

DECISION-THEORETIC SIMULATED ANNEALING

Todd W. Neller

Christopher J. La Pilla



OVERVIEW

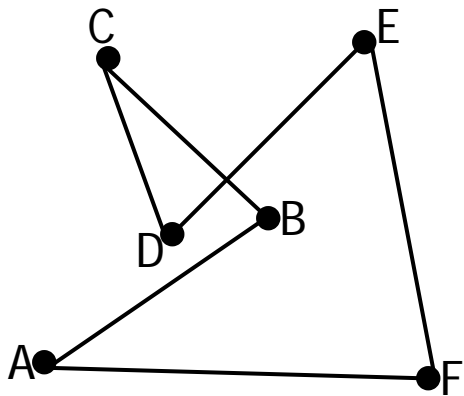
- Given: a computational process that requires dynamic tuning of parameter(s) throughout with utility tradeoffs between time and quality
- Takeaway: principled approach to reinforcement learning of automated tuning
- Outline:
 - Introduce simulated annealing (SA)
 - Formulate annealing control as MDP
 - Share applications of reinforcement learning

SIMULATED ANNEALING

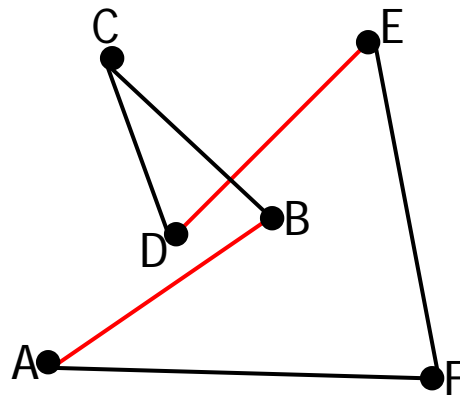
DESIGN DECISIONS

- Four design decisions with traveling salesman problem (TSP) examples:
 - State representation: sequence of cities
 - Next state generation function: reverse order of a sub-sequence in the cycle (next slide)
 - Energy (or objective) function: path cost (e.g. length) of cycle
 - Cooling or annealing schedule: ??? “black art”

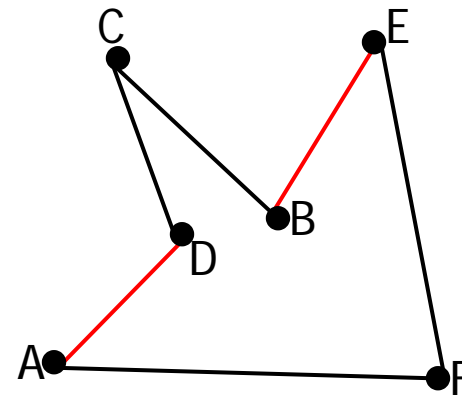
TSP NEXT STATE GENERATION



ABCDEF



ABCDEF



ADCBEF

SIMULATED ANNEALING ALGORITHM

- ⦿ Pick an initial state s .
- ⦿ While cooling (i.e. reducing) the temperature T according to a given schedule:
 - Generate a next state s' .
 - Compute $\Delta E = E(s') - E(s)$, i.e. the change in energy.
 - If $\Delta E < 0$, $s \leftarrow s'$, i.e. accept the next state.
 - Otherwise, accept the next state with probability $e^{-\Delta E/kT}$, where k is Boltzmann's constant.
- ⦿ Return state s^* that minimized E .

TSP ANNEALING DEMO

Simulated Annealing Applet

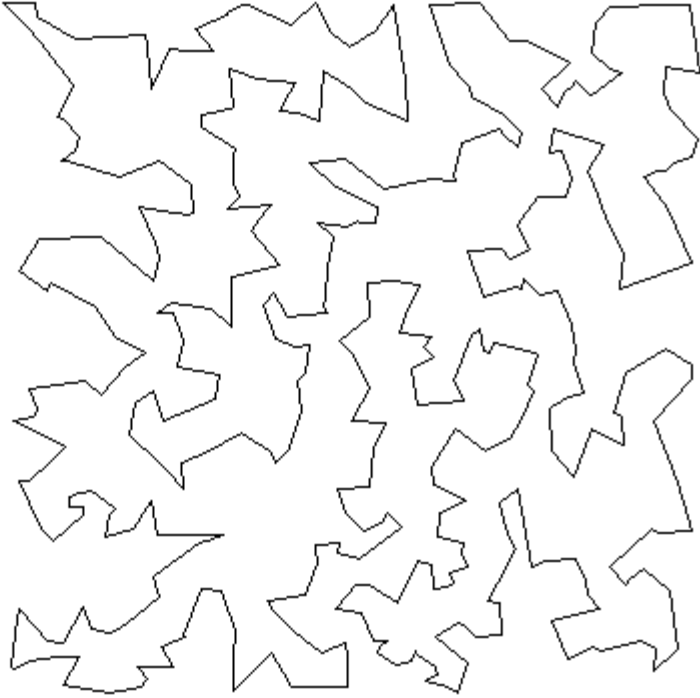
Simulated Annealing
TSP Demonstration
Todd Neller
Gettysburg College

Show State:
 Current
 Best

Anneal
 Clusters

New Problem

Length
5314
Temperature
1.8E-2



< ||| >

GSAT/WALKSAT VS. SA STRAWMAN

- SA used in industry, out of fashion in AI.
Why?
- Selman and Kautz used SA “strawman” for comparison in GSAT/WalkSAT work:
 - Primary contribution: next state generation for SAT
 - Next state for SA: random bit flip
 - No annealing/restarts in SA! Constant temperature for single run! → Weak strawman → discredits SA
- Meanwhile, SA still preferred technique for many difficult combinatorial optimization problems (e.g. VLSI layout, channel routing).

DECISION-THEORETIC SA

- ⦿ Empirical ad-hoc → principled approach.
When/Why?
 - Body of problems
 - Expectation of frequent usage
- ⦿ With each SA iteration:
 - Potential gain in solution quality
 - Definite loss of computational time

DTSA TERMINOLOGY AND NOTATION

- ⊙ Energy \neq Utility (but monotonic)
- ⊙ $U_o(s)$ - object-level (intrinsic) utility of state s
- ⊙ $U_i(t)$ - inference-level utility at time t
- ⊙ s_t^* - minimal energy state at time t
- ⊙ $U(s_t^*, t) = U_o(s_t^*) - U_i(t)$ - net utility at time t
- ⊙ Abstract SA as control problem with annealing *process* states, actions and (utility) rewards

PARAMETER-LEARNING AGENT

- States: single-state, n -armed bandit MDP
- Actions: choose m iterations for complete SA run
 - m is evenly-distributed logarithmically from 10^5 to 10^7 , inclusive
 - Start temperature 20 times standard deviation of random walk ΔE 's
 - End temperature - uphill step highly unlikely
 - Geometric annealing schedule computed from these
- Rewards: net utility for complete SA run

EXPERIMENTAL RESULTS

- ϵ -greedy action selection, three problems (TSP, clustered TSP, class scheduling), 16384 (2^{14}) learning trials
- Policies reported and evaluated for mean utility after 2^n trials for each n
- Significant learning tends to occur between 32 and 128 trials, with plateau before and after
- ϵ -greedy action selection simple, softmax leads to faster convergence but has its own tuning

ANNEALING CONTROL AGENT

- Finer grained dynamic control across SA run to take advantage of problem *specific heat*
- States: annealing temperatures
 - partition previous temperature range logarithmically into 10 subranges
- Actions: choose iterations through next temp. subrange, or *terminate*
- Rewards: change in net utility across subrange

RESULTS AND OBSERVATIONS

- SARSA learning
- Mean peak utility not significantly greater, but convergence significantly faster
- Many interesting possibilities for future exploration:
 - Additional state information: specific heat estimation, current state
 - Additional actions: Simulated quenching with restarts, simulated tempering

CONCLUSION

- ⦿ For serious, repeated problem-solving applications, apply reinforcement learning to move beyond ad-hoc “black arts” .
- ⦿ Main contributions:
 - Decision-theoretic formalization to put SA control MDP formulation on sound foundation
 - Demonstrative experiments with common combinatorial optimization problems