Hyvinkää, FINLAND

Abstract

Caching the chromosomes with bi-directional circular linked lists reduces the number of fitness evaluations in the GA's providing a built-in bookkeeping property to store and maintain only the chromosomes referenced from the current active region of the problem search space. Keywords: GA, Caching, Epochal evolution.

1 PRINCIPLE AND TEST CASE

Caching is a commonly used technique in the computer science. It has been applied also with the GA's, e.g. in [Kra99]. Instead of the linear tables or linked lists, the bi-directional circular linked lists in the context of GA's are able to reduce the memory space requirements of the cached chromosomes by exploiting the epochal behaviour of the evolutionary search. While the GA's proceed towards the solution of the problem the population has typically periods - epochs - when it wanders around a local extreme point. During these epochs, the number of unique chromosomes that standard mutation and crossover can produce is limited. Only the chromosomes belonging to the present region of the search space need to be cached. Arranging the hash table overflow areas as illustrated in the Figure 1 yields to the built-in behaviour of overwriting the older, obsolete chromosomes in the cache memory with the chromosomes from the current, active region of the search space. Extra bookkeeping to manage data is not needed. The evolution of the fitness value in the Figure 2b illustrates the epochal behaviour of the search with a simple string search problem. The contour map in the Figure 2a shows the age of the chromosomes in the cache memory. When a new epoch begins, the earlier cached chromosomes are no more valid and the new chromosomes from the active search space region conquer the cache. Good examples occur during the generations 36-62 and 63-100. The theoretical probability to find a chromosome from the cache with the actual hits is also shown in the Figure 2a [TY99]. The savings in computational load and memory space are often crucial in the embedded real world real time applications. There the presented method may offer one alternative to obtain the tough goals of an online system.

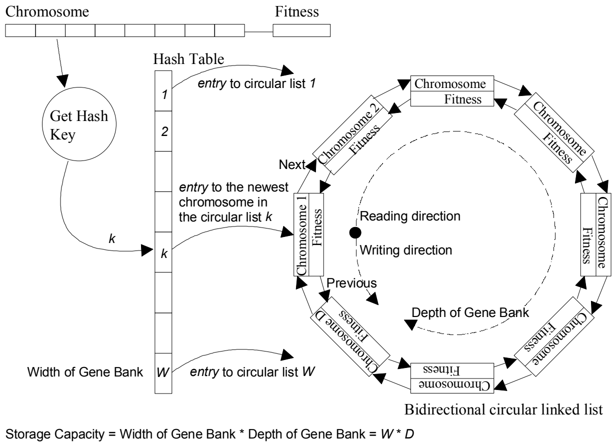

Figure 1. Illustration of the cache organisation

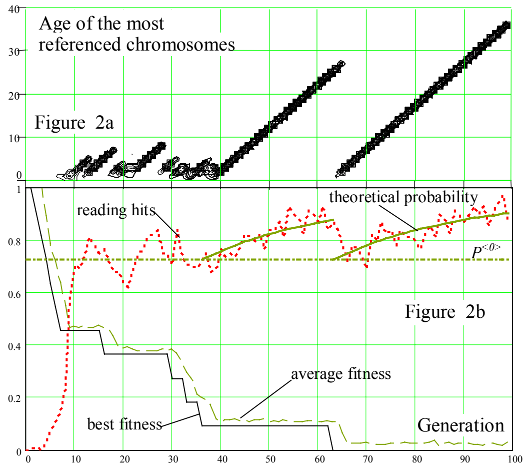

Figure 2. A simple string search test problem

References

[Kra99] Jozef Kratica: *Improving Performances of the Genetic Algorithm by Caching*, Computers and Artificial Intelligence, Vol. 18, No. 3, pp. 271-283, 1999.

[TY99] Tyni T., Ylinen J.: *Improving the Performance of Genetic Algorithms with a Gene Bank*. Proceedings of EUROGEN99, Report A2/1999, University of Jyväskylä, Finland, pp. 162-170, 1999.