CEE 5290/ CS 5722/ORIE 5340 Heuristic Methods for Optimization

General Instructions for Project

Submission of Names on Team and Topic on Existing Projects due October 11
PhD students should submit draft Project Description by October 11
Written Report due Nov. 11, 2013(slightly soft deadline)

This course includes a group project. The topics for the project have been designed to introduce the students to some of the real-world problems in various disciplines where heuristic methods are applied for optimization. This will also give you experience with algorithm performance comparisons and with being creative about how you might modify your heuristic algorithms and neighborhood functions to have improved performance on your application problem.

Note: All students are expected to attend the talks about the projects presented by outside speakers and by Prof. Shoemaker

I. General Information

1. All students enrolled for 4 credits are required to work on the group project to meet the course requirements.
2. All students are expected to attend all the talks about the projects topics even if the student is doing the 3 credit-no project option. The purpose of the talks is to help students understand how heuristics are used in different situations so everyone should hear these talks. There are some homework problems related to the talks about project topics.
3. You will want to use the codes you have written from homework for the project so you probably want to look NOW at Section V.1 about which coding steps you might want to take in your homework codes.
4. Guest speakers from various departments and the course instructors will introduce the topics for the various projects that have been established. The topics of "Existing Projects" and the corresponding speakers are given below:

- **Satisfiability Testing for Artificial Intelligence**: Prof. Bart Selman, Computer Science (Oct. 9 presentation)
-
- **Resource Allocation in Cellular (Phone) Networks**: Prof. Steven Wicker, Electrical and Computer Engineering (Sept. 30 presentation)

- **Optimal Control of Partial Differential Equation Systems with Environmental Applications**: Prof. Christine Shoemaker, Civil and Environmental Engineering. (Oct. 4 presentation)

The following topics are available but do not have guest lectures.

- **Protein Folding**: TBD

-
- **Job Shop Scheduling or Alternative ORIE Problem TBD**
-
- **Financial Engineering project (American option)**

5. Students should form groups of 2 to 4 persons to work on the projects. Ph.D. students can work alone on a project of their choosing that is different than the six listed above provided it is related to their thesis. In this case, students should notify the TA Min <mp723> by email of their team by Oct 5th . Students can make use of Piazza to find partners.
6. Every member of the group is expected to contribute equally to the project, and work closely with the other members in the group. You should know at all times what the other members are working on.
7. The team member names and the topic of the project must be finalized by date given at beginning of these instructions and the TA notified about the composition of the group. Ph.D students who want to create their own project will need to give an abstract draft of their projects by the Oct 14th .

- .

Project Teams

Unless you are a Ph.D. student, you should have **two to four people** on your project team. Ph.D. students can work alone or in a group. Three is usually a good number. You will be asked in the report to indicate how the work was divided among you. For example you might say "John focused on the SA analysis and Mary focused on ….., etc.". However, all students in the project are expected to understand all the work done by all team members, i.e. the student working on the GA application might be asked to explain rational for the neighborhood function used for SA.

Project Requirements

1. For all integer-valued optimization problems, you are expected to run and compare the following algorithms:
   (a) Simulated Annealing (SA)
   (b) Binary Genetic Algorithm (GA)
   (c) Tabu Search (TS)
   (d) Random Sampling (RS) algorithm (i.e. a Monte Carlo Search) or Stochastic Greedy Search (GS).

   For all real-valued optimization problems, you are expected to run and compare the following algorithms:
   (a) Simulated Annealing (SA)
   (b) Real-value Genetic Algorithm (GA)
   (c) Dynamically Dimensioned Search (DDS) or a Tabu Search (TS) modified for real-valued problems
   (d) Random Sampling (RS) algorithm (i.e. a Monte Carlo Search) or Stochastic Greedy Search (GS).

2. You are also expected to propose and try out at least two type of variant of one or more of the above algorithms that demonstrates some creativity and thought by the group and *hopefully* improves algorithm performance. Examples of this include implementing the following: an advanced neighborhood function, changing parameters dynamically (e.g. changing them as you make more iterations), implementing a GA operator that was not requested in the homework or for the other algorithms, or adding a new modification, like an integer version of DDS. These are just examples and you can create your own variant.
3. It is *very preferable* to use at least 30 optimization trials per algorithm (more if you want to). Therefore, you should expect to perform a minimum of **roughly 5 days of dedicated (serial) PC run time.** Obviously, run-times will vary with project and algorithm coding, and this is something you should try to estimate well in advance of the due date.

4. The point here is that you will almost certainly not be able to complete all optimization runs in a short time, and you need additional time to summarize your results and write the report. And since you have other use for your laptops, you may want to plan ahead so you can spread these runs over times when you don't need your computer (e.g. when sleeping). This is why you are given over a month for completing this project. If you do not or are unable to perform 30 optimization trials, the minimum number of trials to perform per algorithm is 10 and you must justify in your report why less than 30 trials are used. (For example the run time for the COST function are longer for some of the projects like the PDE problem.) You also need to discuss why you need to shorten the number of runs with your TA at least 2 weeks before the report is due.

5. For continuous optimization problems the students can get in touch with the TA through Piazza for details of implementing discrete algorithms (such as Tabu Search) if you wish to explore that.
6. You will need to determine suitable neighborhood definitions and crossover/mutation operators for your particular problem. This may involve some research, experimentation and/or justification based on problem structure, including constraints.
7. You will also need to do some experimentation to pick good algorithm parameters. For example, in simulated annealing you will need to determine suitable temperature and alpha values; for GA, the population size, mutation rate, crossover rate etc., are relevant parameters.
8. More specific instructions are provided for each project.

IV. Report Submission

1. Your grades for the project will be based on the excellence of the your work (including correctness of calculations, clarity of written report, creativity) as reported in a jointly submitted report (submit both a hardcopy and a softcopy of their projects by email to mp723@cornell.edu

The file should be named as such: Heuristic_ProjectDescription(e.g. PDE, SAT, JobShop, Protein, Cellular,Other)_LastNames. For example: Heuristic_ProjectPDE_Wan_White.

2. Your final report should be in the form of a professional quality conference paper and is not to exceed 8 pages. **Use** the format described in the MS Word template file (word_template.doc) based on the *Congress on Evolutionary Computation* Conference guidelines. You can simply start writing over the sample with your project text.
3. In the report be sure to include:
    a. A very brief description of the problem and cost function.
    b. A brief summary of the algorithms you apply to the problem - one paragraph here is sufficient for algorithm sIII.1 (a) through (e).
    c. A description of your idea/implementation for improving one or more of the algorithms and why you thought it might be better than the basic algorithm implementation.
    d. A concise description of your experiments to determine suitable neighborhood definitions and algorithm parameters (and associated results)
    e. Plots/tables comparing the performance of these algorithms with the best parameters found for each
    f. Summary of statistical test results
    g. Best solution obtained (decision variable values) along with its cost.
    h. Plots that compare algorithm results (best cost versus the number of function evaluations). If necessary, use more than one plot to distinguish algorithm results (e.g. 10 lines on one plot may be difficult to distinguish).
    i. Comments and conclusions regarding results.
4. You are strongly discouraged from handing in any appendices to the report – they will not be graded. **Do not submit code.** (Unless you are one of the people who are doing projects other than the 6 examples given above).
5. The project due date is given at the top of this document. Explain the contributions of each member (i.e. which codes, text, graphs, etc. were done by each member) at the end of the report (after references).

V. Other Project Notes and Recommendations

1. You can get started 'early' on your projects by considering the following when working on your next few homework assignments:
    a. Determine the coding required in your main program so that you can save the algorithm output of interest to some form of output file
    b. Be creative and think about, record and possibly test out any improvements to the basic algorithms you are asked to code in the homework.
    c. Make your algorithm and main program coding general enough so that you can either quickly modify it for a new function and new neighborhood or so that it is general enough that it can handle any user input functions.

2. Although not required, feel free to try any other optimization algorithm of interest to you on your problem.
3. Plan your optimization runs carefully. Keep the following in mind:
    a. To ensure fair comparisons, each algorithm should be run for the same number of function evaluations. Report total CPUtime, and the total number of function evaluations for convergence (maybe to maximum iterations). For example, the total number of function evaluations will be the same (1500) in all the following cases:
        i. SA with 1500 iterations (evaluate one member of N(S) per iteration).
        ii. GA with population size of 50, 30 generations, (50*30 = 1500)
        iii. TS with 5 neighbors and 300 iterations, (5*300 = 1500)
        iv. DDS with 1500 iterations (evaluate one member of N(S) per iteration).
        v. Random Sampling or Greedy Search with 1500 iterations.
    b. One of your **first tasks** should be to assess available PC resources, compute estimates of runtimes for various iteration requirements, and calculate when computational runs must be started by so that computer resources do not prevent you from turning the work in on time. Recall that you want to allow enough tie to re-do runs if necessary.