MAE 552 – Heuristic Optimization

Lecture 22

March 15, 2002

Topic: Tabu Search

tp://www.cs.sandia.gov/opt/survey/ts.ht

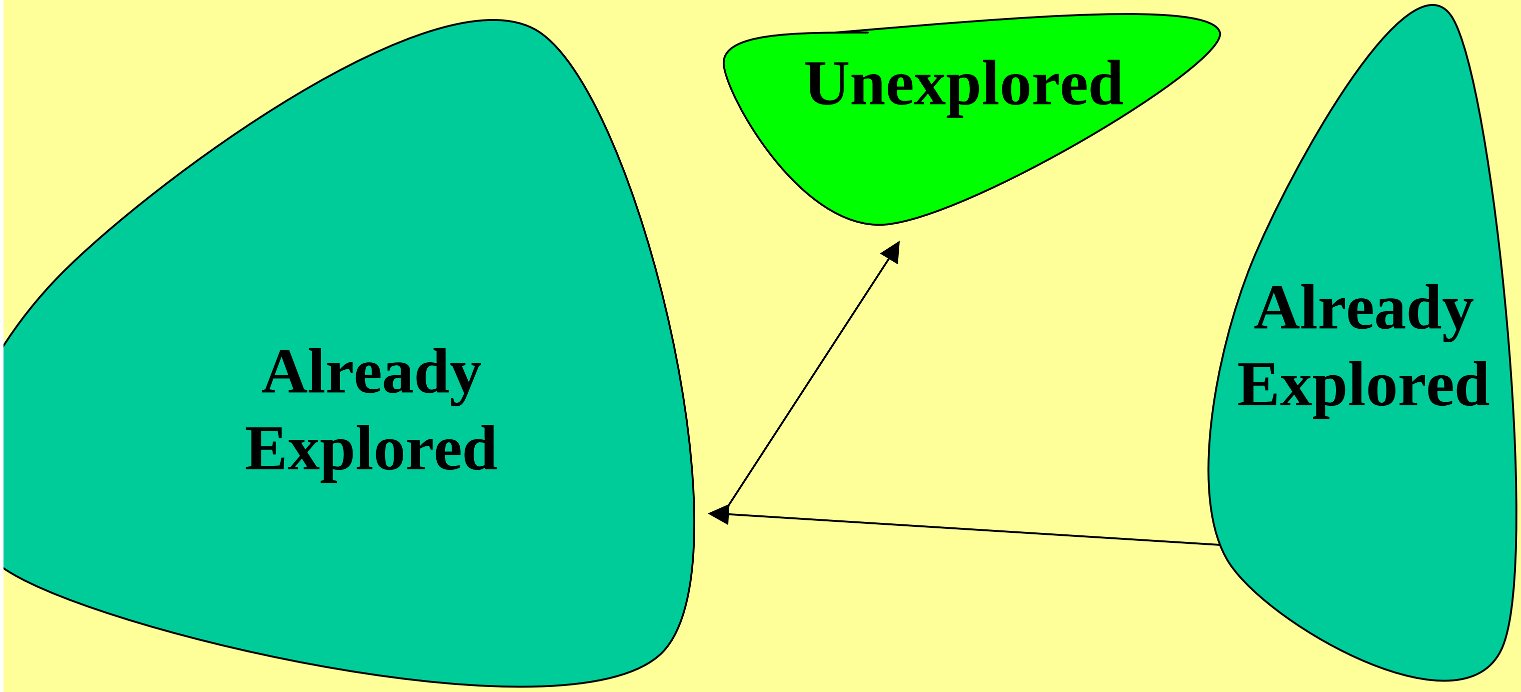
Tabu Search –

When we discussed Simulated Annealing the use o
the Temperature parameter allowed the search to escape local optima.

Another method ‘Tabu Search’ uses another metho
to accomplish the same task.

A ‘memory’ forces the search to explore new areas
of the search space.

Tabu Search –



Algorithm **'remembers'** where is was and is directed to new unexplored areas of the search space

Tabu Search - Characteristics

Unlike SA, Tabu Search is deterministic

The overall approach is to avoid entrainment in cycles by forbidding or penalizing moves which ta the solution, in the next iteration, to points in the solution space previously visited (hence "Tabu").

Tabu search is essentially an extension of steepest descent. At each iteration we examine the entire neighborhood and choose the best move not classified `Tabu.

Tabu Search –Origins

The Tabu search is fairly new, Glover attributes it' origin to about 1977 (see Glover, 1977).

The method is still actively researched, and is continuing to evolve and improve.

The Tabu method was partly motivated by the observation that human behavior appears to operat with a random element that leads to inconsistent behavior given similar circumstances.

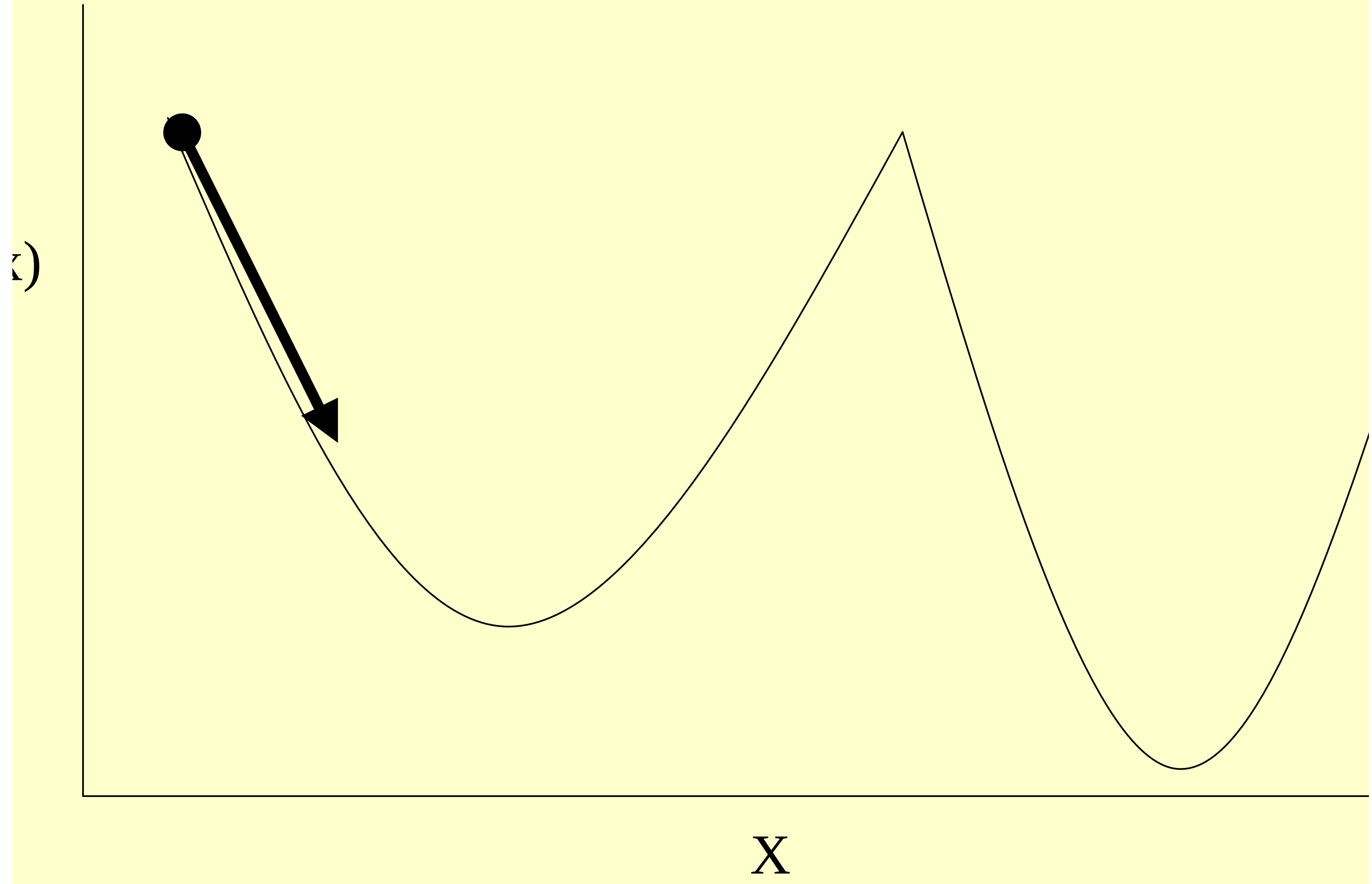
As Glover points out, the resulting tendency to deviate from a charted course, might be regretted a a source of error but can also prove to be source of gain.

Tabu Search –

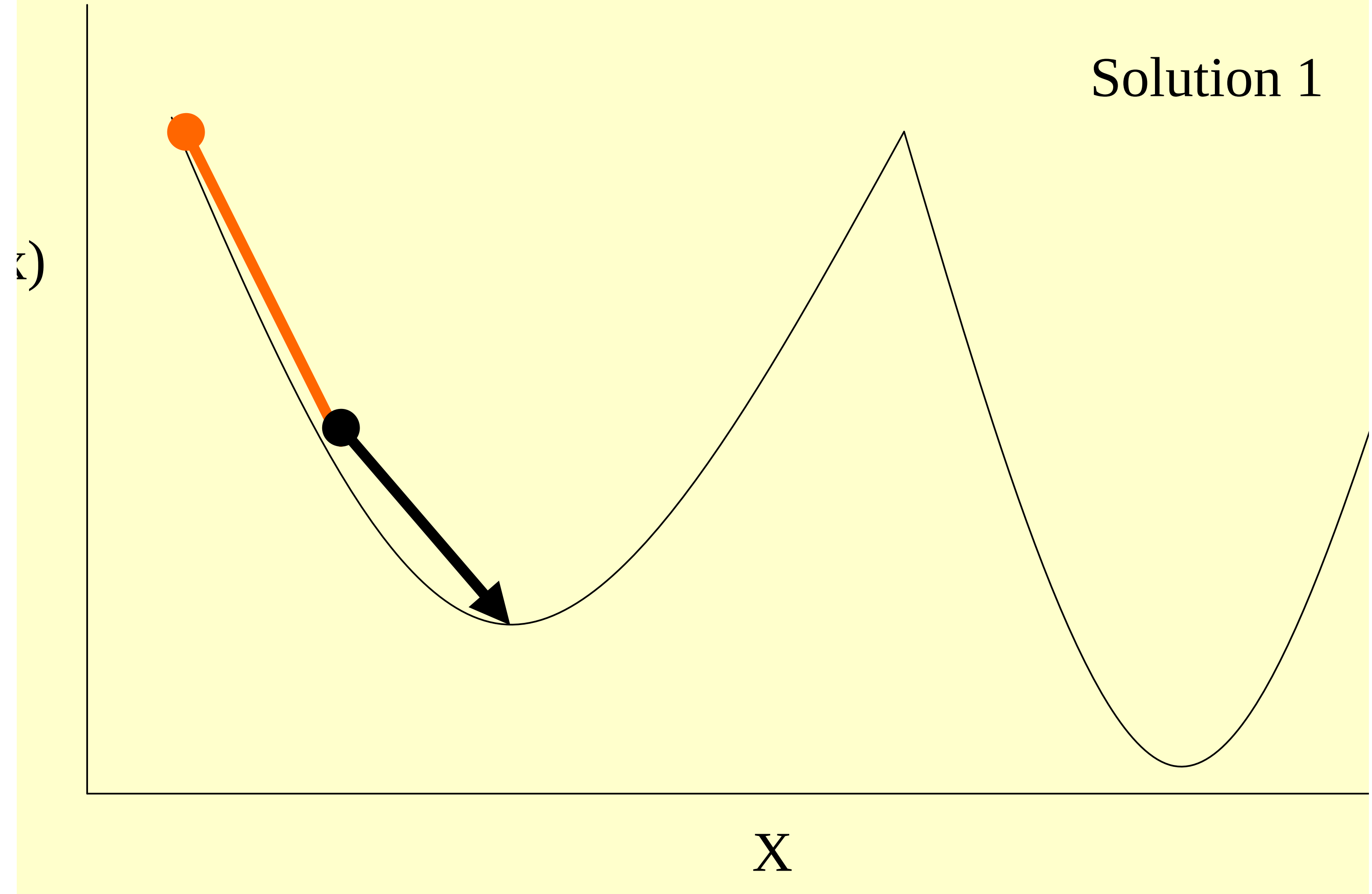
Instead the Tabu search proceeds according to the supposition that there is no point in accepting a ne (poor) solution unless it is to avoid a path already investigated.

This insures new regions of a problems solution space will be investigated in with the goal of avoiding local minima and ultimately finding the desired solution.

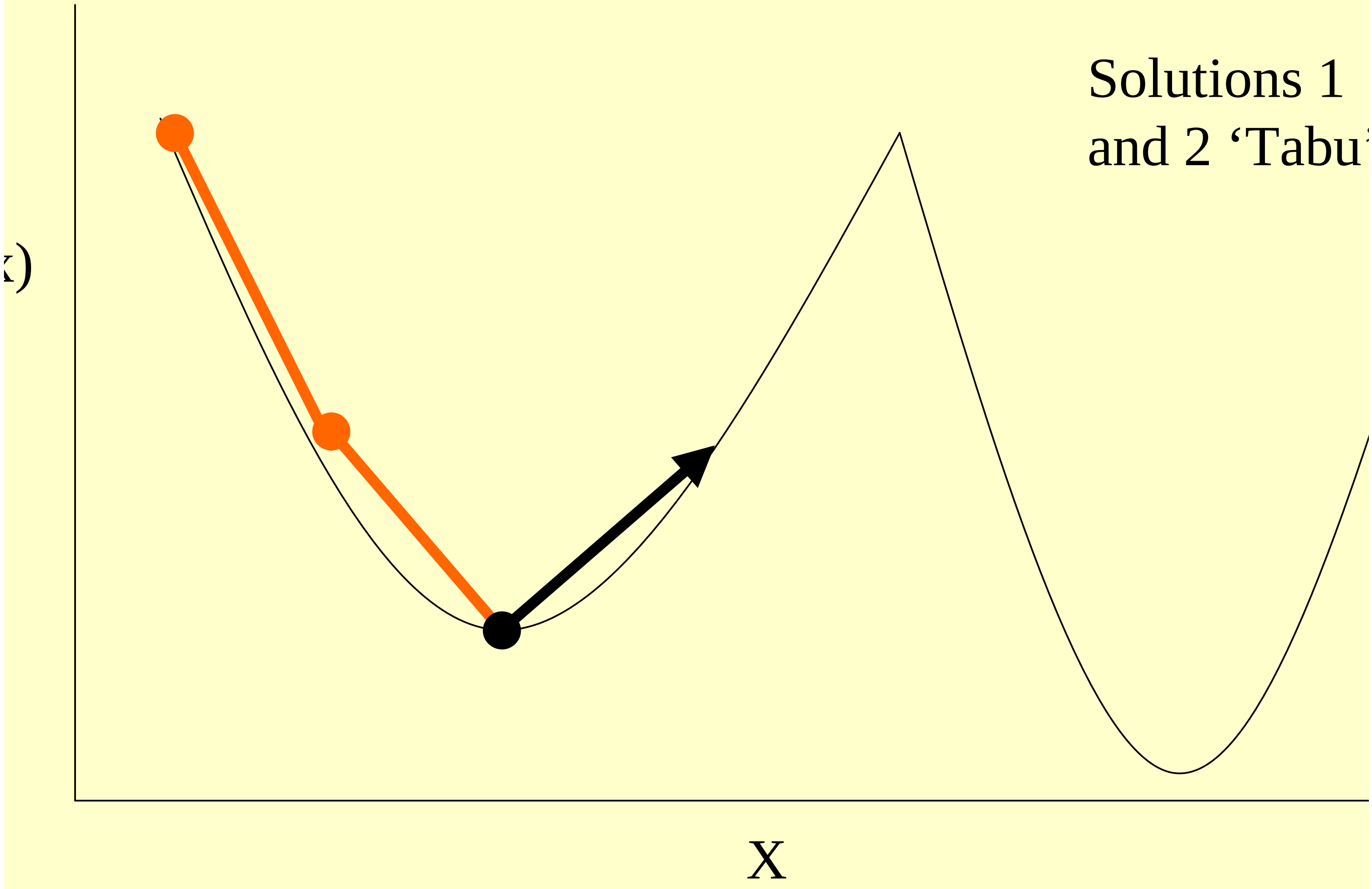
Tabu Search –Move 1



Tabu Search – Move 2

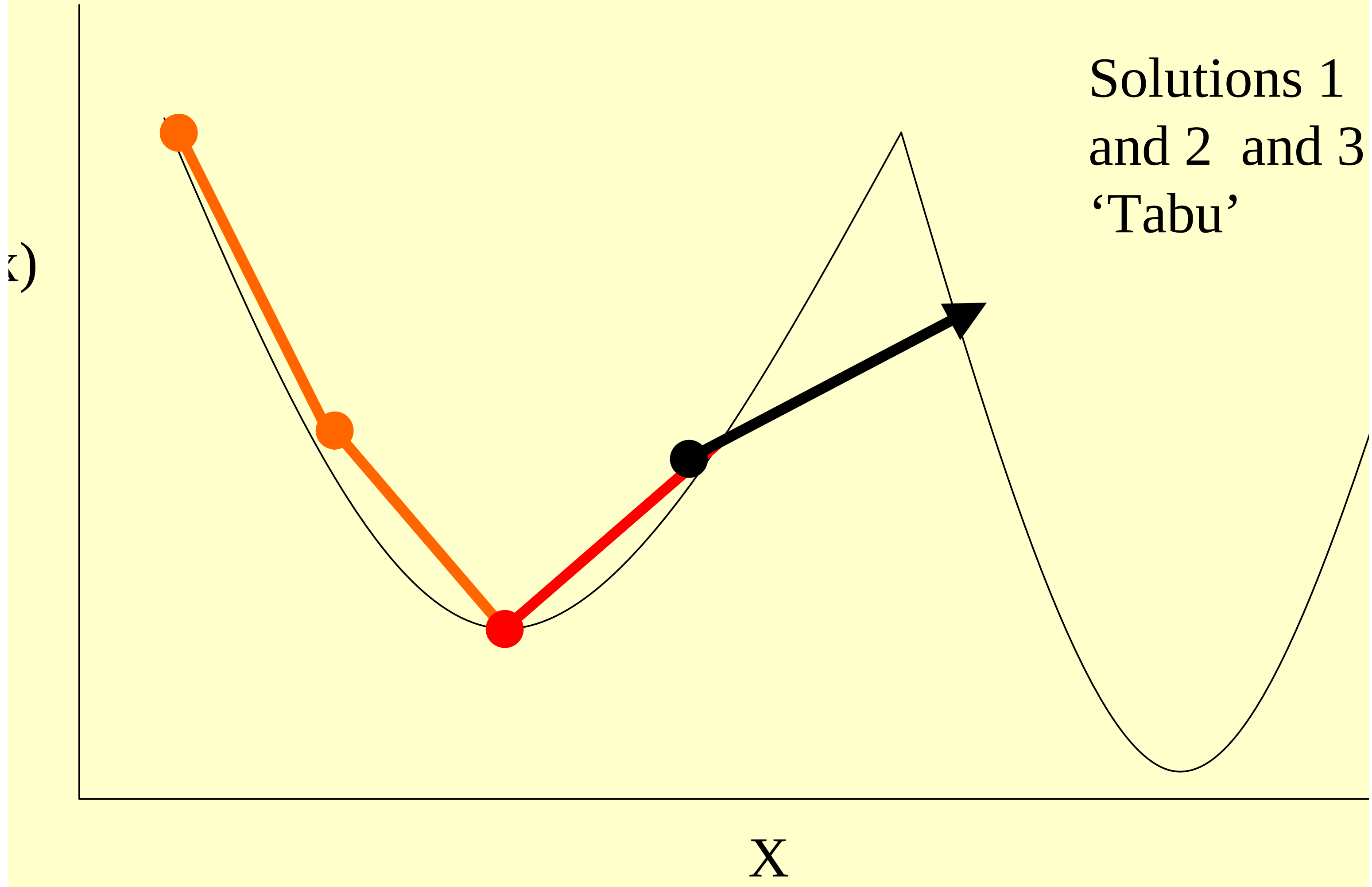


Tabu Search – Move 3



Solutions 1
and 2 ‘Tabu

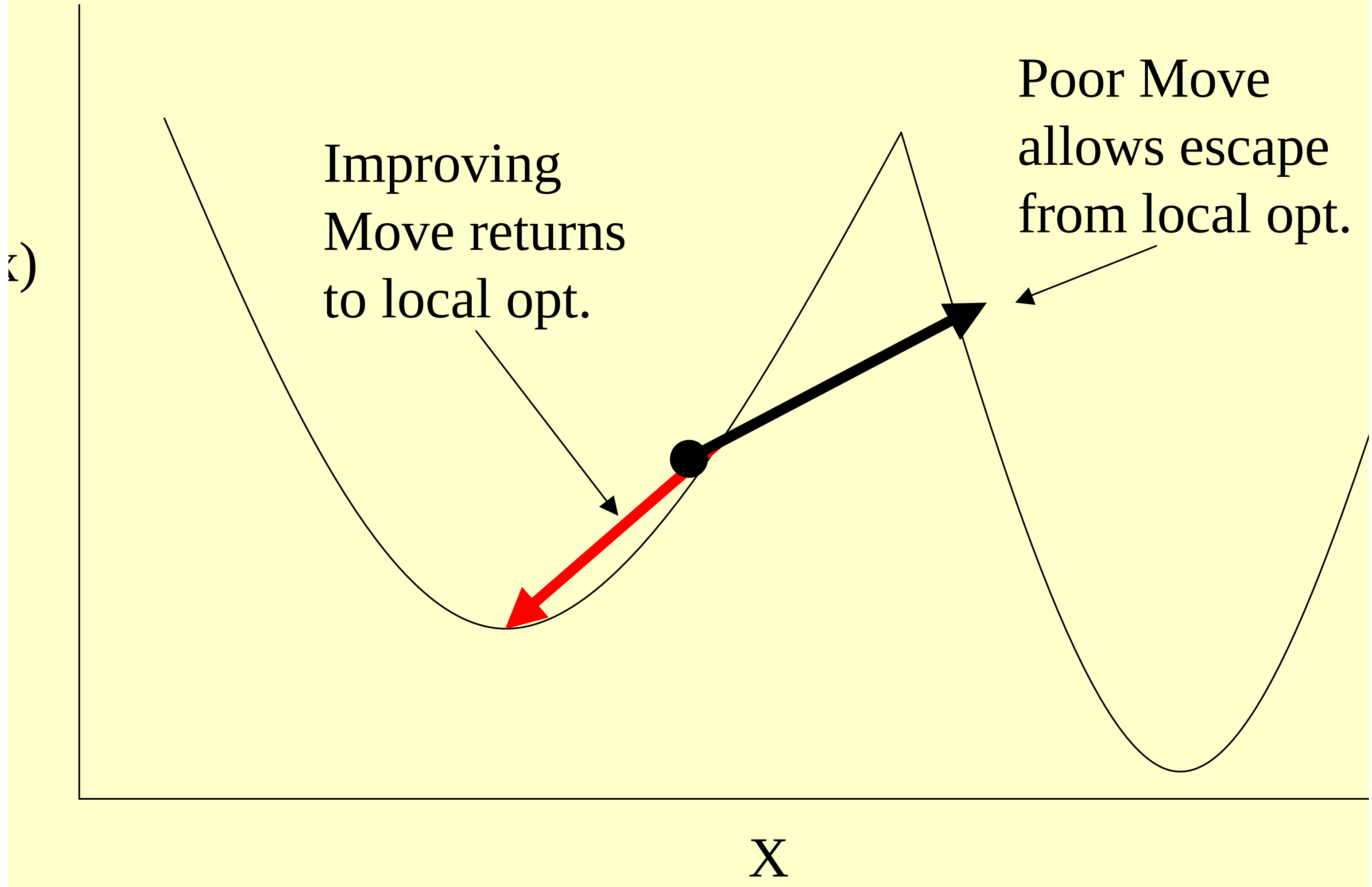
Tabu Search – Move 4



Solutions 1
and 2 and 3
‘Tabu’

Making previous candidate solutions Tabu allows search to escape local optima

Tabu Search – Move 4



- This is important because a strategy of simply choosing th best move would lead us to get stuck at a local optimum

Tabu Search

The Tabu search begins by marching to a local minima.To avoid retracing the steps used, the method records recent moves in one or more Tabu lists.

The role of the memory can change as the algorith proceeds.

– At initialization the goal is make a coarse examination of the solution space, known as 'diversification’.

– As candidate locations are identified the search more focused to produce local optimal solution in a process of 'intensification'.

Tabu Search

In many cases the differences between the various implementations of the Tabu method have to do w the size, variability, and adaptability of the Tabu memory to a particular problem domain.

The Tabu search has traditionally been used on combinatorial optimization problems.

The technique is straightforwardly applied to continuous functions by choosing a discrete encoding of the problem.

Many of the applications in the literature involve integer programming problems, scheduling, routin traveling salesman and related problems.

Tabu Search –Basic Ingredients

Many solution approaches are characterized by identifying a **neighborhood** of a given solution which contains other so-called transformed solutic that can be reached in a single iteration.

A transition from a feasible solution to a transform feasible solution is referred to as a move.

A starting point for Tabu search is to note that suc move may be described by a set of one or more attributes (or elements).

These attributes (properly chosen) can become the foundation for creating an attribute based memory

Tabu Search

Following a steepest descent / mildest ascent approach, a move may either result in a best possib improvement or a least possible deterioration of th objective function value.

Without additional control, however, such a proces can cause a locally optimal solution to be re-visite immediately after moving to a neighbor, or in a future stage of the search process, respectively.

To prevent the search from endlessly cycling between the same solutions, a tabu list is created which operates like a short term memory.

Tabu Search –

Attributes of all explored moves are stored in a list named a **running list** representing the trajectory o solutions encountered.

Then, related to a sublist of the running list a so-called **tabu list** may be introduced. Based on certa restrictions

The **tabu list** implicitly keeps track of moves (or more precisely, salient features of these moves) by recording attributes complementary to those of the running list.

Tabu Search –

These attributes will be forbidden from being embodied in moves selected in at least one subsequent iteration because their inclusion might lead back to a previously visited solution.

Thus, the tabu list restricts the search to a subset o admissible moves (consisting of admissible attribu or combinations of attributes).

The goal is to permit "good" moves in each iteratio without re-visiting solutions already encountered.

Tabu Search – Pseudo-Code

```
ven a feasible solution $x^*$ with objective function value $z^*$:
t $x := x^*$ with $z(x) = z^*$.
ation:
ile  stopping criterion is not fulfilled  do
    begin
    (1) select best admissible move that transforms  $x$ into $x'$ with objective function value
        $z(x')$ and add its attributes to the running list
    (2) perform tabu list management: compute moves  (or attributes) to be set tabu, i.e.,
        update the tabu list
    (3) perform exchanges:
   $x = x'$,  $z(x) = z(x')$;
   if  $z(x) < z^*$  then
            $z^* = z(x)$, $x^* = x$
   endif
while
sult:  $x^*$ is the best of all determined solutions, with objective function value $z^*$.
```