An Introduction to Simulated Annealing

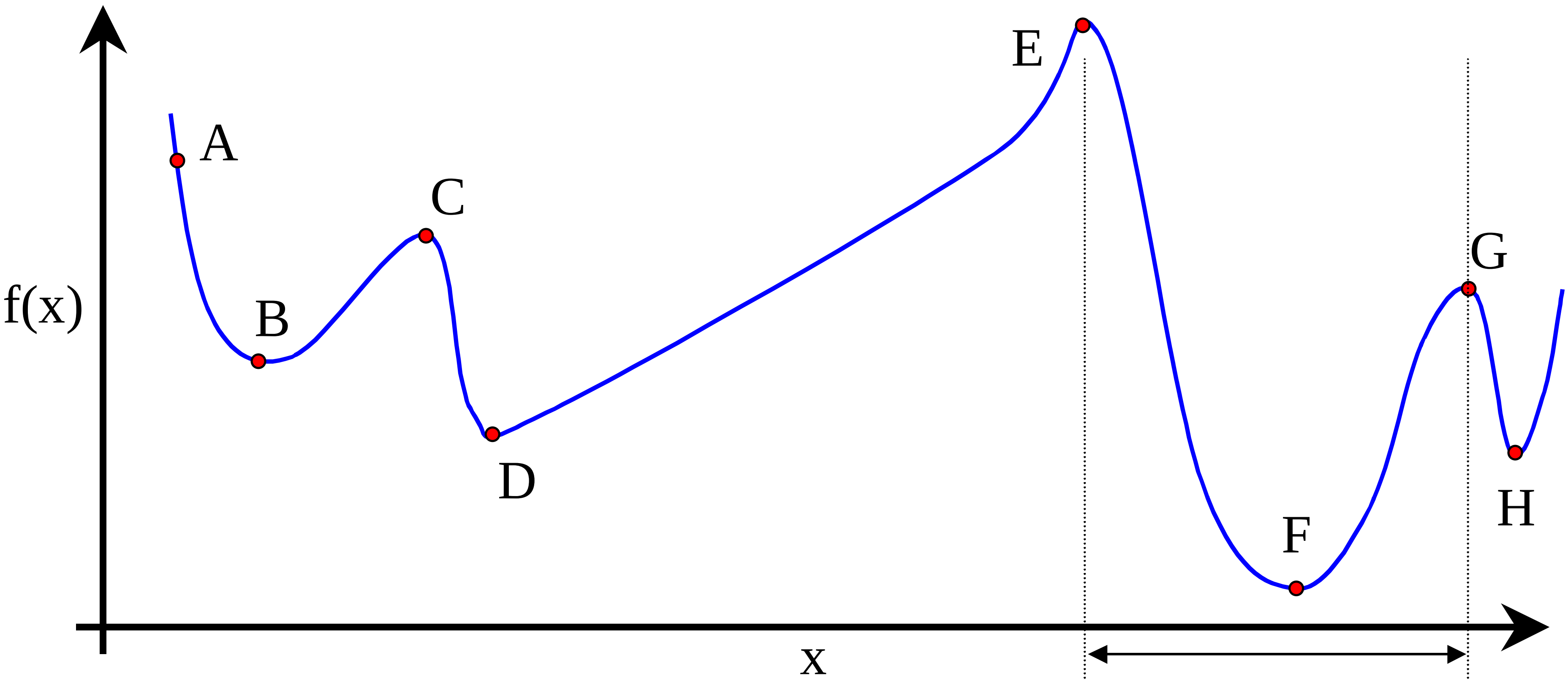
Rajkumar Pant
Aerospace Engineering Department
IIT Bombay

PRESENTATION OUTLINE

- HISTORICAL BACKGROUND
- GLOBAL OPTIMIZATION OF MULTI-MODAL FUNCTIONS
- *MARBLE IN CUBE* ANALOGY
- S A ALGORITHM
- FEATURES OF S A
- TUNING OF S A PARAMETERS
- S A V/S CONVENTIONAL METHODS
- S A FOR FUNCTIONS OF CONTINUOUS VARIABLES
- APPLICATIONS OF S A
- IMPROVEMENTS TO THE BASIC ALGORITHM
- *SIMANN* S A ALGORITHM

GLOBAL OPTIMIZATION OF MULTI-MODAL FUNCTIONS

- How to reach **F**, starting from any **x** ?
  - Greedy Algorithms ==> nearest local minimum
    - A ==> B C ==> D E ==> F G ==> H
    - multiple starts from various initial **x** values



NON-GREEDY ALGORITHMS

- Permit occasional uphill moves
  - sparingly, and in a controlled manner

- Large uphill moves
  - In the initial stages
    - better domain exploration
  - Large changes in **f**(**x**)
    - better chance of improvement
  - Once in a while
    - to climb out of local minima

HISTORICAL BACKGROUND

(Numerical simulation of Annealing — Metropolis et. al, 1953

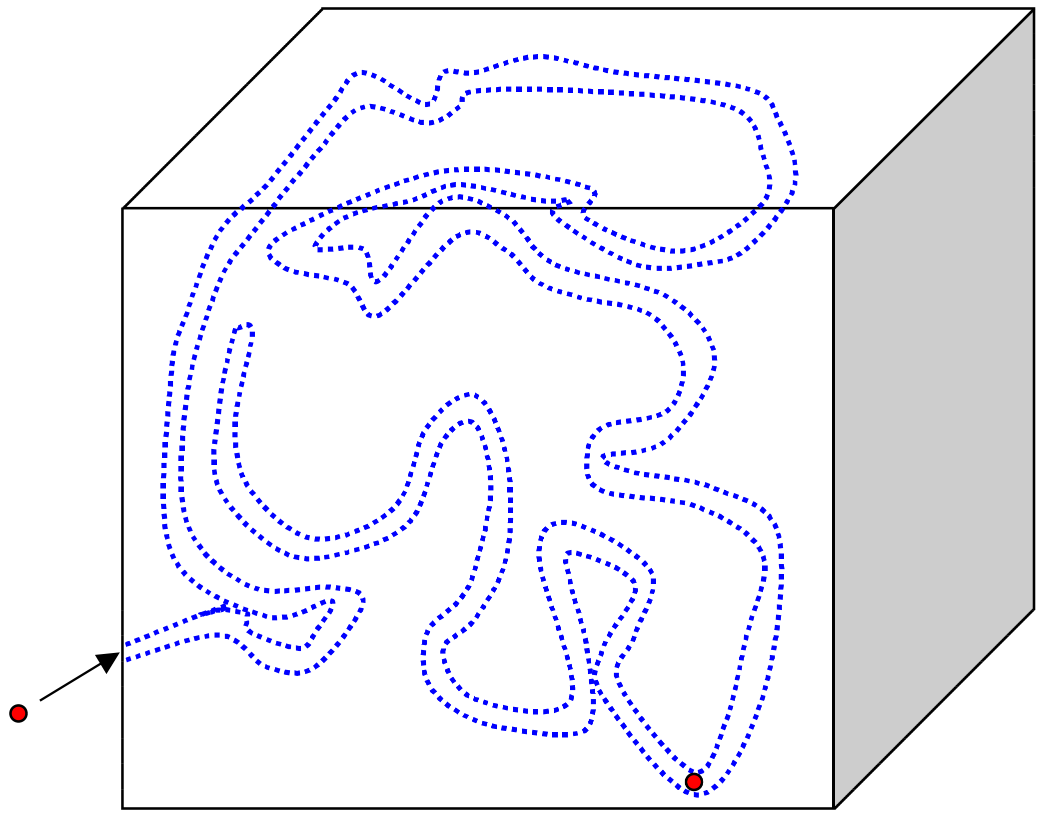
$$p(dE) = e^{(-\,dE/kT)}$$

T	= *temperature*
p(d E)	= *probability of an increase in energy by d E*
k	= *Boltzmann's constant*

(Combinatorial Optimization — Kirkpatrick et. al, 1980; Cerny, 1985

Thermodynamic Simulation	**Combinatorial optimization**
System States	**Feasible solutions**
Energy	**Cost**
Change of state	**Neighboring solution**
Temperature	**Control Parameter**
Frozen state	**Heuristic Solution**

MARBLE-IN-CUBE ANALOGY



How to take the marble to the lowest position in the cube ?

SA ALGORITHM

Solution space X
Objective function f
Neighborhood structure N

- **Select Initial point s_0**
- **Select Initial temperature $T_0 > 0$**
- **Select temperature reduction function α**
  - **Repeat**
    - **Repeat**
      - **Randomly select $s \in N(s_0)$**
      - **$\delta f = f(s) - f(s_0)$**
      - **If $\delta f < 0$**
      - **then $s_0 = s$**
      - **else**
      - **generate a random number $r \in (0,1)$**
        - **if $r < e^{(-\delta f / T_0)}$ then $s_0 = s$**
    - **Until iteration_count = max. iteration**
    - **Set $T = \alpha(T)$**
  - **Until stopping condition = TRUE**
  - **s_0 is the approximation to the global minimum solution**

FEATURES OF SA

- Direct Method & Non-Greedy algorithm
  - Global optimization of multi-modal, discontinuous & noisy functions
- Mathematically proven to converge to global optimum
- Very simple architecture

- Parameters to be decided
  - Solution space **X**, Objective function **f**
    - user defined
  - Neighborhood structure **N**
    - should be adaptively modified
  - Initial point $\mathbf{s_0}$
    - can be randomly selected
  - Initial temperature $\mathbf{T_0}$ & Temperature reduction function α
    - ensure proper “annealing”
  - Stopping Criteria
    - max. number of function evaluations
    - minimum improvement in **f** acceptable

SA V/S CONVENTIONAL METHODS

- Very large number of function evaluations
  - nearly 1000 times more !
  - Exact optimal solution not reached in finite time

- Tuning of SA parameters required before starting
  - may take up 50% of the total time !

- Cannot implicitly handle constraints
  - Penalty Function approach

Example of Penalty Function

Objective Function = $F_{objt} + \Sigma P_k$

$P_k = \quad iv_k \; iact_k \; w_k \; constr_k$

- k = number of constraints
- $constr_k$ = numerical value of k^{th} constraint
- iv_k = 1 if $constr_k > tol_k$, = 0 otherwise
- tol_k = tolerance on target value for k^{th} constraint
- $iact_k$ = 1 if k^{th} constraint is active, = 0 otherwise
- w_k = weight on the value of k^{th} constraint

APPLICATIONS OF SA

- Combinatorial problems
  - VLSI & Computer system design
    - optimal placement of > 10^6 transistors on a chip
    - optimal location of services on a computer network
  - Sequencing & production scheduling
    - Shop-floor, inventory management, FMS
  - Transport Scheduling & Time-tabling
    - Travelling Salesman problem, Locomotive Scheduling
    - Image processing, Building layout design, DNA mapping
- Continuous and mixed functions
  - Engineering Design
    - Aircraft Conceptual Design, Composite Structure modelling
  - Statistical Functions
    - Banking industry, and Financial analysis

SA FOR CONTINUOUS VARIABLES

- Corana et. al
  - ACM transactions on Mathematical Software, **13**(3), 1987
- Features
  - Iterative random search procedure, with adaptive step size reduction
  - maintaining approx. 1:1 rate between accepted and rejected moves
- Tests
  - against Nelder-Mead simplex & Adapted Random Search, on
    - 2 & 4 dim. Rosenbrock's valley function
      - always reached the global minimum
      - 500 to 1000 times higher n_{eval} , compared to Nelder-Mead Simplex
    - parabolic, multi-minima discontinuous function
      - sometime converged to near-global optimal solutions
      - 20% lower total n_{eval} compared to other methods
- Pending tasks
  - How to decide T_{int}, better stopping criteria, and SA parameters ??

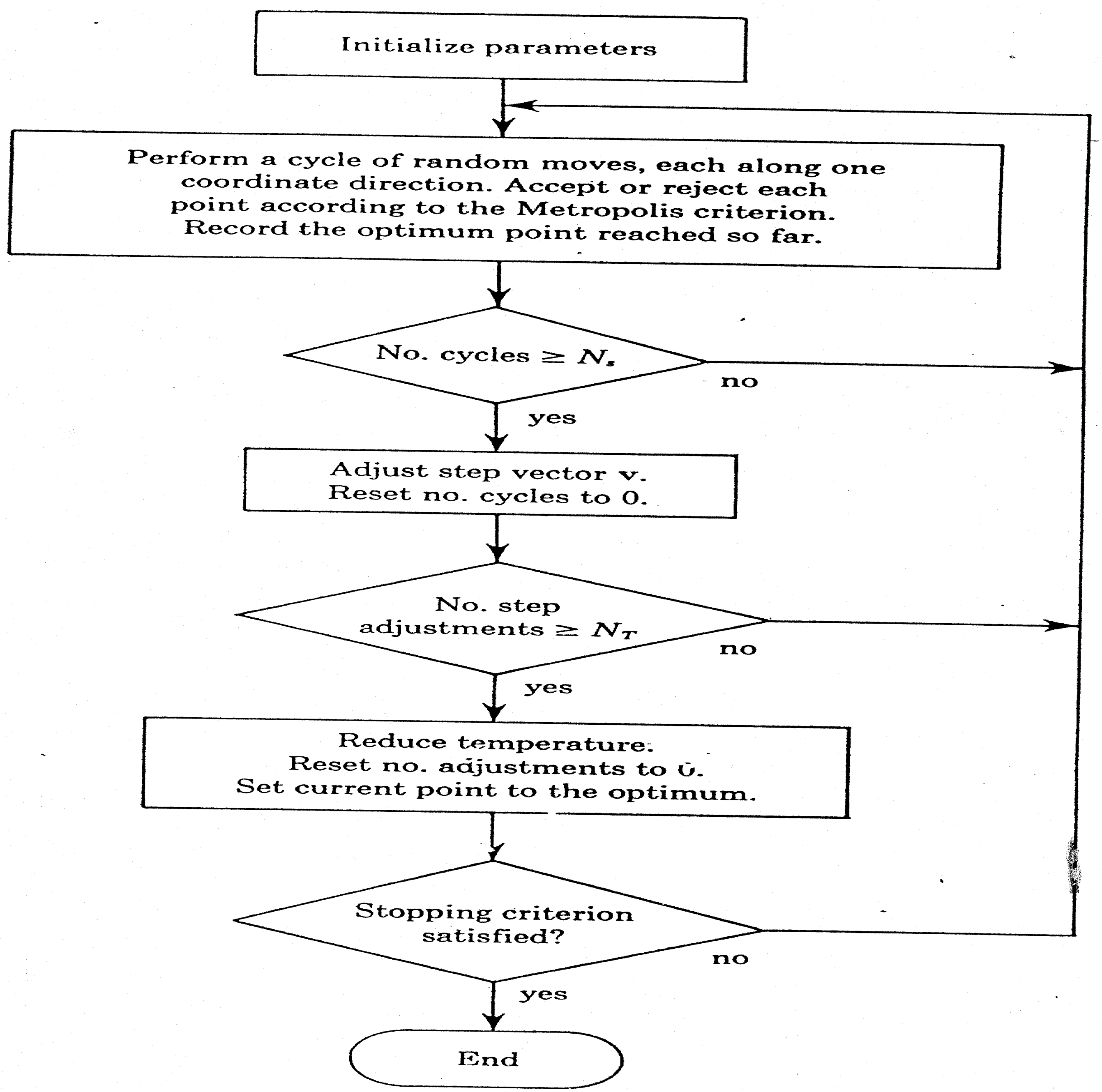
TUNING OF SA PARAMETERS

- Initial Annealing Temperature (T_{int})
  - of the order of expected objective function value
- Temperature reduction factor (R_T)
  - 0.85
- No. of cycles before Temperature reduction (N_T)
  - max (100, 5*n)
- Initial step sizes for design variables (v_i)
  - does not matter, as it is changed adaptively
- No. of cycles before step reduction (N_S)
  - 20
- No. of cycles for checking convergence (N_{eps})
  - 4
- Minimum reduction in Obj. Fun. before termination (eps)
  - user defined

SIMANN SA CODE

- Developed by William Goffe, Univ. of S. Mississipi, 1990
  - Journal of Econometrics, **60**, pp. 65-99, 1994
  - based on algorithm by Corana et. al
- FORTRAN source code available from author / Internet
- Quandt's GQOPT6 Statistical Optimization Package
- Well tested for several statistical objective functions
  - 4 econometric problems & 3 optimization methods from IMSL library
  - best solution in each case with SIMANN
  - independent of starting values
- Improvements
  - test for *globalness* of solution
  - restriction of the search area to parameter subspace
  - methodology for tuning of SA parameters

SIMANN FLOW CHART



The SA minimization algorithm.

TUNING OF *SIMANN* PARAMETERS

- Determination of T_{int}
  - Trial run with T_{int} = 1 and R_T = 1.5
    - Determine T^* at which v_i cover design variable range
  - Trial run with very high T_{int}
    - Determine T^* at which v_i decrease rapidly
  - Set T_{int} slightly greater than T^*
- Determination of R_T & N_T
  - low value => Quenching
  - High value => increase in N_{eval}
  - Few trial runs with progressively decreasing R_T & N_T values
  - Assign highest values without loss in quality of the solution

TUNING OF *SIMANN* PARAMETERS

- Selection of v_i
  - 50% of the range of each design variable
  - Not very important, as it is adjusted automatically
- Selection of N_s
  - Problem dependent, and determined by trial-and-error
- Selection of N_{eps}
  - Large value (4 or 5) for multi-minima functions
- Selection of eps
  - Problem dependent
  - Accuracy of objective function calculation
  - perception of what constitutes worthwhile improvement

SPECIAL FEATURES OF *SIMANN*

- Robust and easy to use algorithm
  - fully self-contained, including random number generator
  - easy to use input file structure, and fairly detailed output file
- Final step sizes indicate sensitivity of design variables
- Excellent tutorial, with Judge’s 2 variable test function
- Number of function evaluations almost constant

The End