

CEE 5290/CS 5722/ORIE 5340 Heuristic Methods for Optimization
Homework 5: Satisfiability and Tabu Search

Assigned: Fri Sep 26, 2014

Due: Fri Oct 3, 2014 @ noon

Prof. Shoemaker Office Hours: Mon, Wed, Fri 2:30-3:30

TA Office Hours: Thu (3:00-4:30) at Hollister 359

Readings: Lecture slides on Satisfiability by Prof Selman (see blackboard) and Readings on Tabu Search in the Xeroxed notes.

1. **(no code writing) Satisfiability Formulation:** In propositional calculus it is possible to represent logical expressions as clauses by applying a sequence of simple logical operators. For example, “C is true if either A or B is true” can be represented as $C = A \vee B$, where \vee represents a logical ‘OR’. Similarly, \wedge represents a logical ‘AND’; and \neg represents a logical ‘NOT’.

A problem can be stated in conjunctive normal form (CNF) as a sequence of clauses that must be satisfied for the problem to be solved. For example,

$$E = (A \vee B) \wedge (\neg A \vee C) \wedge (C \vee \neg D)$$

is in conjunctive normal form with 3 clauses and 4 variables. Note that the variables take on either a 0 (FALSE) or 1 (TRUE) assignment. We will attempt to formulate a simple assignment problem as a 2-SAT (containing 2 literals in each clause) problem.

Ann, Brad, Cindy, Dan, Eugene and Frank have to get back to their apartment from a club. They have exactly one car (can fit 5 people maximum), and not all can fit into the car. Anyone not in the car will have to get a ride from the bartender who will not leave for another hour. After an inebriated argument they decide on the following:

- (a) Only Eugene and Dan can drive, so at least one of them should be in the car
- (b) At least one of the women should be in the car
- (c) If Ann is not in the car then Brad and Frank must accompany her since the bartender has been hitting on her
- (d) If Cindy is in the car then Frank cannot be in the car. If Cindy is not in the car then Eugene must go with her

Consider the problem of assigning people to go in the car given the constraints above:

- (i) How many decision variables should this problem have in order to be formulated as a 2-SAT problem? What is the total number of possible assignments (including the assignments that violate constraints)?

- (ii) Identify appropriate clauses using only 2 literals in each to reflect their requirements. Use the first letter of the name to represent each person in the clauses. Give the CNF representation for the problem. (Hint: Six clauses should be sufficient)
 - (iii) Is this problem satisfiable? If so, come up with a satisfiable assignment. (Hint: You do not have to enumerate all possible assignments; this can be solved by logical reasoning)
2. **(involves coding) Solving 3-SAT with Tabu Search:** Design and Implement a Tabu Search to solve the Satisfiability instance which is provided on the course web page (as a MATLAB load-able text file **uf20_01.txt**; type *load uf20_01.txt* at the command prompt). This is a randomly generated 3-SAT instance with 20 variables, 91 clauses, and is taken from the SATLIB library of benchmark problems on the web. Each row in the matrix corresponds to a different clause. For example the first row, which reads [4 -18 19] means that the first clause is $(s_4 + (\sim s_{18}) + (s_{19}))$, where + indicates a logical OR, and \sim a logical NOT, and s_k is the k^{th} binary variable.

Write a short-term tabu search algorithm (Tabu.m) to solve the Satisfiability instance uf20_01.txt:

- Base the Tabu Algorithm on the pseudo-code in homework4. NOTE: this is modified from the version in the Xeroxed text.
- Define neighbors in this satisfiability problem as those solutions that differ from the current solution by only one bit.
- Use the *best solution aspiration criterion*: accept a tabu move if and only if it leads to a cost lower than any previously encountered. For this problem, a flip on a bit is tabu if the flip has occurred on the same bit in the last K moves.
- The algorithm parameter that you should experiment with is the tabu tenure length $K=|T|$.
- Let your tabu list T consist of the index of bits which have been flipped in the last K iterations, i.e. it is tabu to flip a bit that has been flipped within the past K iterations.
- This is a *deterministic* Tabu implementation (i.e. set $V=\text{number of neighbors in the neighborhood}$ (Note: This means that all members of the neighborhood of the current solution)).
- You must also write a cost function for this satisfiability problem (costSAT.m).

Questions

- (i) Submit code for *only* Tabu.m, costSAT.m (cost function evaluation) and the neighborhood implementations *if* your neighborhoods are implemented outside of Tabu.m.
- (ii) How many different solutions or variable assignments (i.e. satisfiable and unsatisfiable solutions) are there in this problem?
- (iii) Design an experiment to determine the best values of K. Briefly outline the design of the experiment and any assumptions. There are many reasonable answers.
- (iv) *For* the best value of K found above, solve the instance 100 times from 100 different initial solutions. You should determine the number of iterations you wish to run and justify why you picked that number (e.g. x iterations is sufficient to find a satisfiable solution). In the past, students have used number of iterations of around 100. Report the following:
 - The number of trials for which you find a satisfying solution and if you found more than one satisfiable solution (not how many total, just note if there are more than one).
 - The number of cost function evaluations required on average to find *the satisfying solutions* (do not count trials that do not find satisfying solutions in this average)
 - For any trials that do not find satisfying solutions, report the number of cost function evaluations per trial and average value of your best objective function(aspiration level) over 100 trials .