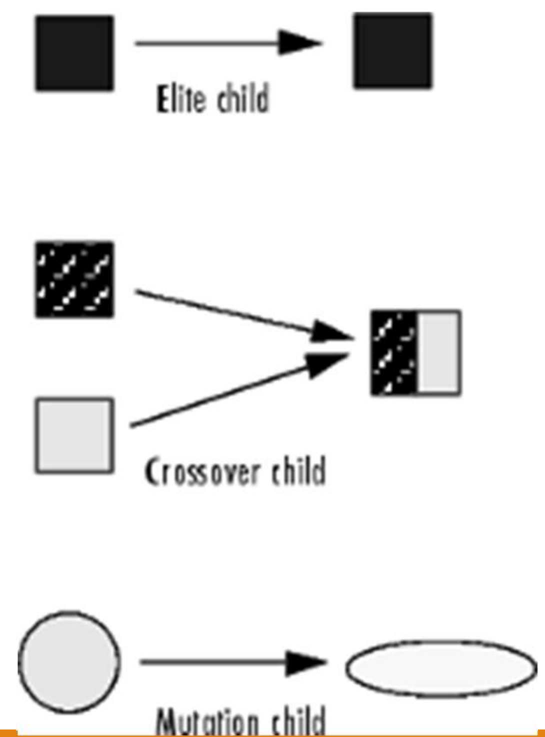
Advanced Algorithms Course Review & Project Reminders

PROFESSOR KALISKI

Generic Methods

Genetic Algorithms: How they work

1. Creates random initial population
2. Creates new population, based on current population
   1. Determine fitness of current population, scale to expectation values
   2. Select member, “parents”, based on expectation
   3. Select some lower fitness members and pass to next generation
   4. Produce next generation based on parents via mutation or crossover
   5. Replace current population with next generation
3. Repeat step 2 until stopping condition met



Genetic Algorithms: Tradeoffs

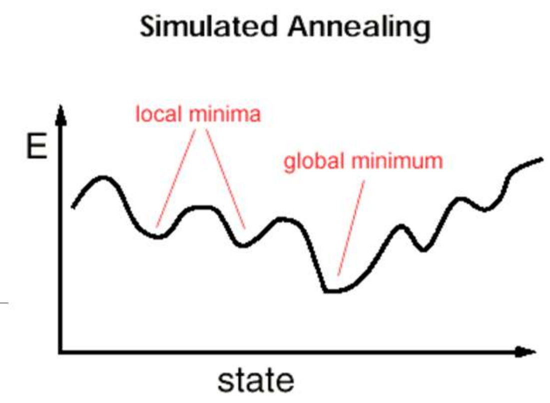
Pros:

1. Faster than traditional algorithm
2. Easier, if vector representation is possible (less analysis).

Cons:

1. May not find an optimum
2. Not a complete algorithm, i.e. may get stuck in local max/min.

Simulated Annealing: How it works



Method of solving optimization problems

- unconstrained and bound-constrained
- models physical process, heating up, then slowly cooling down (Decrease defects)

In each iteration of simulated annealing algorithm, new random point generated

- Distance of new point from current point depends on probability distribution with a scale proportion to the temperature
- Accepts all points below are lower than objective, but will accept new points higher than objective (with some probability) (avoid local minima)

Annealing schedule selects points which systematically decrease the temperature as the algorithm proceeds

- Search range decreases → converges to a minmum

Simulated Annealing Tradeoffs:

Pros:

Can handle arbitrary systems and cost functions

statistical guarantee of finding optimal solution

easy to implement (even for complex problems)

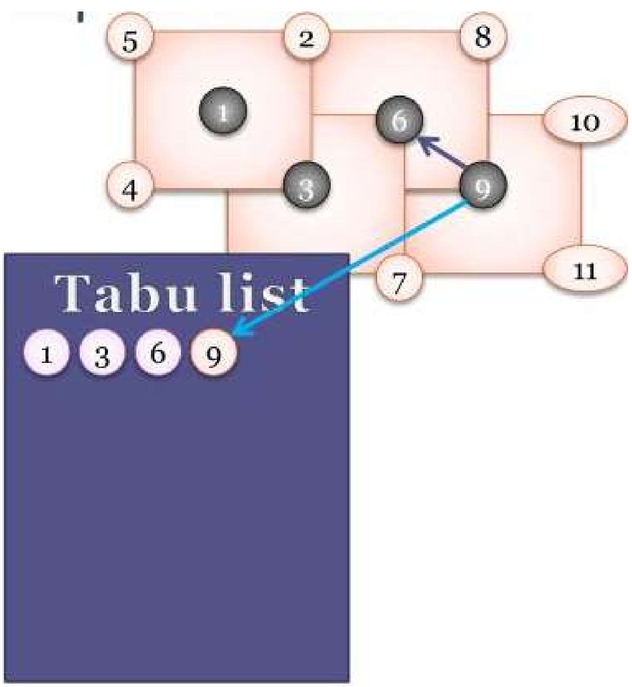
General gives a “good” solution

Cons

- Repeated annealing can be slow, 1/log k (cost function may be expensive to compute)
- May be overkill, if energy landscape is smooth (few minima)
- May not offer a better solution than system specific heuristics
- Unable to determine if optimal solution found

Tabu Search: How it works

- Start at some solution candidate
- Searches local / neighborhood for improved solution (includes non-improved solution) via steepest descent search.
- Stopping criteria or attempt limit / score threshold required
- Similar to simulated annealing (a special form of Tabu Search)



Tabu Search: Pros and Cons

Pros:

- Generates generally “good” solutions for optimization problems, compared to other AI methods

Cons:

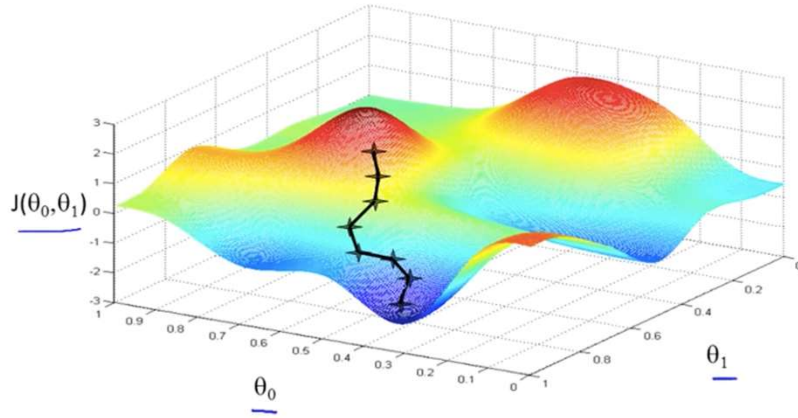
- List construction is problem specific
- No guarantee that global optimum will be reached

Gradient Descent: How it works

Popular in machine learning / deep learning

Consider a gradient / walking downhill to minimize a function

$$\mathbf{b} = \mathbf{a} - \gamma \nabla \mathbf{f}(\mathbf{a})$$



Gradient descent: Pros and Cons

Pro:

- Simple idea, iterations are “cheap”
- Very fast convergence for well-conditioned, strongly convex problems

Con:

- Can be very slow, due to difficulty in characterizing problem. Problem may not be convex nor well-conditioned
- Cannot handle nondifferentiated functions

NP complete / NP hard problems

Max-Cut Problem

Combinatorial optimization problem

- aims to find division of a vertex set into 2 parts. Goal: Maximize sum of weights across vertex subset

Applications

- Network design, Statistical Physics, VLSI, Circuit Layout design, data clustering

Partition Problem

Determining how to evenly split weights/number/etc among N sets

Applications:

- Load Balancing
- Parallel computing

Max-Sat Problem

Determine the maximum number of satisfiable caluses

Applications:

- Debugging Digital Logic / C code
- Course timetabling
- Combinatorial auctions
- Minimizing disclosure of Private Information in Credential-Based Interactions
- Software installation – Determine maximum number of installable packages
- Reasoning over Biological Networks
- Logic Minimization, Digital Filter Design, FSM synthesis
- ....

Boolean Satisfiability problem

Solves combinatorial assignment problems

Similar problem: 3-SAT

Applications

- crosstalk noise prediction in integrated circuits
- model checking of finite state systems
- design debugging
- AI planning
- software model checking / software testing
- circuit delay computation
- identification of functional dependencies
- ...

Max Clique problem

Determine maximum sized clique, could be weighted

Applications:

- Model Social Networks
- Bioinformatics
- Computational chemistry
- Motion segmentation

Decision version: Clique Decision Problem

List Scheduling Problem

Make an ordered list of processes by assigning them some priority, assign them resources

Application:

- Critical Path
- Project schedule
- Minimize Longest schedule

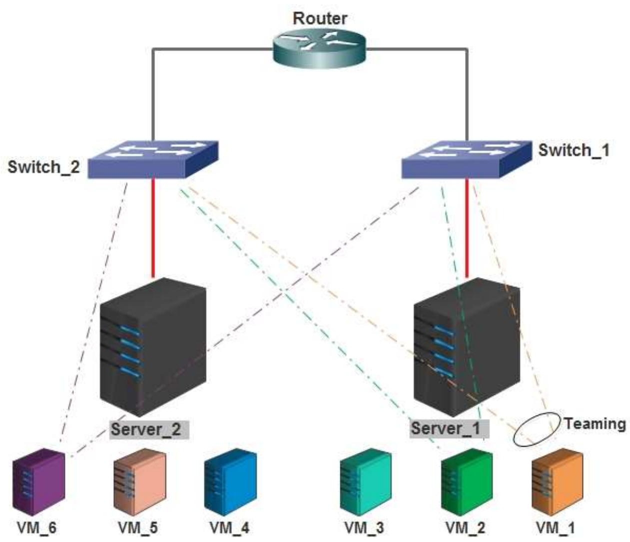
Note: Objective function changes schedule

Load Balancing Problem

Distributing set of tasks over a set of resources

Applications:

- Internet-based services
- Round-robin DNS
- DNS Delegation
- Client-Server Balancing
- Server-side load balancing
- Shortest Path Bridging (Mesh)
- Routing (Network Congestion)



Independent Set

Determine if an independent set / vertices of a certain size exists

Maximum Independent Set applications:

- combinatorial auctions
- graph coloring
- coding theory
- geometric tiling
- fault diagnosis
- pattern recognition
- scheduling
- computer vision
- .....

Vertex Cover

Determine if a vertex cover of a certain size exists

Applications:

- Dynamic Detection of race conditions (minimum lock set size required to be race-free.).
- Matching problems
- Optimization problems

Set Cover

Determine if a set of size N covers all

Applications:

- Searching for substrings (think computer virus) (IBM)
- Cost minimizing Procurement of materials (GM)
- Service Area, city planning

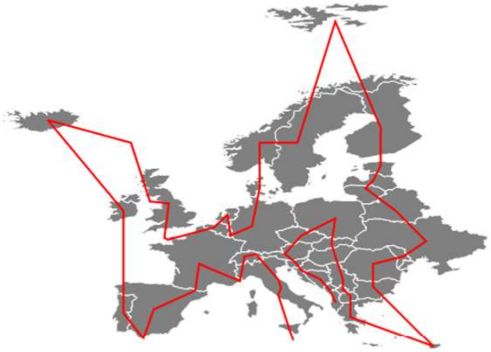
Set Packing

Determine largest number of mutually disjoint sets

Applications:

- Radio resource application
- Sphere packing
- Knapsack

Traveling Salesman problem (TSP)



Applications:

- Drilling printed circuit boards (save time), change heads, drill holes
- Overhauling gas turbine engines (aircraft), uniform gas flow
- X-ray crystallography, position time of sensor (100K+ positions)
- Computer Wiring (JTAG test bus or circuit testing)
- Order-picking problem in warehouse (think Amazon warehouse)
- vehicle routing (mailbox, mail delivery)
- Mask plotting in PCB production (photo resist and PCB etching)

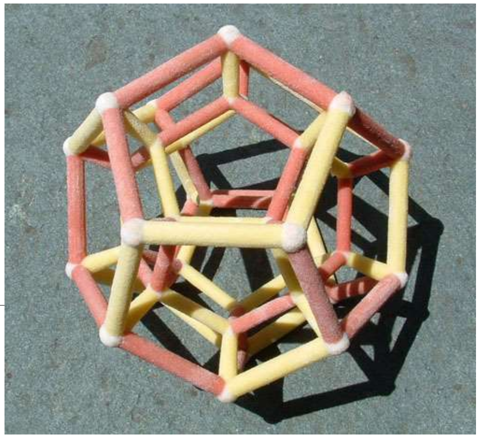
Multi Traveling Salesman Problem (mTSP)

Multiple TSP

Applications:

- Printing press scheduling problem: which form will run, how long, changing plates cost inter-city transit costs
- School bus routing problem: Obtain a bus loading pattern that minimizes the number of routes (time, loading, distance)
- Crew scheduling Problem: deposits from center banks to central office, minimize cost
- Interview scheduling problem: Tour brokers and vendors of tourism industry (T cities)
- ...

Hamiltonian cycle



Recall: A Hamiltonian cycle visits every vertices once, but ends where it start

Applications:

- Time scheduling
- Choice of travel routes
- network topology
- ....

3-Dimensional Matching Problem

Matching based on 3 constraints / set of resources

Application:

- Tri-Partite Matching
- Set Packing

Graph Coloring

Assign colors to graph based on certain constraints, such as no matching colors

Applications:

- Exam schedule (Multiple subjects, many students)
- Mobile Radio frequency assignment
- Sudoku (Every Cell represents vertex, edges between two verticies if in same row, same col, same block
- Register Allocation (Compiler)
- Map Coloring
- Bipartite Graph check

Subset Sum

Problem: Find a subset of integers equal to sum

Application:

- Encryption
- Message Verification (Hash)
- Knapsack
- Factoring

K-center problem

Recall: Find locations of k subsets with minimal distance between subset center C and associated vertices

Applications:

- Facility location
- Data clustering

Dominating Set

Find dominating set where each vertex is in the set or a neighbor

Applications:

- Minimize Exchange points (routing) in a network
- Security (location)
- Medicine (Smallest set of drugs to treat a condition)
- Special resource allocation (fire stations, advertisements)

Minimum Degree Spanning Tree

Find a minimum degree spanning tree

Application:

- Telephone
- Electrical
- Hydraulic
- TV cable
- Computer networks
- Roads

Bin-packing problem

Items of different volumes must be packed into a finite number of bins (fixed volume bins), goal is to minimize number of bins

Applications:

- Filling up containers
- Loading trucks with weight capacity constraints
- creating file backups in media
- mapping in FPGAs

Global Minimum Cut

Determine Minimal cut in a graph

Applications:

- Equivalent to Max flow problem
- Network connectivity/reliability
- Sets of related hypertext documents
- Parallel languages / distributed memory
- Large scale combinatorial optimization, use cuts to determine TSP

Steiner Tree & Forest

Tree of minimum weight, connects all terminals (may have additional vertices)

Applications:

- Network Design
- Protocol Design
- Circuit Layout
- VLSI / Wire routing
- EDA
- water networks

Network Flow

Determine Max Flow / Min Cut

Applications:

- Network reliability
- Data mining
- Project Selection
- Image segmentation
- Network connectivity
- Bipartite matching

Spanning Tree

Minimum Spanning Tree: Find a min weight path connecting all nodes to the root

Application:

- Network Design
- Max Bottleneck Paths
- Cluster Analysis
- Routing

Additional Algorithms

Hashing, Hash Tables, Universal Hash, Hash Maps

Hashing: Uses a mathematical function to map a key to a location

Universal Hashing avoids collisions

Applications:

- Dictionary
- Caching (Bloom Filters, Cuckoo Filters)
- Content Addressable Memories
- Look Up Tables
- Checksum

Approximation & Randomized Algorithms

Approximation Algorithms: Greedy Methods, Local Search

- What is a-Approximation, 1+e / 1-e PTAS?
- Scheduling jobs on a single machine, Scheduling job on identical parallel machines
- K-center problem
- Dominating set
- Load Balancing
- TSP, Metric TSP, Hamiltonian
- Finding Minimum Degree spanning tree
- Knapsack, Bin Packing Problem

Randomized Algorithms

- What are Randomized Algorithms used for?
  - Las Vegas versus Monte Carlo
- Three-Card Monte
- Verifying polynomial identities
- Contention resolution in distributed systems
- Contraction Algorithm → Finding Global Min Cut
- Guessing Cards
- Max 3-SAT
- Use of pivots/select for quicksort and median-finding

Randomized Algorithm

- Hashing, Hash Tables, Universal Hashing
- Finding closest points
- Caching, Cache Marking
- Hash Maps → Dictionary
- Bloom Filters, Counting Bloom Filter (Distributed Caching), Cuckoo Hashing
- Chernoff bounds, load balancing

Chernoff Bounds

Definition: Probability that a ransom variable deviates from its expected value

Applications:

- Coin Tossing
- Measure of robustness of algorithm (slight perturbations in input)
- Set Balancing (Balancing features among groups)
- Permutation routing / Reduction of network congestion
- Machine Learning (probably approximately correct algorithm)

More Approximation Algorithms

Knapsack Problem, Multi-Choice Multi-Dimension

Linear Programming Approximation Algorithms

- Polynomial Time Approximation
- Weighted Vertex Cover
- Matchings
- Approximation Ratio

Branch and Bound

Nash Equilibrium

Recall Nash Equilibrium is a stable assignment of resources to users

Applications: Auctions, bankruptcy, cost sharing, division of resources, fairness, ice cream sharing, pie splitting, coin flipping, minimax decision trees (think computer games), ...

Lab Talk

Application of Approximation algorithms

- Resource allocation for multicast systems
- Advertisement slot allocation for Device-to-Device (D2D) in cellular systems (LTE)
- Advertisement slot allocation for Mobile TV

Grading

- Final project presentation: 25%
- Final project report: 25%
- Final project results: 20%

Final Project Guidelines

Reminder: Project Report, presentation, and source files due on June 19, 2020 (upload to Moodle)

Report Format:

- Length <= 10 pages, single column, single space, 12pt
- Source code does not count towards page count
- Include student IDs and group member names

Final Project Guidelines

Items to include:

- Background (clear) and References
- Problem Definition
- Proposed Solution (include source code)
- Analysis (Complexity and Correctness)
- Comparison to other methods/ Evaluation

Final Presentation

- Presentation duration: 15 minutes + 5 minutes Q & A
- Presentation should be clear, concise, and informative
- Presentation should include demo and comparison to other methods

Tips for a good / effective presentation

- Limit bulleted items to around 5 per page
- Use different size fonts / colors to emphasize
- Make sure presentation has smooth flow and is clear + convincing
- Emphasize how your problem and solution relate to randomized and/or approximation algorithms
- Practice, practice, practice ...