IS703: Decision Support and Optimization

Week 11: Local Search

Lau Hoong Chuin
School of Information Systems
Last Week…

Single-Point Meta-Heuristics
(TS, SA)

Today…

Population-Based Meta-Heuristics
(GA, ACO)

Hybrid Meta-Heuristics
Genetic Algorithms

John Holland (1975)

Components:

• Encoding / solution (gene, chromosome)
• Initialization procedure (creation)
• Evaluation function (fitness in environment)
• Selection of parents (reproduction)
• Genetic operators (mutation, crossover)
Standard Genetic Algorithm

initialize population;
evaluate population;
while TerminationCriteriaNotSatisfied
{
    select parents for reproduction;
    perform genetic operations to generate offspring (new population)
evaluate population;
}
TSP Example

Encoding is an ordered list of city numbers

1) London   3) Dunedin   5) Beijing   7) Tokyo
2) Venice   4) Singapore   6) Phoenix   8) Victoria

CityList1   (3  5  7  2  1  6  4  8)
CityList2   (2  5  7  6  8  1  3  4)
Crossover

Parent1  (3  5  7  2  1  6  4  8)
Parent2  (2  5  7  6  8  1  3  4)
Child    (2  5  7  2  1  6  3  4)

Note: may generate infeasible offspring!
Crossover

Cycle crossover:

Parent1  (2 3 5 6 4 1 7 8)
Parent2  (1 4 2 3 6 5 8 7)

Child1   (1 3 2 6 4 5 8 7)
Child2   (2 4 5 3 6 1 7 8)

- generates feasible offspring by identifying subsets of vertices that occupy the same subset of positions in both parents.
Mutation

Mutation involves reordering of the list:

Before: \((5 \ 8 \ 7 \ 2 \ 1 \ 6 \ 3 \ 4)\)

After: \((5 \ 8 \ 6 \ 2 \ 1 \ 7 \ 3 \ 4)\)
Genetic Algorithm: Why does it work?

• Children carry parts of their parents’ data
• Only “good” parents can reproduce  
  – Survival of the Fittest (Does it always work?)
• Large population allows many “current points” in search  
  – Consider several search regions at once
Genetic Algorithm: Issues & Pitfalls

• Representation
  – Preserving feasibility
  – Children (after crossover) should be similar to parent

• Coverage
  – Population should be large and random enough to “cover” the range of possibilities
  – Mutation helps with this issue, multiple populations, etc
**Ant Colony Optimization**

Dorigo et al (1991)

- **Uses probabilistic greedy to construct a solution**
  - Solution constructed using probability wrt greedy factors
  - Allow “bad” edges/paths to be chosen

- **Iterative population-based constructing heuristic**
  - Subsequent construction learns from previous experience
  - Using pheromone deposited by previous ants as common communication media
**ANT COLONY OPTIMIZATION**

- **Common Terms**
  - Partial Solutions
  - Constructing Move
  - Local Heuristics
  - Pheromone Trails

**Constructing a Hamiltonian Circuit**
Ant Colony Optimization

Communication Media: Pheromone Trails

- Pheromone Trails allow previous ants to “pass” information to subsequent ants
  - Slowly converges into the best path
  - Over-deposit may leads to over-convergent
Ant Colony Optimization pseudo codes

while terminating conditions not reached

while there is still ants in colony and

for each ant do

   Pick up a path based on *Local Heuristic* and *Pheromone Trail*
   Path is added into the partial solution

   until the tour is completed

   Apply Local Decay

end while

Update Global Deposit

Apply Evaporation

end while
ACO for TSP

Pheromone Trail

- **Local Decay**: Enhance Exploration
  - Ants in same iteration remove a percentage of pheromone on the trails it passes
  - “Persuade” other ants in iteration to explore other paths

- **Global Deposit**: Enhance Exploitation
  - Iteration-best ant trails is deposited with pheromone
  - Encourage other ants in subsequent iterations to follow “good” paths
  - Deposition must consider the decrement by **Local Decay**

- **Evaporation**: Prevent rapid convergence
  - Decrease the pheromone deposit on ALL trails by certain percentage
  - Prevent “good” paths from selection with too high probability
ACO Advanced Strategies

- **Elitist Ants / ASRank**
  - *Ant System*: ALL solutions found by ants are updated
  - *Elitist Ants*: Update only the iteration-best ants
  - *ASRank*: Pheromone update is proportional to their “rank”

- **MAX-MIN Ants**
  - Define a upper and lower bound for pheromone density
  - Prevent over-deposition or over-removal of pheromone

- **Exploitation versus Exploration**
  - Exploitation: Force ants to follow closely to “good” paths
  - Exploration: Allow ants to wander onto poorer paths
  - Rotate between the two strategies using a $q_0$ factor
Hybrid Meta-Heuristics
**HYPER-HEURISTICS**

Burke et al (1996)

- **Heuristics that choose heuristics**
  - High level heuristics: Choice function, GA, etc
  - Low level heuristics: Different move strategies, constructive heuristics, etc

- **Successfully applied in:**
  - Bin Packing
  - Time-tabling
  - Rostering
**HYPER-HEURISTICS (Concept)**

Source: Peter Crowling, U of Bradford
HYPER-HHEURISTICS (Time Tabling)

- High level heuristics search for lists of low level graph heuristics to construct solutions
- Low level graph heuristics: order events by how difficult to schedule them
  - Saturation Degree: least available slots
  - Colour Degree: most conflicted with those scheduled
  - Largest Degree: most conflicted with the others
  - Largest Weighted Degree: students involved
  - Largest Enrolment: students enrolled
  - Random Ordering
HYPER-HEURISTICS (Assessment)

Strengths:
- Low level heuristics easy and quick to implement
- Solution quality is better than pure heuristics

Challenges (Burke, et al 2005):

“The internal state of the hyper-heuristic is a matter for the designer, as is the process to decide on which heuristic to call next…”

“Effective choice function is critical to the success of this method, and this is where most research has been directed.”
Conclusion

- Local Search / Meta-heuristics algorithms generally do not guarantee global optimality
- No Free Lunch Theorem
  - Wolpert and Macready (1996)
  - No one-size-fits-all strategy
    - Different problems require different strategies
    - Different problem instances of the same problem require different strategies