Tabu Search

Developed by Fred Glover in the 1970's. Dr Glover is a business professor at University of Colorado at Boulder. Developed specifically as a combinatorial optimization tool.

Biological inspiration is "memory" – the ability to use past experiences to improve current decision making.

Short term memory = recency based tabu list;

Long term memory = aspiration criteria, frequency based tabu list.

Is basically a single solution, deterministic neighborhood search technique that uses memory (a "tabu list") to prohibit certain moves, even if they are improving. This makes tabu search a global optimizer rather than a local optimizer.

Components:

- encoding
- objective function
- move operator
- neighborhood of candidate moves
- termination criteria
- tabu list(s)
- aspiration criteria (optional)

Memory Aspects

1. recency (short term)
   how recently was I here?
2. frequency (long term)
   how often have I been here?
3. quality (aspiration)
   how good is being here?
4. influence (aspiration)
   how far away am I from where I have just been?

Search attempts to balance intensification (local search) with diversification (global search).

Aspiration is added to allow tabu moves if they are “interesting” (i.e., fulfill aspiration criteria).

Canonical TS

```
set tabu list length, t
randomly generate a single initial solution
      until termination criteria is met {
            for neighborhood i = 1 to N {
                  make a move to a neighboring solution
                  evaluate
                  record in ranked order, j = 1 to N
                  ++ i }
                  for j = 1 to N {
                  if solution j is not tabu, break to 100
                        else if solution j is better than best so far, break
                        to 100
                  ++ j  }
100         add solution j to tabu list
            if tabu list length > t, remove oldest solution from it
            if solution j is better than best so far
                  best so far = solution j
      }
return best so far
```

Tabu List Specifics

- the main **objective** of the tabu list is to avoid “cycles”, thus making a global optimizer rather than a local optimizer
- **length** – fixed, dynamic; generally 7 to 20
- **content** – “from” attributes, “to” attributes; “move” attributes, the more specific, the less restrictive
- **frequency** – tallying similar solutions through tabu content. Usually used in penalty form rather than strict tabu.

Neighborhood Specifics

- complete (deterministic)
- partial (probabilistic)
- first improvement
- only improving

Aspiration Criteria Specifics

- best so far
- best in neighborhood
- dissimilar to existing solution (diversification)
- similar to existing solution (intensification)
- high influence – degree of change in structure or feasibility

More Complex Strategies

Strategic Oscillation

generally moving between feasible and infeasible border

Path Relinking and Tunneling

move step by step (by the shortest route) from one good solution to another good solution (tunneling allows stepping through infeasible solutions). Restart at one of these interim solutions.

if the good solutions are similar, this is intensification

if the good solutions are dissimilar, this is diversification

Backtracking

keep track of “elite” solutions and restart at them but move in a different direction

Restart

like backtracking, but start from purely random new solution.

Theory of TS

none

Downsides of TS

- single solution search method.
- tends to prematurely converge to local optimum – must counteract this by diversification, restart.
- can be sensitive to starting solution
- can be sensitive to tabu list length and content
- is most effective where there is a good neighborhood structure than can be exploited through proper moves
- not very appropriate for continuous optimization.

Example – TSP

- A B C D E encoding of 5 cities

- *t*=3, move = swap position of 2 cities

- tabu list possibilities:
  any move of that city
  any pair of cities swapped
  any pair of cities and their sequence

B A C D E t = {A, B}
t = {A+B}
t = {B(1), A(2)}

B A C E D t = {D, E}
t = {D+E}
t = {E(4), D(5)}

C A B E D t = {B, C} TABU
t = {B+C}
t = {C(1), B(3)}

C B A E D t = {A, B} **TABU**
t = {A+B} **TABU**
t = {A(3), B(2)}

C A D E B t = {B, D} **TABU**
t = {B+D}
t = {B(5), D(3)}

for frequency based tabu list, might use a city in a given position or the swap of two cities, or any movement of a given city

for example, if city A is put too often in position 2, or if cities C and D are swapped too often, or if city E is moved too often.

Web Sites

- http://www.cs.sandia.gov/opt/survey/ts.html
  short page with some C code for a version of TS

- http://www.winforms.phil.tu-bs.de/winforms/research/tabu/tabu.html
  another short page

- http://www.sce.carleton.ca/netmanage/tony/ts.html
  page with links (most of which don't work)

- http://spot.colorado.edu/~glover/
  Fred Glover's homepage at U CO

- http://iis.cse.eng.auburn.edu/~roland/projects/tabu.php
  look at the address!