Properties

- Strategies that "guide" the search process
- Efficiently explore the search space to generate near-optimal solutions
- Simple local search — complex learning process
- Usually non-deterministic & "approximate"
- Have mechanisms to avoid getting trapped in "local optimal" or "other" areas.
- Basic concepts permit an abstract level description.
- Are not "problem-specific"
- Use "domain-level" or "domain-specific" knowledge (heuristics) that are controlled by the upper level strategy.
- Use search experience (stored in memory) to guide the search.

Ant Colony Optimization (ACO)
Evolutionary Computation (including Genetic Algorithms)
Simulated Annealing (SA)
Tabu Search (TS)
Iterated Local Search (ILS)
DIVERSIFICATION: Exploration of the search space

INTENSIFICATION: Exploitation of the accumulated search experience.

OTHER

EXPLOITATION \& EXPLORATION
Short term strategies tied to randomness

INTENSIFICATION \& DIVERSIFICATION
Medium + long term strategies based on the usage of memory.

IDEA: IDENTIFY + SEARCH GOOD REGIONS
AVOID OR SPEND LITTLE TIME ON BAD REGIONS

ESCAPE LOCAL MINIMA: EXPLORE THE SOLUTION SPACE +
MOVE TO BETTER PLACES (WITH BETTER LOCAL MINIMA)

TRAJECTORY METHODS:
TABU SEARCH
ILS
VARIABLE NEIGHBORHOOD SEARCH
GRASP (GREEDY RANDOM ADAPTIVE SEARCH)
SIMULATED ANNEALING

DIFFERENT PHILOSOPHY:
ANT COLONY OPTIMIZATION
EVOLUTIONARY COMPUTATION
INTEGRATE A LEARNING COMPONENT
IMPLICITLY OR EXPLICITLY. TRY TO
LEARN CORRELATION BETWEEN
DECISION VARIABLES TO IDENTIFY
QUALITY AREAS IN THE SEARCH
SPACE
NATURE Inspired: GENETIC Algorithms
ACO

vs.

NON NATURE Inspired: TABU SEARCH (Memory!!)
ILS

Single point SEARCH: Work on single solution (trajectory methods)
TABU SEARCH
ILS
VAR NEIGHBORHOOD SEARCH

Population Based: Evolution of a set of solutions (Points in solution space)

Dynamic Obj Funct:

vs

Static Obj Funct

One neighborhood struct:

fixed landscape

VARIOUS NEIGHBORHOODS

various landscape

Memory

vs

memory less

MODIFY OBJ FUNCTION TO ESCAPE LOCAL MINIMA (Guided local search)

OBJ FUNCTION DOES NOT CHANGE

USE OF SEARCH HISTORY
(Short term + Long term)
**Basic Local Search:**

1. **INITIAL SOLUTION**
2. **Repeat**
   3. **Select** \( N(s) \)
   4. **Improve** \( N(s) \)
5. **Until** no improvement possible
6. **Improve** first improvement
7. **Best** improvement random

**Simulated Annealing:**

Among first metaheuristics.

Statistical mechanics

(Metropolis Alg)

Kirkpatrick 1983

Allow moves that generate probability of such moves "worse" solutions decrease in time

1. **INITIAL SOLUTION**
2. **Set** initial solution
3. **T** initial temperature
4. **While** termination condition not met do
5. **Select at Random** \( N(s) \)
6. **If** \( f(s') < f(s) \) then \( s' = s' \)
7. **Else** Accept \( s' \) as new solution with probability \( p(T, s, s') \)
8. **Update** \( T \) 
9. **Endwhile**

\[ \text{prob} \in \frac{e^{-\frac{f(s') - f(s)}{T}}}{\text{Boltzmann dist}} \]

Uphill moves high initially then lower
**Tabu Search**

\[ S \leftarrow \text{Initial Solution} \]
\[ \text{Tabu List} \leftarrow \emptyset \]

\[ \text{while termination condition not met do} \]
\[ S \leftarrow \text{Choose Best of } (N(s) - \text{Tabu List}) \]
\[ \text{Update (Tabu List)} \]

end while

Glover (1977)

Tabu List: Short-term memory. (Dynamic neighborhood list)

memory controls search (more memory more uphill moves)

amount of memory may be variable

**Explorative Local Search**

Greedy Randomized Adaptive Search Proc (GRASP)

\[ S \leftarrow \emptyset \text{ (partial solution)} \]
\[ L \leftarrow \text{Candidate list length} \]

\[ \text{while solution is not complete do} \]
\[ RCL_2 \leftarrow \text{Generate Candidate list} \]
\[ x \leftarrow \text{Select an random (RCL_2)} \]
\[ S \leftarrow S \cup \frac{1}{3} x^3 \]
\[ \text{Update greedy function(s)} \]

end while
Population Based Methods

Deal with a set of solutions

Evolutionary Computing (Evolutionary process)

\[ \text{operator} \rightarrow \begin{cases} \text{recombination} \\ \text{crossover} \end{cases} \text{combine 2 or more individuals} \]

\[ \text{mutation or modification} \rightarrow \text{Individual} \]

\[ P \rightarrow \text{Initial Population} \]

\[ \text{EVALUATE}(P) \]

while termination conditions not met do

\[ P \leftarrow \text{Recombine}(P) \]

\[ P'' \leftarrow \text{Mutate}(P') \]

\[ \text{EVALUATE}(P'') \]

\[ P \leftarrow \text{Select}(P'' \cup P) \]

Endwhile

EC -- Evolutionary Programming (EP) -- Machine intelligence

\[ \text{continuous opt problem} \]

\[ \text{EVOLUTIONARY PROGRAMMING (EP)} \]

\[ \text{strategies (ES)} \]

\[ \text{continuous opt problem} \]

Genetic Algorithms (GA) -- for CO problems

GA individuals -- genotypes

solutions encoded by individuals are called phenotype.
Ant Colony Optimization:

Food source —— Pheromone —— Nest

Strange pheromone -> Shunt path

Pheromone model

Initialize Pheromone Values ($T$)

While termination condition not met do

For all ants $a \in A$ do

$S_a = Construct Solution (T, H)$

Endfor

Apply online delayed Pheromone Update ($T$, $\{S_a | a \in A\}$)

Endwhile

Random walk of ants in a completely connected graph $G(C, L)$

$C$: Solution Components

$L$: Connections

Partial solution

Feasible solution

Problem dependent

Other partial feasible solution