IE 607 Heuristic Optimization

Introduction to Optimization
Part I

- **Objective Function**

Max (Min) some function of *decision variables*

Subject to (s.t.)

equality (=) *constraints*

inequality ($<, >, \leq, \geq$) *constraints*

- **Search Space**

  Range or values of decisions variables that will be searched during optimization. Often a calculable size in combinatorial optimization

Types of Solutions

- A *solution* to an optimization problem specifies the values of the decision variables, and therefore also the value of the objective function.
- A *feasible solution* satisfies all constraints.

  v.s. *infeasible solution*
- An *optimal solution* is feasible and provides the best objective function value.

  → *global* optimal solution v.s. *local* optimal solution
- A *near-optimal solution* is feasible and provides a superior objective function value, but not necessarily the best.

Continuous vs Combinatorial

- Optimization problems can be *continuous* (an infinite number of feasible solutions) or *combinatorial* (a finite number of feasible solutions).
- Continuous problem generally maximize or minimize a function of continuous variables such as min $4x + 5y$ where x and y are *real numbers*.
- Combinatorial problems generally maximize or minimize a function of discrete variables such as min $4x + 5y$ where x and y are *countable items* (e.g. integer only).

Definition of Combinatorial Optimization

- Combinatorial optimization is the mathematical study of finding an optimal arrangement, grouping, ordering, or selection of discrete objects usually finite in numbers.

  \- *Lawler*, 1976

Definition of Combinatorial Optimization (cont.)

- Combinatorial optimization is concerned with the efficient allocation of limited resources to meet desired objectives when the values of some or all of the variables are restricted to be integral.
- In practice, combinatorial problems are often more difficult because there is *no derivative information* and the *surfaces are not smooth*.

Types of Constraints

- Constraints can be *hard* (must be satisfied) or *soft* (is desirable to satisfy).

  Example: In your course schedule a hard constraint is that no classes overlap. A soft constraint is that no classes can be arranged before 10 AM.

Types of Constraints (cont.)

- Constraints can be *explicit* (stated in the problem) or *implicit* (obvious to the problem).

  Example: When considering a hospital scheduling problem, it is implicitly a integer problem. → People can’t be divided.

Aspects of an Optimization Problem

- Continuous or Combinatorial
- **Size** – number of decision variables, range/count of possible values of decision variables, search space size
- **Degree of constraints** – number of constraints, difficulty of satisfying constraints, proportion of feasible solutions to search space
- **Number of objectives:** *Single* or *Multiple*

Aspects of an Optimization Problem (cont.)

- *Deterministic* (all variables are deterministic) or *Stochastic* (the objective function and/or some decision variables and/or some constraints are random variables)
- **Decomposition** – decompose a problem into series problems, and then solve them independently
- **Relaxation** – solve problem beyond relaxation, and then can solve it back easier

Characteristics of an Optimization Problem

Simple	Hard
Few decision variables	Many decision variables
Differentiable	Discontinuous, combinatorial
Single modal	Multi modal
Objective easy to calculate	Objective difficult to calculate
No or light constraints	Severely constraints
Feasibility easy to determine	Feasibility difficult to determine
Single objective	Multiple objective
Deterministic	Stochastic

Characteristics of an Optimization Problem (cont.)

- For Simple problems, enumeration or exact methods such as *differentiation* or *mathematical programming* or *branch and bound* will work best.

- For Hard problems, differentiation is not possible and enumeration and other exact methods such as math programming are not computationally practical. For these, *heuristics* are used.

Basic Terms

- **Search** is the term used for constructing/improving solutions to obtain the optimum or near-optimum.

Solution Encoding (representing the solution)

Neighborhood Nearby solutions (in the encoding or solution space)

Move Transforming current solution to another (usually neighboring) solution

Evaluation The solutions' feasibility and objective function value

Classifications of Search Techniques

- **Constructive** search techniques work by constructing a solution step by step, evaluating that solution for (a) feasibility and (b) objective function.

- **Improvement** search techniques start with an initial solution, moving to a neighboring solution, evaluating that solution for (a) feasibility and (b) objective function.

Classifications of Search Techniques (cont.)

- Search techniques may be *deterministic* (always arrive at the same final solution through the same sequence of solutions, although they may depend on the initial solution). Examples are LP (simplex method), tabu search, simple heuristics like FIFO, LIFO, and greedy heuristics.

- Search techniques may be *stochastic* where the solutions considered and their order are different depending on random variables. Examples are simulated annealing, ant colony optimization and genetic algorithms.

Classifications of Search Techniques (cont.)

- Search techniques may be *local*, that is, they find the nearest optimum which may not be the real optimum. Example: greedy heuristic (local optimizers).

- Search techniques may be *global*, that is, they find the true optimum even if it involves moving to worst solutions during search (non-greedy).

Heuristics

- Heuristics are rules to search to find optimal or near-optimal solutions. Examples are FIFO, LIFO, earliest due date first, largest processing time first, shortest distance first, etc.

- Heuristics can be *constructive* (build a solution piece by piece) or *improvement* (take a solution and alter it to find a better solution).

- Many constructive heuristics are *greedy* or *myopic*, that is, they take the best thing next without regard for the rest of the solution.

  Example: A constructive heuristic for TSP is to take the nearest city next. An improvement heuristic for TSP is to take a tour and swap the order of two cities.

Meta-Heuristics

An iterative generation process which guides a subordinate heuristic by combining intelligently different concepts derived from classical heuristics, artificial intelligence, biological evolution, natural and physical sciences for exploring and exploiting the search spaces using learning strategies to structure information in order to find efficiently near-optimal solutions.

- *Osman and Kelly*, 1996

Meta-Heuristics (cont.)

Metaheuristics are high level concepts for exploring search spaces by using different strategies. These strategies should be chosen in such a way that a dynamic balance is given between the exploitation of the accumulated search experience (intensification) and the exploration of the search space (diversification)…

- *Blum and Roli*, 2001

Classifications of Meta-Heuristics

- Nature-inspired vs Non-nature-inspired
- Population-based vs Single point search
- Dynamic vs Static objective function
- One vs Various neighborhood structures
- Memory usage vs Memory-less methods

- This course will focus on **meta-heuristics** inspired by *nature*.
- Meta-heuristics are not tied to any special problem type and are general methods that can be altered to fit the specific problem.
- The inspiration from nature is:

Simulated Annealing (SA) – molecule/crystal arrangement during cool down

Evolutionary Computation (EC) – biological evolution

Tabu Search (TS) – long and short term memory

Ant Colony and **Swarms** - individual and group behavior using communication between agents

Advantages of Meta-Heuristics

- Very flexible
- Often global optimizers
- Often robust to problem size, problem instance and random variables
- May be only practical alternative

Disadvantages of Meta-Heuristics

- Often need problem specific information / techniques
- Optimality (convergence) may not be guaranteed
- Lack of theoretic basis
- Different searches may yield different solutions to the same problem (stochastic)
- Stopping criteria
- Multiple search parameters