

# **Changing Representation During Search: Delta Coding**

# Introduction

**Remapping** strategy to focus search around good solutions that have already been discovered either by initial GA or by delta coded GA

The best solution parameters are saved as the *interim solution* and the GA is restarted with a new random population

Parameter substrings are decoded such that they represent a distance or *delta value* ( $\pm\delta$ ) away from the interim solution parameters: a new hypercube is formed with the interim solution at its origin

Delta coding sustains search by periodically **reinitializing** the population, thereby avoiding less and less population diversity: reduces and enlarges the size of hypercube currently being searched

# Delta Coding Algorithm

Delta coding uses GENITOR as the basic engine for GA

Each parameter, when decoded for fitness evaluation, is applied as a delta value ( $\pm\delta$ ) to the interim solution saved from the previous iteration

**Remapping** (numeric shift) is performed using the 1<sup>st</sup> bit in each parameter as a sign bit and remaining bits as the delta value (Table 1)

This remapping method (by delta coding) of applying the new strings to the interim solution searches a new hypercube with the interim solution at its origin (Figure 2)

Thus avoiding local optima and converging on the global optimum (Figure 3)

The process of selection, crossover and inserting offspring continues

# Delta Coding Algorithm - Remapping

Table 1. Delta coding numeric shift (remapping) example.

numeric parameters	0	1	2	3	4	5	6	7
binary coding	000	001	010	011	100	101	110	111
numeric shifts	0	1	2	3	-3	-2	-1	-0
simple delta coding	000	001	010	011	111	110	101	100

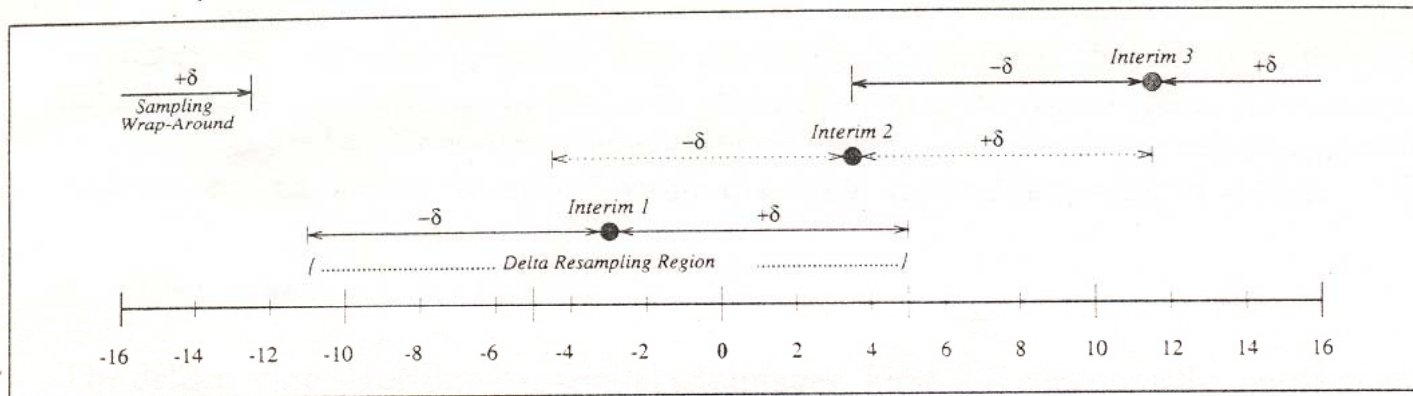
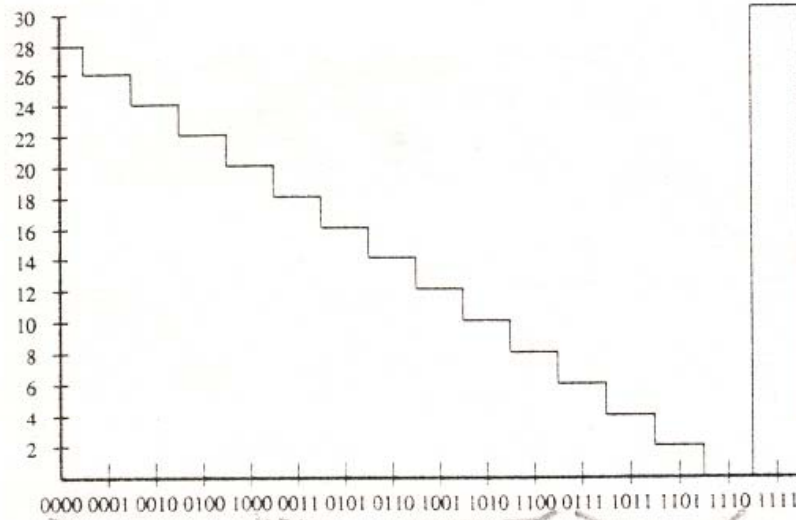
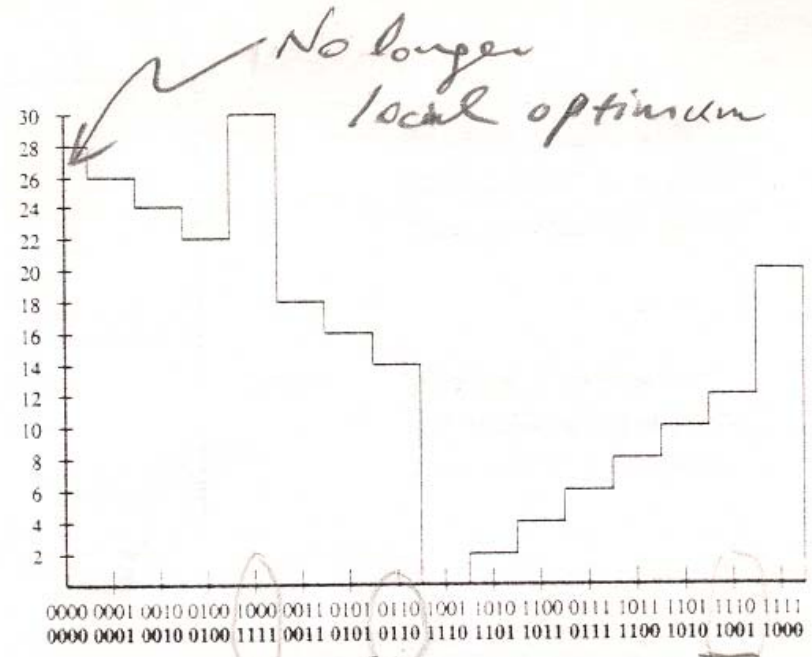


Figure 2. Points sampled in a one-dimensional numeric space using delta coding.

# Delta Coding Algorithm - Remapping



(a)



(b)

Figure 3. Remapping search space with delta coding; interim solution 0000. Figure (b) includes the delta encoding (row 1) and the original encoding (row 2).

## Delta Coding Algorithm - Reinitialization

At the end of each delta iteration, the size of the hypercube is changed by altering the parameter length: **reinitialization**

If the new best solution is different from the previous interim solution, the number of bits used to encode each delta value ( $\pm\delta$ ) is reduced by one bit: this shrinks the hypercube and focuses the search in subpartitions that appear promising (Figure 4)

When a delta iteration converges to exactly the same solution, the number of bits used is increased by one bit: this expands the hypercube and allows the GA to search a larger partition of the search space (Figure 5)

# GA and Tabu Search

Chae Y. Lee

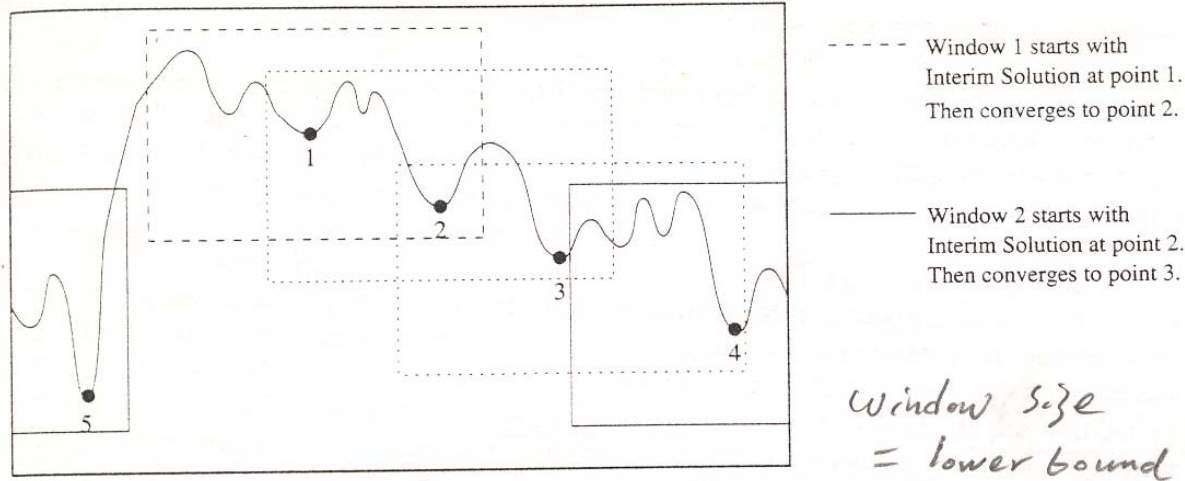


Figure 4. Crawling along a rough fitness surface. Assuming the search has converged to the interim solution at point 1, the subpartition of hyperspace defined by window 1 is searched. This allows the search to converge to point 2. The subpartition of hyperspace surrounding point 2 is then searched, converging on point 3. The search eventually converges on point 5.

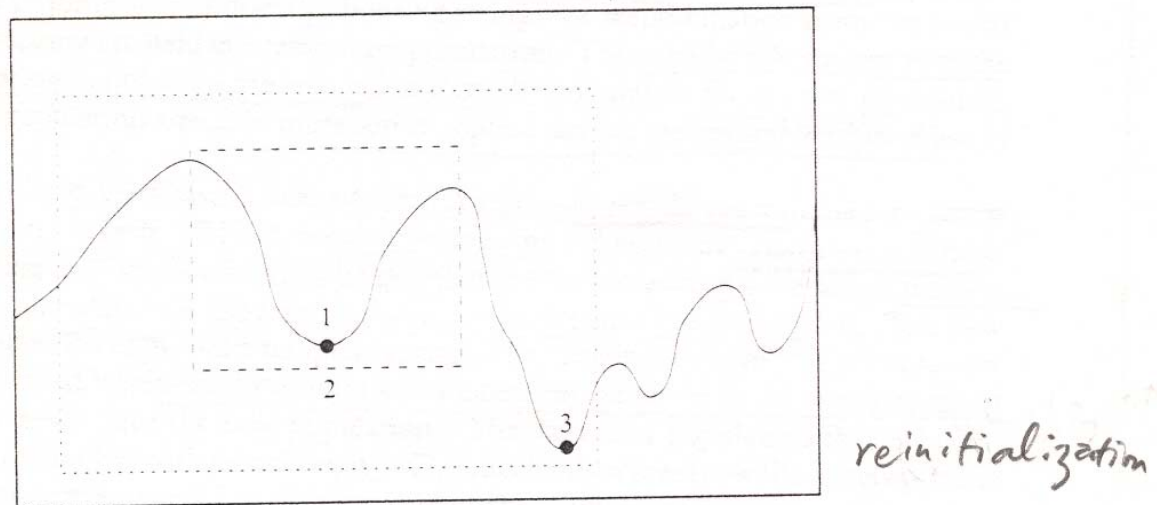


Figure 5. Expanding the delta coding search window after converging to the same interim solution on consecutive iterations using a reduced search space string representation.

# De Jong Function Test Bed

