<https://www.princeton.edu/~chaff/publication/DAC2001v56.pdf>

One of the most popular strategies, which falls somewhere in the middle of this spectrum, is the dynamic largest individual sum (DLIS) heuristic, in which one selects the literal that appears most frequently in unresolved clauses.

A counter is assigned to each literal, which records a score. The counters are initialized to the number of occurrences of a literal in the initial formula. During the search process, the zChaff SAT-Solver learns new clauses, constructed by the function analyzeConflict(), which are added to the clause database during the search process. When these clauses are added to the database, the score associated with each literal in the clause is incremented. When a branching decision is made, the literal with the highest score is chosen and the variable which belongs to this literal is initialized in a way such that the literal is satisfied. If there are multiple literals with the same score the literal is chosen randomly among them. Finally, all the counters are periodically divided by a constant. This has the effect that VSIDS puts more weight on the most recently added clauses. In other words, VSIDS is an attempt to satisfy conflict clauses, but in particular an attempt to satisfy the most recent clauses. It is considered a quasi-static heuristic, because it doesn’t depend on the variable state such as MOMs heuristics (discussed in next subsection), but it still gradually changes as new clauses are added; it takes search history into account. The following example shows this property.