Stochastic Local Search Algorithms

Alan Mackworth

UBC CS 322 – CSP 7

February 8, 2013

Textbook §4.8

Lecture Overview

- Announcements
- Recap: stochastic local search (SLS)
- Types of SLS algorithms
- Algorithm configuration
- AI in the news: Watson and Siri

Announcements & Reminders

- Assignment 2 is due next Friday, February 15, 1:00pm
  – Don't leave it to the last minute
- Reminder: midterm is Wednesday, March 6, 3:00 – 3:50pm
  – Set of short questions to be provided: subset on midterm.
- Final exam is scheduled for Thursday, April 18, 8:30am
  – Will schedule extra review session(s) after classes end, before exam

Practice exercises

- Who has used them?

- Try Exercise 5 for SLS practice in AIspace.

Lecture Overview

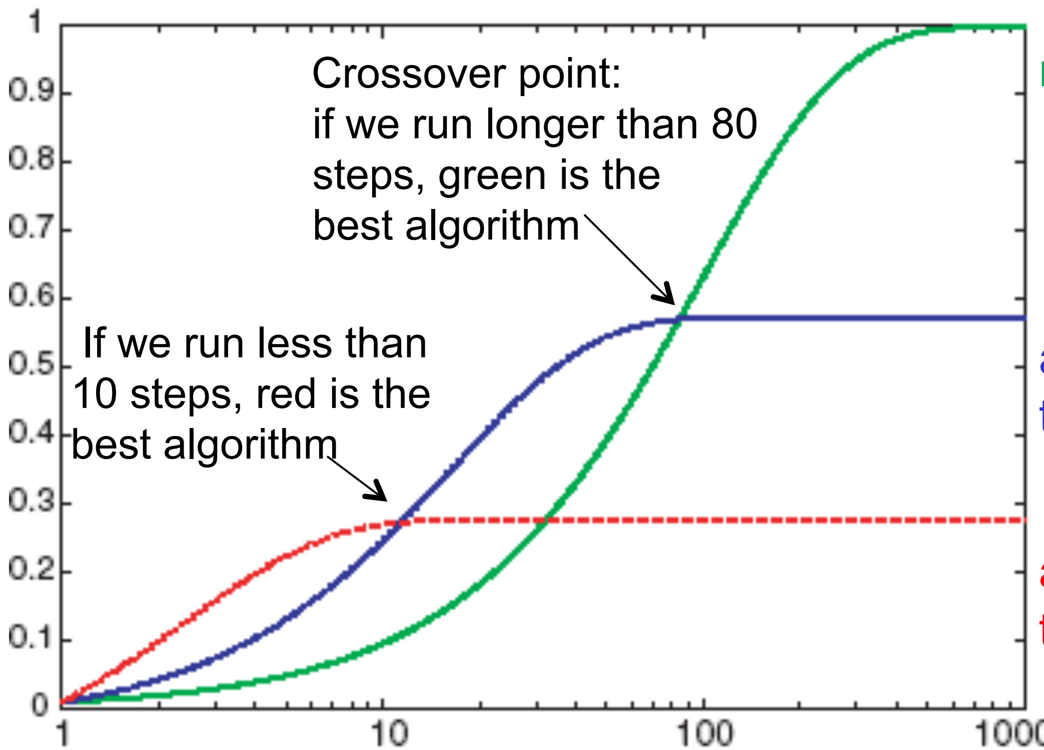
- Announcements
- Recap: stochastic local search (SLS)
- Types of SLS algorithms
- Algorithm configuration
- AI in the news: Watson and Siri

Comparing runtime distributions

- SLS algorithms are randomized
  - The time taken until they solve a problem is a random variable
- Runtime distributions
  - x axis: runtime (or number of steps, typically log scale)
  - y axis: proportion (or number) of runs solved in that runtime

Fraction of solved runs, i.e.

P(solved by this time)



# of steps

Slow, but does not stagnate

57% solved after 80 steps, then stagnate

28% solved after 10 steps, then stagnate

Pro's and Con's of SLS

- Typically no guarantee to find a solution even if one exists
  - Most SLS algorithms can sometimes stagnate
    - Not clear whether problem is infeasible or the algorithm stagnates
    - Very hard to analyze theoretically
  - Some exceptions: guaranteed to find global minimum as time $\to \infty$
    - In particular random sampling and random walk:
      strictly positive probability of making N lucky choices in a row
- Anytime algorithms
  - maintain the node with best h found so far (the "incumbent")
  - given more time, can improve their incumbent
- Generality: can optimize arbitrary functions with n inputs
  - Example: constraint optimization
  - Example: RNA secondary structure design
- Generality: dynamically changing problems

SLS generality: Constraint Optimization Problems

- Constraint Satisfaction Problems
  - Hard constraints: need to satisfy all of them
  - All models are equally good

- Constraint Optimization Problems
  - Hard constraints: need to satisfy all of them
  - Soft constraints: need to satisfy them as well as possible
  - Can have weighted constraints
    - Minimize h(n) = sum of weights of constraints unsatisfied in n
    - Hard constraints have a very large weight
    - Some soft constraints can be more important than other soft constraints → larger weight
  - All local search methods we will discuss work just as well for constraint optimization
    - all they need is an evaluation function h

Example for constraint optimization problem

Exam scheduling

- Hard constraints:
  - Cannot have an exam in too small a room
  - Cannot have multiple exams in the same room in the same time slot
  - …
- Soft constraints
  - Student should not have to write two exams at the same time (important)
  - Students should not have multiple exams on the same day
  - It would be nice if students had their exams spread out
  - …

SLS generality: optimization of arbitrary functions

- SLS is even more general
  - SLS's generality doesn't stop at constraint optimization
  - We can optimize arbitrary functions $f(x_1, \ldots, x_n)$ that we can evaluate for any complete assignment of their n inputs
  - The function's inputs correspond to our possible worlds, i.e. to the SLS search states

- Example: RNA secondary structure design

Example: SLS for RNA secondary structure design

- RNA strand made up of four bases: cytosine (C), guanine (G), adenine (A), and uracil (U)
- 2D/3D structure RNA strand folds into is important for its function
- Predicting structure for a strand is "easy": $O(n^3)$
- But what if we want a strand that folds into a certain structure?
  - Local search over strands
    - Search for one that folds into the right structure
  - Evaluation function for a strand
    - Run $O(n^3)$ prediction algorithm
    - Evaluate how different the result is from our target structure
    - Only defined implicitly, but can be evaluated by running the prediction algorithm

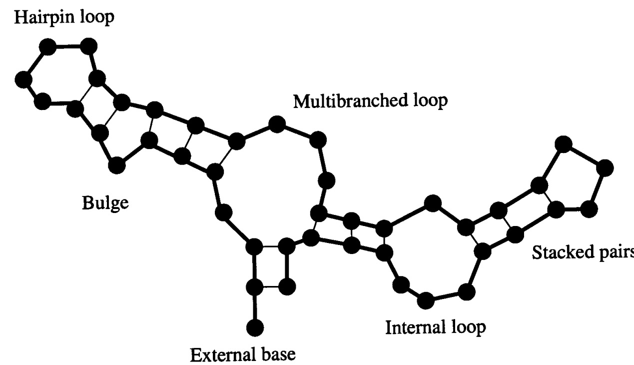
RNA strand

GUCCCAUAGGAUGUCCCAUAGGA

Easy

Hard

Secondary structure



Best algorithm to date: Local search algorithm RNA-SSD developed at UBC [Andronescu, Fejes, Hutter, Condon, and Hoos, Journal of Molecular Biology, 2004]

SLS generality: dynamically changing problems

- The problem may change over time
  - Particularly important in scheduling
  - E.g., schedule for airline:
    - Thousands of flights and thousands of personnel assignments
    - A storm can render the schedule infeasible

- Goal: Repair the schedule with minimum number of changes
  - Often easy for SLS starting from the current schedule
  - Other techniques usually:
    - Require more time
    - Might find solution requiring many more changes

Lecture Overview

- Announcements
- Recap: stochastic local search (SLS)
- Types of SLS algorithms
- Algorithm configuration
- AI in the news: Watson and Siri

Many different types of local search

- There are many different SLS algorithms
  - Each could easily be a lecture by itself
  - We will only touch on each of them very briefly
  - Only need to know them on a high level
- For more details, see
  - UBC CS grad course "Empirical Algorithmics" by Holger Hoos
  - Book "Stochastic Local Search: Foundations and Applications" by Holger Hoos & Thomas Stützle, 2004 (in reading room)

Simulated Annealing

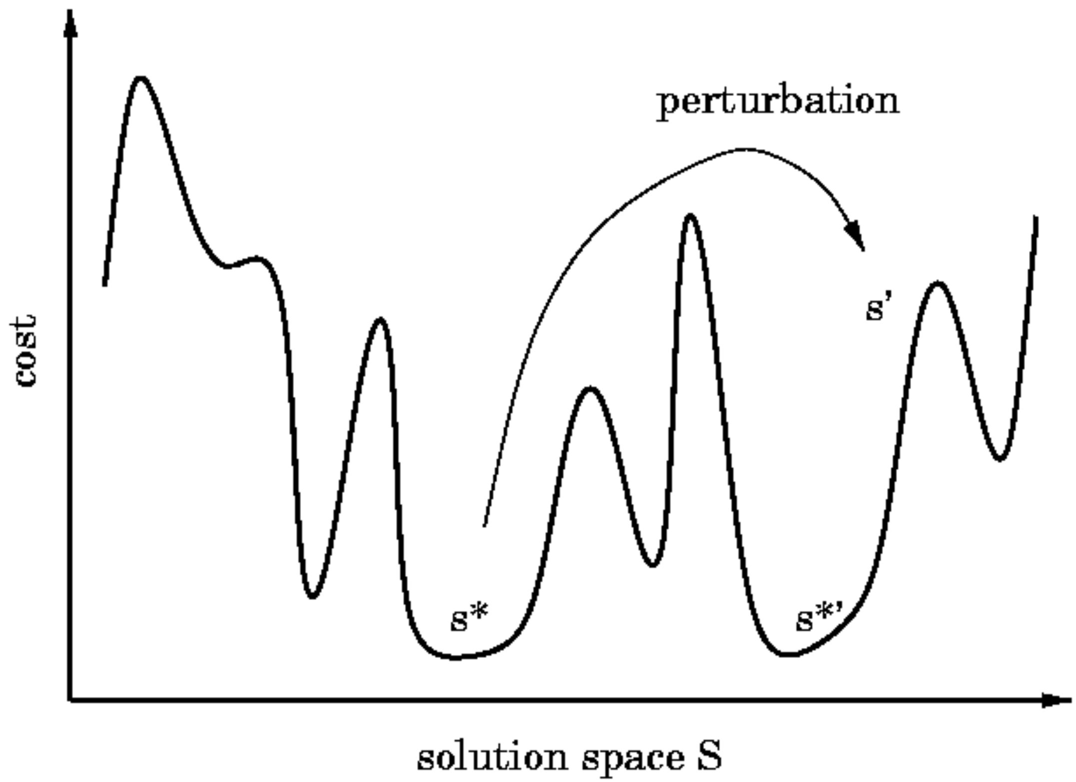
- Annealing: a metallurgical process where metals are hardened by being slowly cooled so settle into lowest energy state
- Analogy:
  - start with a high ‘temperature’: great tendency to take random steps
  - Over time, cool down: only take random steps that are not too bad
- Details:
  - At node n, select a random neighbour n’
  - If h(n’) < h(n), move to n’ (i.e. accept all improving steps)
  - Otherwise, adopt it with a probability depending on
    - How much worse n’ is than n
    - the current temperature T: high T tends to accept even very bad moves
    - Probability of accepting worsening move: $\exp((h(n) - h(n')) / T)$
  - Temperature reduces over time, according to an annealing schedule
    - “Finding a good annealing schedule is an art”
    - E.g. geometric cooling: every step multiply T by some constant < 1

Tabu Search

- Mark partial assignments as tabu (‘taboo’= forbidden)
  - Prevents repeatedly visiting the same (or similar) local minima
  - Maintain a queue of k variable=value assignments that are tabu
  - E.g., when changing V_7’ s value from 2 to 4, we cannot change V_7 back to 2 for the next k steps
  - k is a parameter that needs to be optimized empirically

Iterated Local Search

- Perform iterative best improvement to get to local minimum
- Perform perturbation step to get to different parts of the search space
  - E.g. a series of random steps (random walk)
  - Or a short tabu search



Beam Search

- Keep not just 1 assignment, but k assignments at once
  - A ‘beam’ with k different assignments (k is the ‘beam width’)
- The neighbourhood is the union of the k neighbourhoods
  - At each step, keep only the k best neighbours
  - Never backtrack
- When k=1, this is identical to:

Greedy descent	Breadth first search	Best first search

  - Single node, always move to best neighbour: greedy descent
- When k=∞, this is basically:

Greedy descent	Breadth first search	Best first search

  - At step m, the beam contains all nodes m steps away from the start node
  - Like breadth first search, but expanding a whole level of the search tree at once
- The value of k lets us limit space and parallelism

Stochastic Beam Search

- Like beam search, but you probabilistically choose the k nodes at the next step (‘generation’)

- The probability that neighbour n is chosen depends on h(n)
  - Neighbours with low h(n) are chosen more frequently
  - E.g. rank-based: node n with lowest h(n) has highest probability
    - probability only depends on the order, not the exact differences in h
  - This maintains diversity amongst the nodes

- Biological metaphor:
  - like asexual reproduction:
    each node gives its mutations and the fittest ones survive

Genetic Algorithms

- Like stochastic beam search, but pairs of nodes are combined to create the offspring

- For each generation:
  - Choose pairs of nodes n_1 and n_2 ('parents'), where nodes with low h(n) are more likely to be chosen from the population
  - For each pair (n_1, n_2), perform a cross-over: create offspring combining parts of their parents
  - Mutate some values for each offspring
  - Select from previous population and all offspring which nodes to keep in the population

Example for Crossover Operator

- Given two nodes:

  $X_1 = a_1,\ X_2 = a_2,\ \ldots,\ X_m = a_m$

  $X_1 = b_1;\ X_2 = b_2,\ \ldots,\ X_m = b_m$

- Select i at random, form two offspring:

  $X_1 = a_1,\ X_2 = a_2,\ \ldots, X_i = a_i,\ X_{i+1} = b_{i+1},\ \ldots,\ X_m = b_m$

  $X_1 = b_1,\ X_2 = b_2,\ \ldots, X_i = b_i,\ X_{i+1} = a_{i+1},\ \ldots,\ X_m = a_m$

- Many different crossover operators are possible

- Genetic algorithms is a large research field
  - Appealing biological metaphor
  - Several conferences are devoted to the topic

Lecture Overview

- Announcements
- Recap: stochastic local search (SLS)
- Types of SLS algorithms
- Algorithm configuration
- AI in the news: IBM Watson

Parameters in stochastic local search

- Simple SLS
  - Neighbourhoods, variable and value selection heuristics, percentages of random steps, restart probability
- Tabu Search
  - Tabu length (or interval for randomized tabu length)
- Iterated Local Search
  - Perturbation types, acceptance criteria
- Genetic algorithms
  - Population size, mating scheme, cross-over operator, mutation rate
- Hybridizations of algorithms: many more parameters

The Algorithm Configuration Problem

Definition

- Given:
  - Runnable algorithm A, its parameters and their domains
  - Benchmark set of instances B
  - Performance metric m
- Find:
  - Parameter setting ('configuration') of A optimizing m on B

UBC Ph.D. thesis (Hutter, 2009): "Automated configuration of algorithms for solving hard computational problems"

Motivation for automated algorithm configuration

Customize versatile algorithms
for different application domains

- Fully automated
  - Saves valuable human time
  - Can improve performance dramatically


Solver config 1


Solver config 2

Generality of Algorithm Configuration

Arbitrary problems, e.g.

- SAT, MIP, Timetabling, Probabilistic Reasoning, Protein Folding, AI Planning, ….

Arbitrary parameterized algorithms, e.g.

- Local search
  - Neighbourhoods, restarts, perturbation types, tabu length, etc
- Genetic algorithms & evolutionary strategies
  - Population size, mating scheme, crossover operators, mutation rate, hybridizations, etc
- Systematic tree search
  (advanced versions of arc consistency + domain splitting)
  - Branching heuristics, no-good learning, restart strategy, pre-processing, etc

Simple Manual Approach for Configuration

Start with some configuration

repeat

Modify a single parameter

if *results on benchmark set improve* **then**

keep new configuration

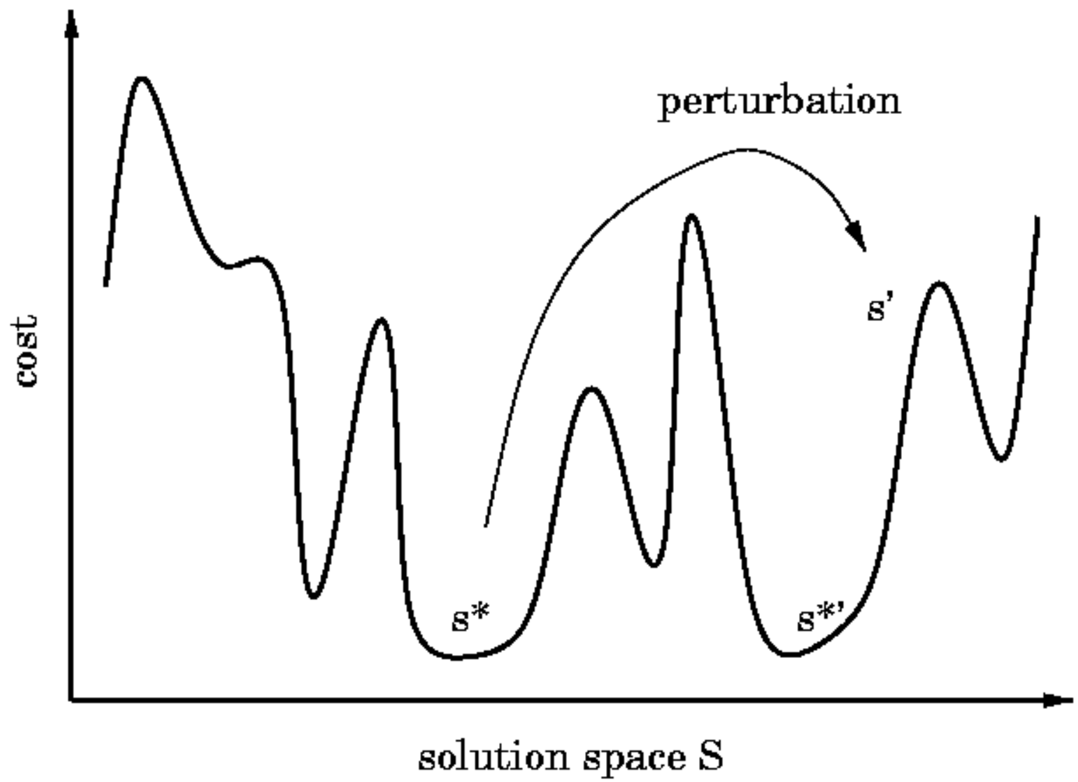
until *no more improvement possible (or "good enough")*

→ *Manually executed local search*

The ParamILS Framework

[Hutter, Hoos & Stützle; AAAI '07 & Hutter, Hoos, Leyton-Brown & Stützle; JAIR'09]

Iterated Local Search in parameter configuration space:



Example application for ParamILS: solver for mixed integer programming (MIP)

IP: NP-hard constraint optimization problem

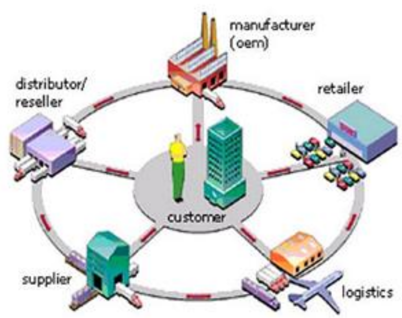
$$\begin{aligned} \min \quad & c^{\mathsf{T}} x \\ \text{s. t.} \quad & Ax \le b \\ & x_i \in \mathbb{Z} \text{ for i} \in \text{I} \end{aligned}$$

MIP = IP with only some integer variables

Commercial state-of-the-art MIP solver IBM ILOG CPLEX:

– licensed by > 1000 universities and 1300 corporations, including ⅓ of the Global 500

Transportation/Logistics:
SNCF, United Airlines, UPS, United States Postal Service, …



Supply chain management software:
Oracle, SAP, …

Production planning and optimization:
Airbus, Dell, Porsche, Thyssen Krupp, Toyota, Nissan, …

Up to 50-fold speedups just by optimizing the parameters!

Learning Goals for local search

- Implement local search for a CSP.
  - Implement different ways to generate neighbors
  - Implement scoring functions to solve a CSP by local search through either greedy descent or hill-climbing.
- Implement SLS with
  - random steps (1-step, 2-step versions)
  - random restart
- Compare SLS algorithms with runtime distributions
- Understand principles of types of SLS algorithms

Lecture Overview

- Announcements
- Recap: stochastic local search (SLS)
- Types of SLS algorithms
- Algorithm configuration
- AI in the news: Watson and Siri

IBM’s Watson

- Automated AI system participated in real *Jeopardy!*
  - Won practice round against two all-time Jeopardy champions
  - 3-day match on air February 14-16, 2011

- Jeopardy website with videos: http://www.jeopardy.com/minisites/watson/

- NYTimes article: “What Is I.B.M.’s Watson?
  http://www.nytimes.com/2010/06/20/magazine/20Computer-t.html

- Wired magazine:
  “IBM’s Watson Supercomputer Wins Practice Jeopardy Round”
  http://www.wired.com/epicenter/2011/01/ibm-watson-jeopardy/#

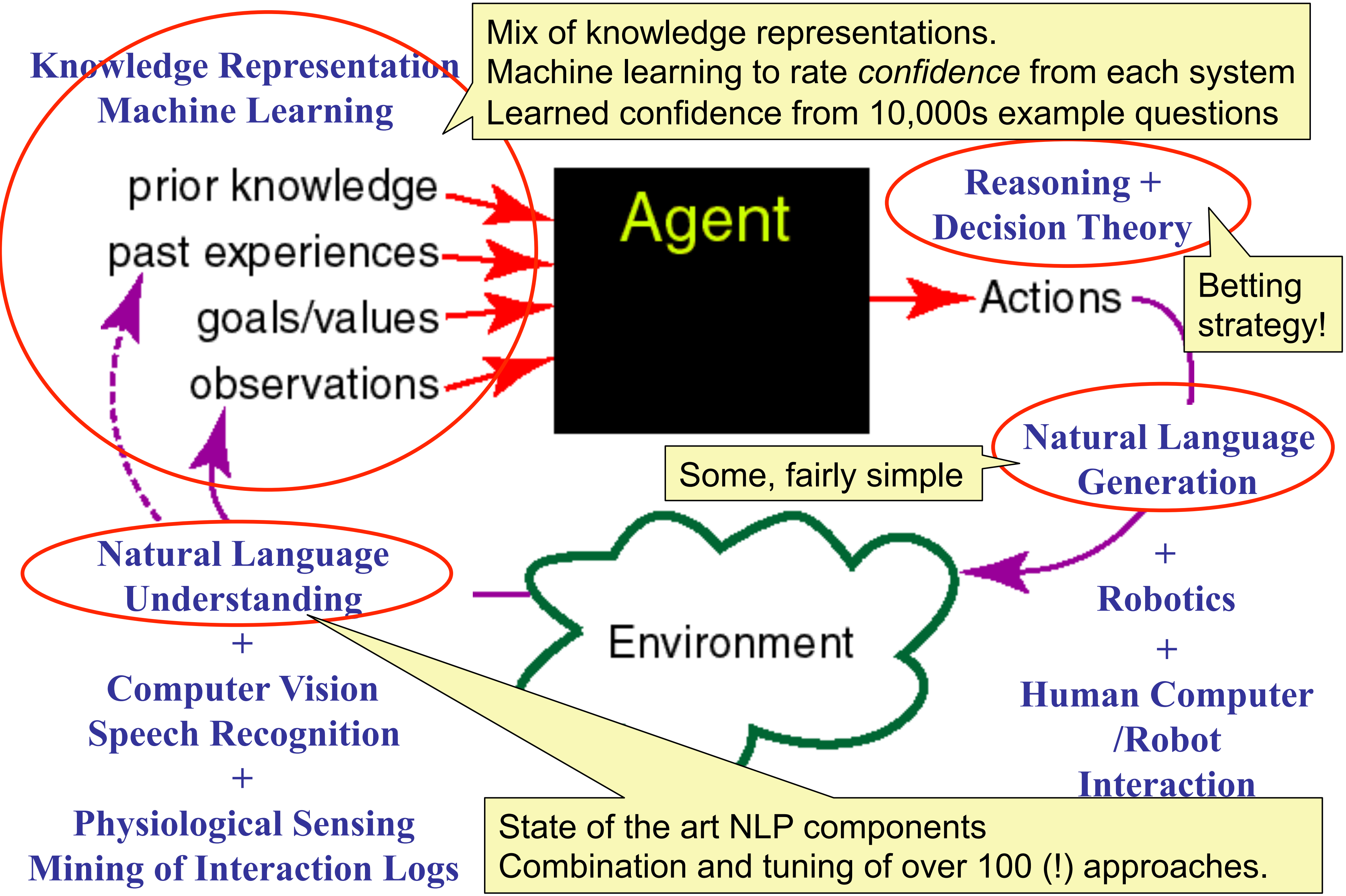
- More technical: AI magazine
  “Building Watson: An Overview of the DeepQA Project”
  http://www.stanford.edu/class/cs124/AIMagzine-DeepQA.pdf

IBM’s Watson: some videos

- “IBM and the Jeopardy Challenge”: http://www.youtube.com/watch?v=FC3IryWr4c8
- “IBM's Supercomputer Beats Miles O'Brien at Jeopardy”: http://www.youtube.com/watch?v=otBeCmpEKTs
- Video of practice round: http://www.engadget.com/2011/01/13/ibms-watson-supercomputer-destroys-all-humans-in-jeopardy-pract/
  - Watson won against Jeopardy champions Ken Jennings and Brad Rutter (by a small margin)
  - Including interview describing some of the underlying AI
    - But if you’re really interested, see the AI magazine article
- "Doctor" Watson To Inform Medical Decisions

http://www.medicalnewstoday.com/articles/234253.php

Watson as an intelligent agent (see lecture 1)



Apple's Siri

- Original SIRI is an offshoot of the DARPA-funded project, CALO, based at SRI. It was part of DARPA's PAL initiative (Personalized Assistant that Learns).
- http://www.apple.com/iphone/features/siri.html
- Interact with the calendar.
- Search contacts.
- Read and write messages (text and email).
- Interact with the Maps app and location services.
- Forward search phrases to certain pre-defined data providers (Yahoo! Weather, Yahoo! Finance, Yelp, Wolfram|Alpha, or Wikipedia).
- Dick Tracy's watch next ….