John Beighle

GBL rev k 5/20

Department of Electrical and Computer Engineering

CSCE686 Advanced Algorithm Design

Spring Quarter 2020

Complex Optimization Algorithm Design Project

Introduction to Project Problem Selection

The intent of this individual project is to incorporate the major educational aspects of search algorithm development for combinatoric complex optimization problem solutions (NP-Complete, PSpace, …) . These discrete optimization projects can be an aspect of the student's research or selected from the list provided (*note that each student should have a unique project*).

General Educational Objectives:

1. Develop an *ability to design and evaluate algorithms* for solving complex NPC or more complex scientific and engineering optimization problems using explicitly the design techniques presented in class. This includes explicit problem domain specifications and selection and integration of appropriate search algorithm templates from the deterministic and stochastic spectrum. This could result in software engineered code that is effective and efficient and well documented. And, thus, should make testing less complex and maintenance less expensive.

2. Develop an ability to define appropriate experimental design in order to characterize computational performance (*effectiveness and efficiency*) for solving such optimization problems.

Project Development: ***(Define your project problem!)***

a) **Problem Selection:** (English → symbols → math/logic). "with constraints" Select and discuss a specific real-world discrete optimization problem. It should **incorporate at least TWO independent NP-Complete problem models (multi-objective).** Aggregate objectives into one evaluation function (additive, …). Another project selections include **a dynamic NPC problem with a nonlinear model or with extensive constraints or a PSPACE-Complete Problem**. What does the search landscape look like? Di and Do

John Beighle

Project Domain Description:

This CSCE686 final project proposal supports broader picture research/thesis domain-of-study objectives and will assist providing potential prospective inputs into that extended area of work. Thus, it is why this storyline was selected in hopes it helps expand, at least, creative possibilities and discussion points for the overarching goals of those efforts.

Thesis objectives, at this point, involve building a space-based game of sorts that involves contested space and pursuer evader satellites in low earth orbit. To fall in line with both the requirements of CSCE686 final project and thesis work the CSCE686 project will be proposed as the following scenario:

There is a Red (R) team recon Satellite that likes to fly in low orbit to take high resolution photographs of varying and multiple earth-based targets during each day. There are ten circular orbits, each within a 50km range of each other, that it can fly to according to the daily photo objectives and other associated risk factors. There are also two Blue (B) satellites that would like to capture the Red satellite, dock it, and reprogram its firmware to send back adjusted recon data. They are constantly trying to do this. R knows it's being chased but it must both continue its mission requirements and switch orbits every hour. So, every day it must build a route schedule that drops it into the bottom 3 recon orbits for the daily scheduled durations, and then fly into higher orbits that use less fuel after each photo shoot. These shoots take 1 hour each. R also, on account of risk, must also change orbit every hour. These changes can be schedule for the entire day as if B's did not move. However, they will move. Travel cost and constraints will be as follows:

B Position			+4	X	+4	+4	X	+4		
Cost	+3	+2	+1	0	+1	+2	+3	+4	+5	+6
Orbits	1	2	3	4	5	6	7	8	9	10

24 Hour Photo Example Daily Mission Requirement

@ 0100 and @0300	@0500 and @0700	@0900 and @1200

R's 24-hour route plan can be created with a simple TSP algorithm. However, real life is not that easy, and the B's are making things difficult. So, a TASKORD project has been funded to create a dynamic TSP algorithm that can recalculate R's best route at any time needed during the day, anytime the risk factors change, i.e. the B's have changed orbits and increased the capture risk profile for R. Space HQ wants this designed and documented before 3 July, 2020.

Algorithm Domain Details

The basic algorithm will be TSP employed in a dynamic way with a random timer to reposition Bs. However, the basic algorithm will suffice to for route planning.

"TRAVELLING SALESMAN PROBLEM A salesman has a list of cities, each of which he must visit exactly once. There are direct roads between each pair of cities on the list. Find the route the salesman should follow for the shortest possible round trip that both starts and finishes at any one of the cities. Definition The general problem can be stated as the following: If we are given a set of places {n1, n2, … n N} and for each pair {ni, nj} of distinct cities the distance between them is d(ni, nj). If a line is drawn trough all the places visiting each place only once, and end up where it started. The goal is to find an ordering of the place that minimizes the total length of that line. This creates a Hamiltonian cycle in the graph of places Stated more formal: Using the same notation, the mathematical expression is to minimize:

$$\sum_{i=1}^{N-1} d\big(V_{\alpha(i)}, V_{\alpha(i+1)}\big) + d\big(V_{\alpha(N)}, v_{\alpha(1)}\big)$$

[2]

Basic Algorithm BFS(v) {

```
// A breadth first search of G is carried out beginning
// at vertex v. For any node i, visited [ i] = 1 if i has
// already been visited. The graph G and array visited[]
// are global; visited [] is initialized to zero.
        u := v; // q is a queue of unexplored vertices.
        visited[v] := 1;
        repeat {
                for all vertices w adjacent from u do {
                        if (visited [w] = 0) then
                                Add w to q; // w is unexplored.
                                Visited[w] := 1;
                }
                if q is empty then return; // No unexplored vertex.
                Delete u from q; // Get first unexplored vertex.
        } until (false);
}
```

REFERENCES:

[1] Generating Human-readable Algorithms for the Travelling Salesman Problem, P. Ryser-Welch, J. Miller, S. Asta 2015

[2] International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 1, January 2016

John Beighle

Thoughts on possible CSCE686 Discrete Optimization Projects: (my interests)

1. Self-organized Autonomous Multi- Agents in Computer Network

- ***Define/optimize agent behavior given attacks using POMDP (Partially Observable Markov Decision Process Models)***
- ***Optimal multi-agent network attack classification using reputation***

2. UAV (aircraft, robots, satellites) "optimizing self-organizing rules/process" in dynamic environment – behavior optimization

- ***Recon/loitering scheduling, max coverage, min risk***
- ***Routing and task/weapon assignment, min response time***
- ***Combat, engagement, attack (rule-based & bio-inspired)***

3. The Vehicle Routing Problem (3D) with constraints

- ***Optimal Movement of Swarms of Autonomous Unmanned Aerial Vehicles in dynamic environment***

4. Optimal Intrusion detection (ID) and network anomaly detection (AD)

- ***ID and AD Pattern Recognition (feature selection using Heuristic search) optimization with multi-agents***

5. Real Time Strategy Games using Computer Generated Forces (AI Opponent)

- ***Tactical Planning for Optimal Real-time Games***

6. **Game Theory Modeling for Cyber Warfare**

- ***Optimize probabilistic strategy selection over Nash equilibrium.***

7. Machine learning for optimal learning of information collection and processing

8. **Cyber Security Agents**

- ***Distributed Constraint Optimization (risk, surveillance, cost)***
- ***Drone Placement and Surveillance Coverage***

Submit your initial PD model proposal by June 12 (English, symbols, and math/logic formulation) – discuss with professor as appropriate. Algorithm design (deterministic, stochastic) reporting dates are to be indicated later.