Multidisciplinary System Design Optimization (MSDO)

Simulated Annealing

Lecture

5 March 2004

by

Dr. Cyrus D. Jilla

Professor Olivier de Weck

Today's Topics

- What are Heuristics?
- The Origin and Analogy of Simulated Annealing
- The Simulated Annealing Algorithm
- An Example: The Terrestrial Planet Finder Mission
- Sample Results
- Ways to Tailor the Simulated Algorithm
- Summary
- References

What is a Heuristic?

- A Heuristic is simply a rule of thumb that hopefully will find a good answer.
- Why use a Heuristic?
  - Heuristics are typically used to solve complex (large, nonlinear, nonconvex (ie. contain many local minima)) multivariate combinatorial optimization problems that are difficult to solve to optimality.
- Unlike gradient-based methods (such as the simplex algorithm) in a convex trade space, heuristics are NOT guaranteed to find the true global optimal solution in a single objective problem, but should find many good solutions (the mathematician's answer vs. the engineer's answer)
- Heuristics are good at dealing with local optima without getting stuck in them while searching for the global optimum.
- Reference:
  - Schulz, A.S., "Metaheuristics," 15.057 Systems Optimization Course Notes, MIT, 1999.

Types of Heuristics

- Heuristics Often Incorporate Randomization

- **2 Special Cases of Heuristics**
  - Construction Methods
    - Must first find a feasible solution and then improve it.
  - Improvement Methods
    - Start with a feasible solution and just try to improve it.

- **3 Most Common Heuristic Techniques**
  - Genetic Algorithms
  - Simulated Annealing
  - Tabu Search

Origin of Simulated Annealing (SA)

- **Definition:** A heuristic technique that mathematically mirrors the cooling of a material to a state of minimum energy.
- **Origin:** Applying the field of Statistical Mechanics to the field of Combinatorial Optimization (1983)
- Draws an **analogy** between the cooling of a metal and the solving of an optimization problem.
- **Original Paper** Introducing the Concept
  - Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Volume 220, Number 4598, 13 May 1983, pp. 671-680.

The Analogy

- **Statistical Mechanics:** The behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature.
- **Combinatorial Optimization:** Finding the minimum of a given function depending on many parameters.
- **Analogy:** If a liquid material (ie. metal) cools and anneals too quickly, then the material will solidify into a sub-optimal configuration. If the liquid material cools slowly, the crystals within the material will solidify optimally into a state of minimum energy (ie. ground state).
  - This ground state corresponds to the minimum of the cost function in an optimization problem.

4 Key Ingredients for Simulated Annealing (Kirkpatrick et al, 1983)

- A concise description of a configuration (ie. architecture) of the system (Design Vector).
- A random generator of rearrangements of the elements in a configuration (Neighborhoods).
- A quantitative objective function containing the trade-offs that have to be made (Simulation Model and Output Metric(s)).
- An annealing schedule of the temperatures and the length of times for which the system is to be evolved.

The SA Algorithm

- **Terminology:**
  - Γ = Design Vector (ie. Design Architecture)
  - E = System Energy (ie. Objective Function Value)
  - T = System Temperature
  - Δ = Difference in System Energy Between Two Design Vectors

- **The Simulated Annealing Algorithm**

1) Choose a random Γ_i, select the initial system temperature, and outline the cooling (ie. annealing) schedule
2) Evaluate $E(\Gamma_i)$ using your simulation model
3) Perturb Γ_i to obtain a neighboring Design Vector (Γ_{i+1})
4) Evaluate $E(\Gamma_{i+1})$ using your simulation model
5) If $E(\Gamma_{i+1}) < E(\Gamma_i)$, Γ_{i+1} is the new current solution
6) If $E(\Gamma_{i+1}) > E(\Gamma_i)$, then accept Γ_{i+1} as the new current solution with a probability $e^{(-\Delta/T)}$ where $\Delta = E(\Gamma_{i+1}) - E(\Gamma_i)$.
7) Reduce the system temperature according to the cooling schedule.
8) Terminate the algorithm.

- **TPF Example:** We will walk through each of the 8 steps.

Terrestrial Planet Finder (TPF)

- **Mission Statement:** "To study all aspects of planets ranging from their formation and development in disks of dust and gas around newly forming stars to the presence and features of those planets orbiting the nearest stars. Specifically, to conduct a search for Earth-like planets in star systems located within 15 parsecs of our solar system."

- **Primary Mission:** **To detect Earth-like planets** around nearby stars, especially those in the habitable zone where liquid water is likely to exist
  - Bracewell Nulling interferometer

- **Secondary Mission:** To characterize approximately 50 of these Earth-like planets
  - Medium **spectroscopy** (50 planets)
  - Detailed **spectroscopy** (5 planets)

- To **image astrophysical structures** to within milli-arcsecond angular resolution (Michelson interferometer) requires longer baselines

- **http://tpf.jpl.nasa.gov/**

Mathematical Formulation of the TPF Design Problem

16.899
ESD.834

TPF Design Vector: $G = [\gamma_1 \ \gamma_2 \ \gamma_3 \ \gamma_4]$

Variable		Allowable Values
Heliocentric Orbital Altitude (AU)	γ_1	1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5
Collector Connectivity/Geometry	γ_2	SCI-1D, SCI-2D, SSI-1D, SSI-2D
# Collector Apertures	γ_3	4, 6, 8, 10
Diameter of Collector Apertures(m)	γ_4	1, 2, 3, 4

Original Optimization Formulation

$$\text{Objective}: \quad Min \ \frac{\sum_{y=1}^{5} F_y(G)}{\sum_{y=1}^{5} ?_y(G)}$$

Constraints: *Subject to*

Isolation

$$2.5 \le \theta_r \le 20 \text{ milli-arcsec}$$

$$O \le 10^{-6}$$

Integrity

Surveying $SNR \ge 5$

Med. Spectroscopy $SNR \ge 10$

Deep Spectroscopy $SNR \ge 25$

- 640 unique TPF mission architectures
- Complete enumeration determines accuracy
- MDO methods limited to the evaluation of 48 design vectors (7.5% of the full-factorial trade space)

Where

y = year in the mission

Φ= cost

Ψ= number of "images"

θ_r= angular resolution

Ω= null depth

SA Step 1 – Starting Out

- **2 Methods to Select the Initial Design Vector Γ_i**
  - Select a random point (ie. random design architecture) in the trade space
  - Select a known point in the trade space (ie. a design architecture which has already been studied/demonstrated)

- Typically, a random Design Vector is used to begin the SA algorithm (Exception: Construction Problem with few known feasible solutions).

- After a "good" solution(s) has been found, this can be used as the starting point for a new run of the SA algorithm to try and find an even better solution (Assumption: You have now found the most favorable portion of the trade space).

SA Step 2 - Evaluate $E(\Gamma_i)$ Using Your Simulation Model

16.899
ESD.83J

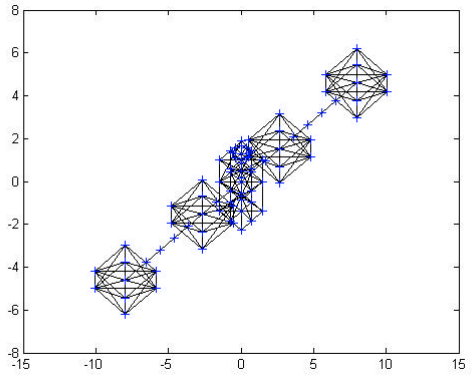
Design Vector

Orbit	1 AU
Interferometer Type	SCI-1D
Number of Apertures	4
Size of Apertures	4 m

Environment → Aperture Configuration → Spacecraft Bus/Payload → Dynamics, Optics, Control, & Structures → Operations → Systems GINA

Metrics

Total Cost	$902M
Total # of Images	1273
Cost/Image	$709K



SA Step 3 - Perturb Γ_i To Obtain a Neighboring Design Vector (Γ_{i+1})

- **What is a Neighbor?**
  - Γ_2 is a neighbor of Γ_1 if Γ_2 can be obtained from a modification of Γ_1.

- **Neighborhood Degrees of Freedom (DOF)**
  - The number of parameters that can vary between Γ_1 and Γ_2 for which they may still be considered neighbors.

- **Neighbor Example:**

$$G_1 = [1\text{AU SCI-1D 4ap } 4\text{m}] \quad G_2 = [1\text{AU SCI-1D 4ap } 2\text{m}] \quad \text{DOF} = 1$$

- **The perturbation to a neighbor is usually done randomly.**

SA Step 4 - Evaluate $E(\Gamma_{i+1})$ Using Your Simulation Model

16.899
ESD.834

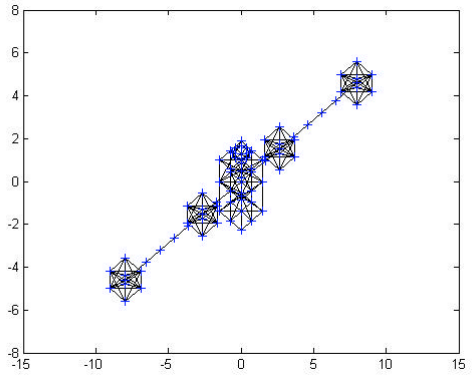
Design Vector

Orbit	1 AU
Interferometer Type	SCI-1D
Number of Apertures	4
Size of Apertures	2 m

Environment → Aperture Configuration → Spacecraft Bus/Payload → Dynamics, Optics, Control, & Structures → Operations → Systems GINA

Metrics

Total Cost	$769M
Total # of Images	769
Cost/Image	$1000K



SA Steps 5 & 6 – Accepting a New Current Solution

- The system energy $E(\Gamma)$ (ie. the Objective Function) is a quantitative measure of the "goodness" of a complex system.
- The current solution is the baseline design architecture from which we explore the trade space. It is the design vector we perturb to find a new neighbor.
- If $E(\Gamma_{i+1}) < E(\Gamma_i)$, Γ_{i+1} is the new current solution.
- If $E(\Gamma_{i+1}) > E(\Gamma_i)$, then accept Γ_{i+1} as the new current solution with a probability $e^{(-\Delta/T)}$ where $\Delta = E(\Gamma_{i+1}) - E(\Gamma_i)$.
- $e^{(-\Delta/T)}$ is called the Boltzman Factor
- This is known as the Metropolis Step
  - Metropolis,N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines", *J. Chem. Phys.*,21, 6, 1087-1092, 1953.

SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

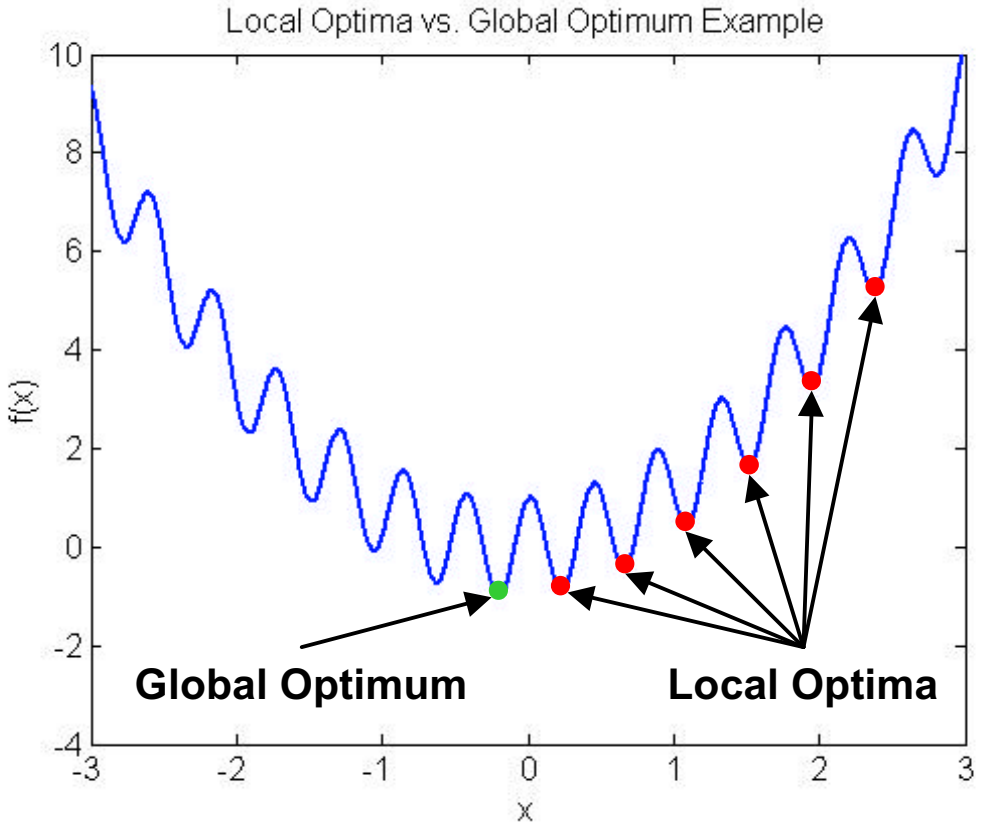
- Why move to a worse current solution?
  - To avoid getting trapped in a local optimum.

- **Local Optimum**
  - A solution is locally optimal if there is no neighbor who has a better objective function value.

- **Global Optimum**
  - A solution is globally optimal if there is no other solution in the entire feasible trade space that has a better objective function value.
  - Note: We are only talking about single objective problems.

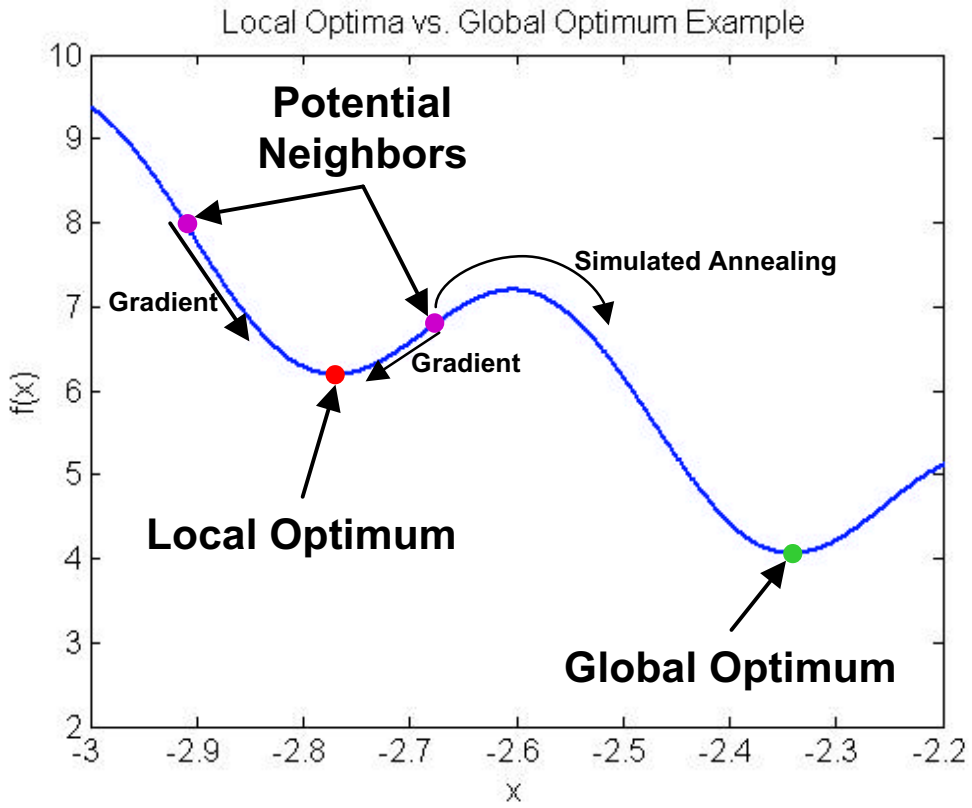
SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- **Local Optima vs. the Global Optimum**
- **Example**
  - **f(x) = cos(14.5x – 0.3) + (x + 0.2)x**
  - **Min[f(x)]**



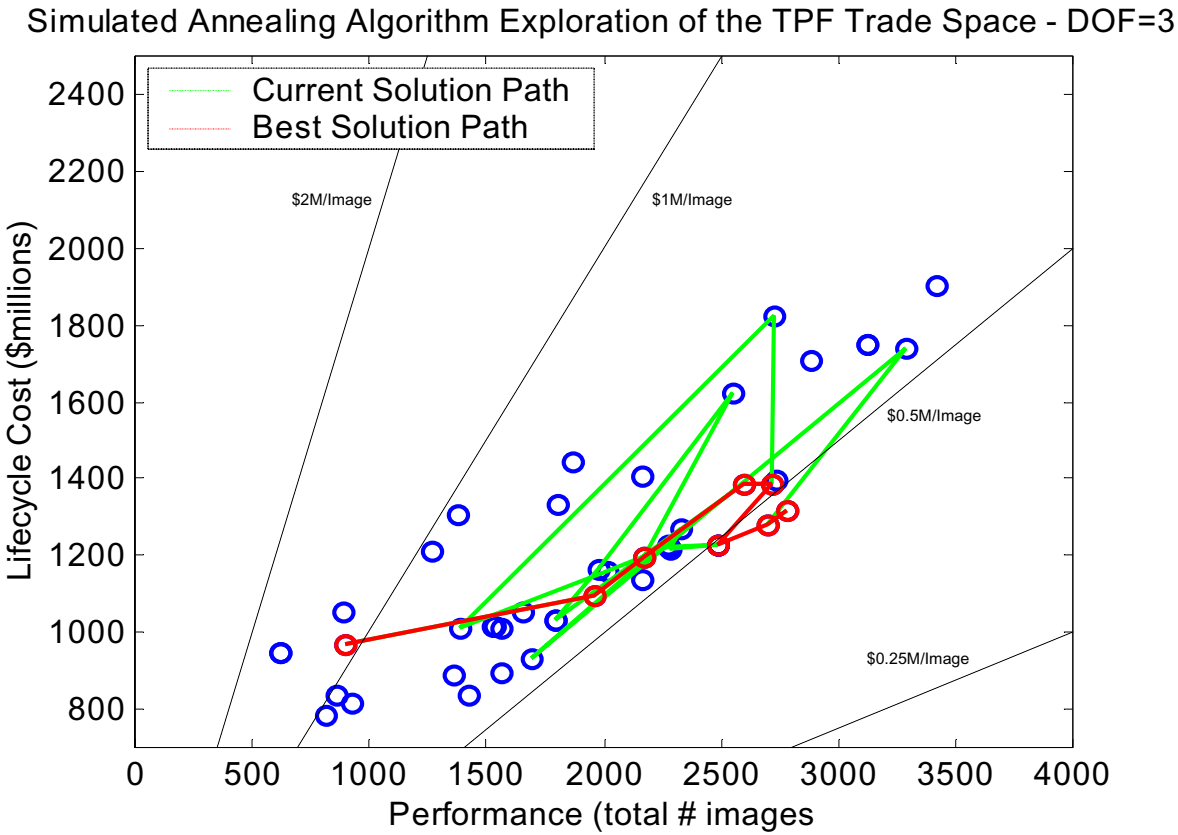
SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- **Local Optima vs. the Global Optimum Example**
- **Min [cos(14.5x – 0.3) + (x + 0.2)x]**
  Subject to -2.2 ≥ x ≥ -3



SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- Can there be a difference between the current solution and the best solution?



SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- Returning to the **TPF Example**
  - $E(\Gamma_1)$ = \$709K
  - $E(\Gamma_2)$ = \$1000K

- If $E(\Gamma_{i+1}) < E(\Gamma_i)$, Γ_{i+1} is the new current solution.
  - Not the Case:
    - \$1000K > \$709K
    - $E(\Gamma_2) > E(\Gamma_1)$
    - Therefore Γ_1 remains the current solution.

- **But** if $E(\Gamma_{i+1}) > E(\Gamma_i)$, then accept Γ_{i+1} as the new current solution with a probability

$$\text{Prob } e^{\left(\frac{-\Delta}{T}\right)}$$

where $\Delta = E(G_{i+1}) - E(G_i)$

SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- $\Delta = E(\Gamma_{i+1}) - E(\Gamma_i)$
- Δ = \$1000K - \$709K = \$291K
- How does the system temperature come into play?
- The probability of moving to a worse solution = $e^{(-\Delta/T)}$
  - As $T \rightarrow \infty$, Probability(moving to a worse solution) $\rightarrow 1$
  - As $T \rightarrow 0$, Probability(moving to a worse solution) $\rightarrow 0$
- $T_o = 1000$ $T_{reduce} = 0.95$
- The probability with which we move to this worse solution with the hope of escaping a possible local minimum decreases over time, and is a function of how many iterations we have executed (ie. How cool the system is. Remember the thermodynamic analogy.).

Iteration #	T	Prob e(-Δ/T)
1	950	0.74
10	599	0.62
20	358	0.44
30	215	0.26
40	129	0.10
50	77	0.02

SA Steps 5 & 6 – Accepting a New Current Solution Cont'd

- Another Key Point
  - The larger the difference Δ in system energy between the two neighbors, the lower the probability of moving to that neighbor.

Iteration #30	Δ	Prob e(-Δ/T)
T = 215	$1600K	0.0006
T = 215	$800K	0.024
T = 215	$400K	0.16
T = 215	$200K	0.39
T = 215	$100K	0.63
T = 215	$50K	0.79

SA Step 7 – Reducing System Temperature

- The initial system temperature is typically large – on the order of magnitude of the expected range of the objective function.

- Two Most Common Cooling Schedules
  - Explore several perturbations at a given temperature, then reduce the temperature to a predetermined value, and repeat.
  - Reduce the temperature between each perturbation, but by a smaller amount.

- Creating a cooling schedule is more of an art than a science, and may involve trial and error.

SA Step 8 - Terminating the SA Algorithm

- How do you terminate the simulated annealing algorithm?
  - Depends upon the cooling/annealing schedule.
- If spending multiple iterations at each temperature, terminate algorithm when T=0.
- If spending only one iteration at each temperature:
  - If a new "best solution" is not found after a given number of iterations, then terminate the algorithm.
  - Set upper bound on the total number of iterations allowed. Terminate the algorithm when this upper bound is reached. This should correspond ~ T=0.
  - Manually terminate the algorithm.

Review of the SA Algorithm

- **Terminology:**
  - Γ = Design Vector (ie. Design Architecture)
  - E = System Energy (ie. Objective Function Value)
  - T = System Temperature
  - Δ = Difference in System Energy Between Two Design Vectors
- **The Simulated Annealing Algorithm**

1) Choose a random Γ_i, select the initial system temperature, and outline the cooling (ie. annealing) schedule.

2) Evaluate $E(\Gamma_i)$ using your simulation model

3) Perturb Γ_i to obtain a neighboring Design Vector (Γ_{i+1})

4) Evaluate $E(\Gamma_{i+1})$ using your simulation model

5) If $E(\Gamma_{i+1}) < E(\Gamma_i)$, Γ_{i+1} is the new current solution

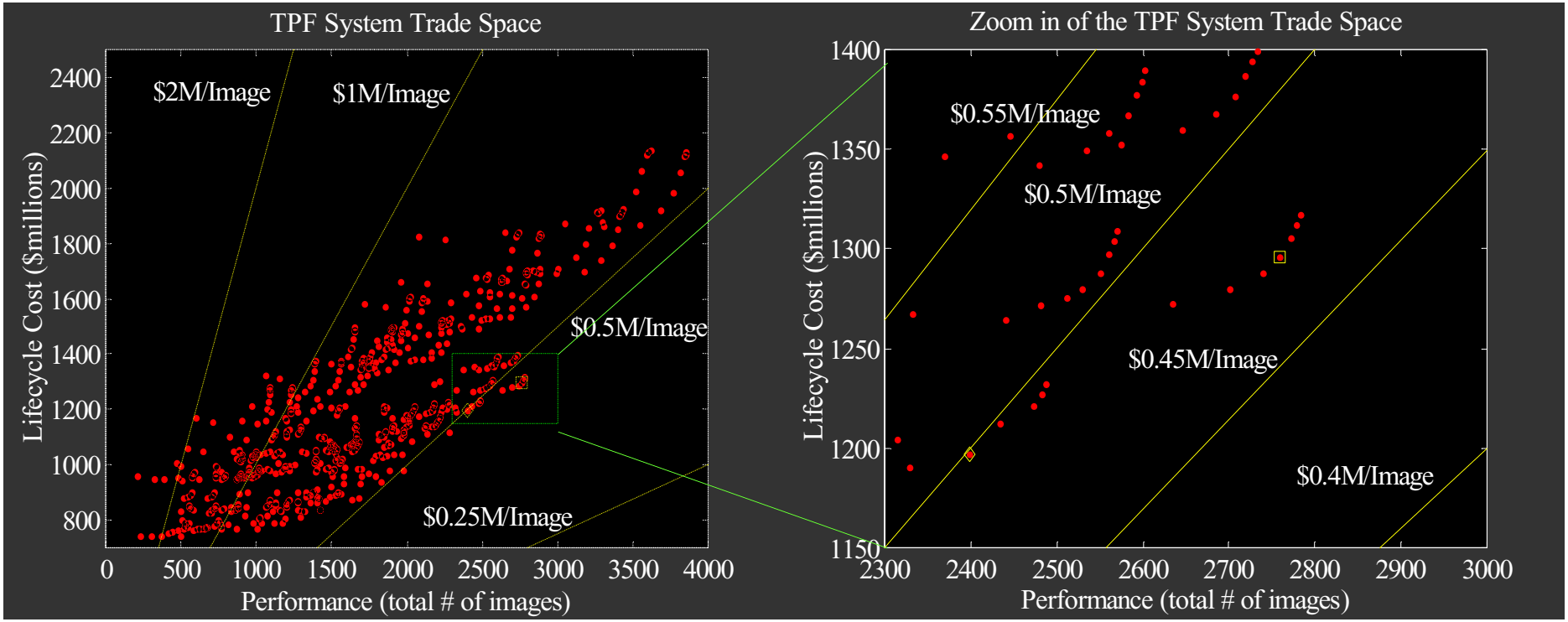
6) If $E(\Gamma_{i+1}) > E(\Gamma_i)$, then accept Γ_{i+1} as the new current solution with a probability $e^{(-\Delta/T)}$

where $\Delta = E(\Gamma_{i+1}) - E(\Gamma_i)$.

7) Reduce the system temperature according to the cooling schedule.

8) Terminate the algorithm.

SA Results on TPF Trade Space

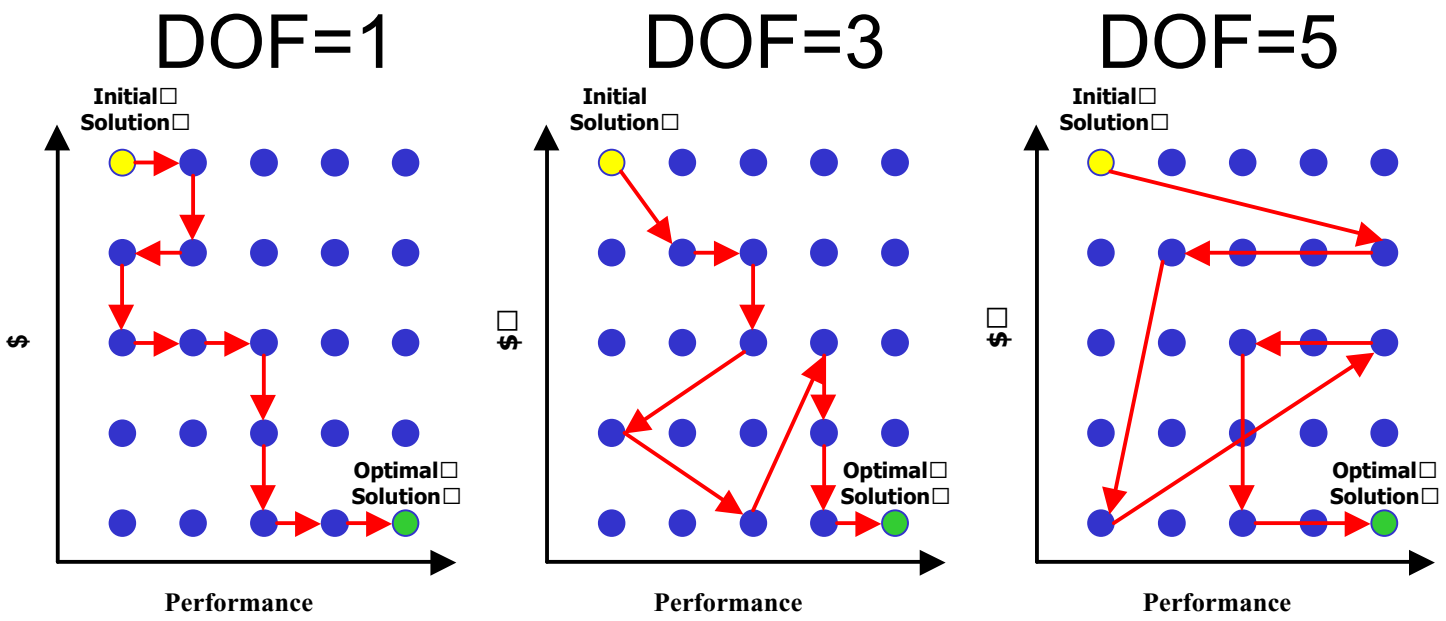


- SA Algorithm finds most cost effective families of design architectures after evaluating only a small fraction of the global trade space.
- Simulated Annealing has been successfully applied to very large combinatorial optimization problems where the total number of variables may range into the tens of thousands.
  - The Traveling Salesman Problem

Degrees of Freedom (DOF) and SA Performance

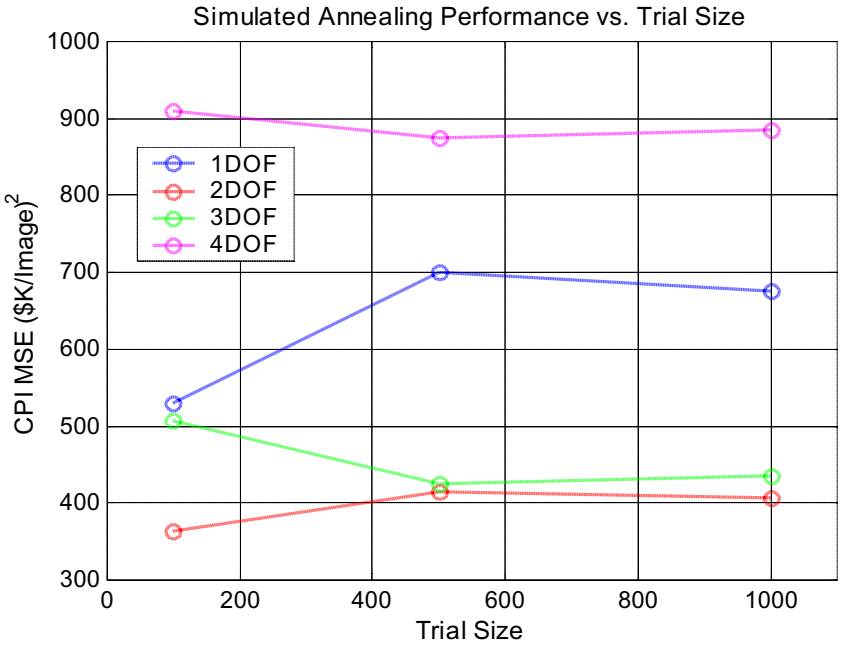
- **Objective:** "What DOF allows simulated annealing to search the DSS trade space most efficiently and why?"
- **Recall Neighborhood Definition:**

$$G_1 = [1\text{AU SCI-1D 4ap } 4\text{m}] \quad G_2 = [1\text{AU SCI-1D 4ap } 2\text{m}] \quad \text{DOF} = 1$$



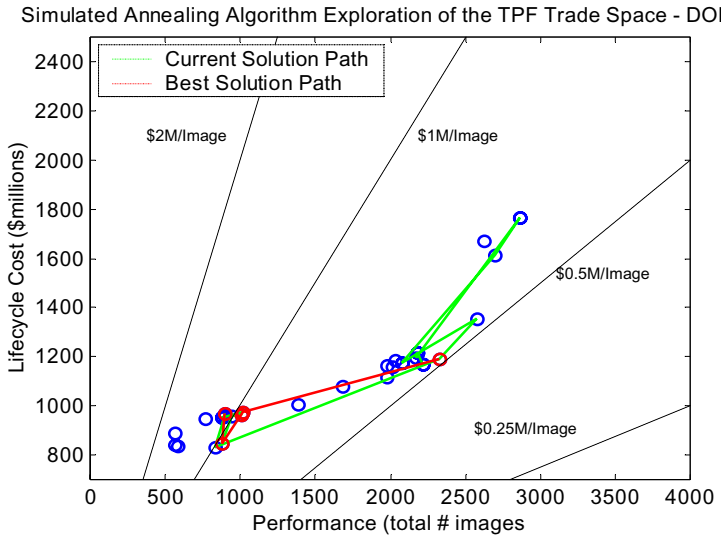
Degrees of Freedom (DOF) and SA Performance

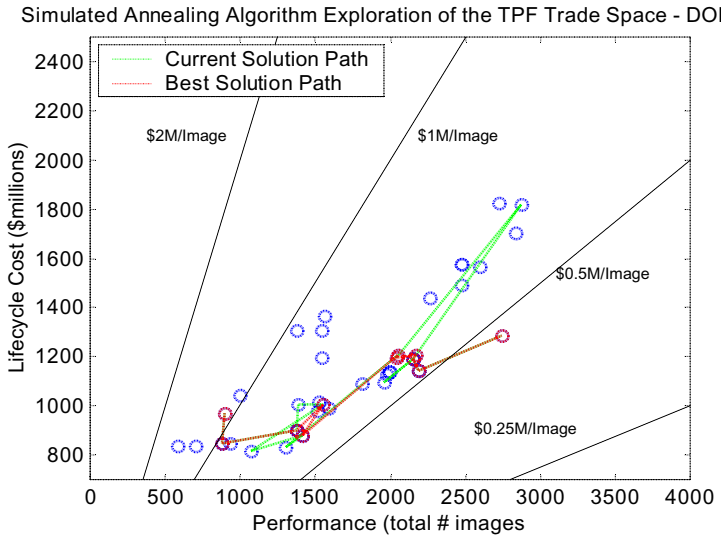
Performance of the Simulated Annealing Algorithm in Finding the Minimum CPI TPF Design Architecture as a Function of DOF

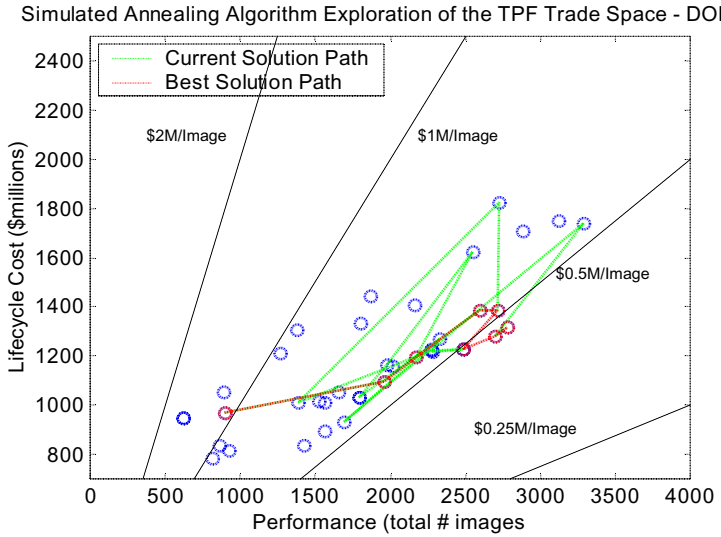


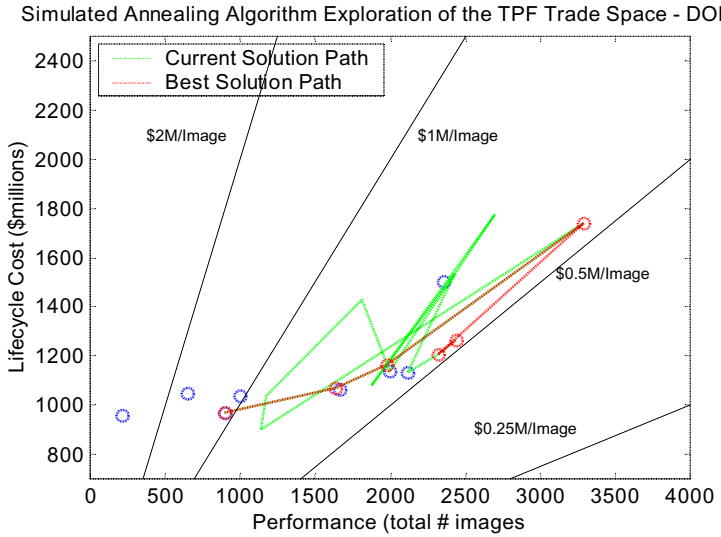
- **Observations:**
  - Setting DOF=2 appears to yield the most efficient exploration of the TPF trade space.
  - DOF=Min. → Overly constrictive neighborhood
  - DOF=Max. → Degenerates to a random search

Degrees of Freedom (DOF) and SA Performance









Simulated Annealing Lecture Summary

- Originated from statistical mechanics. Analogous to the cooling of a material to a state of minimum energy.
- 8-Step Algorithm
- Terrestrial Planet Finder (TPF) Mission Example
- There are many items one may tailor within the algorithm to affect its performance (initial system temp, cooling schedule, DOF within a neighborhood, etc.)
- The simulated annealing algorithm, like other heuristic techniques, is NOT guaranteed to find the global optimum; but it should find several good solutions.

References

- Brooks, R.R., Iyenger, S.S., and Rai, S, "Comparison of Genetic Algorithms and Simulated Annealing for Cost Minimization in a Multisensor System," *Optical Engineering*, Vol. 37(2), Feb. 1998, pp. 505-516.
- Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *J. Opt. Theory Appl.*, 45, 1, 41-51, 1985 .
- Jilla, C.D., Miller, D.W., and Sedwick, R.J., "Application of Multidisciplinary Design Optimization Techniques to Distributed Satellite Systems," *Journal of Spacecraft and Rockets*," Vol. 37, No.4, 2000, pp. 481-490.
- Jilla, C.D., and Miller, D.W., "Assessing the Performance of a Heuristic Simulated Annealing Algorithm for the Design of Distributed Satellite Systems," *Acta Astronautica*, Vol. 48, No. 5-12, 2001, pp. 529-543.
- **Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, Volume 220, Number 4598, 13 May 1983, pp. 671-680.**
- Metropolis,N., A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines", *J. Chem. Phys.*,21, 6, 1087-1092, 1953.
- Schulz, A.S., "Metaheuristics," 15.057 Systems Optimization Course Notes, MIT, 1999.
- http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/anneal/www/tech_reports.html
- http://www.taygeta.com/annealing/simanneal.html