CS 4623/6623 Spring 2022
Wainwright

Possible Problems to Select for your Project

The problems to chose from (or will be assigned to you) are listed below. If you would like to work on another problem not on this list, let me know and we can discuss it. I will still need to approve any problem not on this list.

##

0-1 KNAPSACK Described in Computers and Intractability - A Guide to the Theory of NP-Completeness by Michael R. Carey and David S. Johnson, W.H. Freeman Co., 1979 (C/J p. 247) **(See How to Handle Infeasibles Handout)**

A set of *n* items is available to be packed into a knapsack with capacity *C* units. Each item *i* has value v_i and takes up c_i units of capacity. Identify a subset of items that should be packed into the knapsack to maximize the total value without exceeding the knapsack's capacity; that is:

$$\text{maximize } \sum v_i \text{ while maintaining } \sum c_i <= C$$

Chromosome Representation	Binary string (Some may be infeasible)
Fitness Function	Maximin profit within capacity limits
Min or Max Problem	Maximization

Could use a permutation and fill knapsack with the left part of the chromosome until C is reached. Now all chromosomes are feasible --- but standard crossover techniques may not be useful for this arrangement.
What if infeasible chromosomes are to be considered in the fitness function?
Let TOT be the sum of the values for all N items.
Let CHROM be the sum of the values represented in the chromosome.
Suggest: constant * (TOT – CHROM). Constant is used to keep the value in line with feasible chromosomes

Example: Suppose TOT = 500, and infeasibles tend to have values over 300, so if infeasible the fitness might be say (TOT-CHROM), but if infeasibles tend to have values over 100 the fitness might be 1/5 * (TOT-CHROM)

Consider penalizing infeasible chromosomes proportional to how infeasible they are.
For example, Let Over = 0 for feasible chromosomes and
Let Over = $\sum c_i$ - C for infeasible chromosomes (note this is greater than 0)

Fitness function = $\sum v_i$ - Over * constant, where constant keeps the fitness value competitive for the particular problem.

Discuss possible datasets such as an optimal one below:

Weight	**10**	**10**	**15**	**15**	**16**	**25**	**20**	**15**	**18**	**22**	**11**	**16**	**16**	**11**	**15**
Profit	**40**	**50**	**40**	**45**	**40**	**30**	**40**	**20**	**30**	**40**	**50**	**40**	**45**	**40**	**42**

Capacity = 50
Solution: first 4 items at profit of 175
Randomize (the positions) of the above 15 items and 5, 10, 15 copies

Another possible contrived Knapsack dataset
Solution: Value Random 6..10 Others: Value Random 1..5
Weight Random 1..5 Weight Random 6..10

20 packages: Make 10 solution and 10 other. Make C = 30 expected weight about 80
50 packages: Make 15 solution and 35 other. Make C = 45 expected weight about 120
50 packages: Make 35 solution and 15 other. Make C = 105 expected weight about 280
etc. for larger problems

Harder problems:
Solution: Value Random 1..20 Others: Value Random 1..25
Weight Random 1..10 Weight Random 2..15

##

MAX-CLIQUE (C/J p. 164)

A **clique** of an undirected graph *G* is a complete subgraph of *G*; that is, each vertex in a clique shares an edge with each other vertex. Given an undirected graph *G*, find the size of the largest clique within it.

Chromosome Representation	Binary string for the nodes; all infeasible – any feasible = solution
Fitness Function	Maximin clique size
Min or Max Problem	Maximization (Fitness could be a Max or Min function)

Note: An adjacency matrix or incident matrix should be used to represent the graph.

If one chromosome has 5 one bits then you test the represented nodes for a K_5, and if a chromosome has 12 one bits you test the nodes for a K_{12}. This is if you are trying to solve the entire problem of the max clique all at once.

Fitness function could be (a) number of edges represented by the nodes towards the clique (maximization problem) or (b) number of edges represented by the nodes short of the clique (minimization problem). We always know how many edges a clique must have. A clique of size N, (which is K_N) has N (N-1)/2 edges. Either way chromosomes may have different fitness functions which may make it difficult to drive the population to a common solution.

Instead, consider running GA for a specific clique size, then run again for the next size until no solution is found, rather that trying to determine max clique all at once.

Consider the case of 30 nodes and we are looking specifically for a clique of size 5. The size of the search space is selecting 5 unordered nodes from 30 = C(30,5) = 142,505. This should be very simple for a GA to find.

Note in this case every chromosome needs to have exactly 5 one bits. If it does not, then I recommend a fixup, either random or by some heuristic.

To build the test problem select 5 nodes and make sure the adjacency matrix connects all 5, then connect every other pair of nodes with probability P. I recommend P be between 20 and 60%.

Consider larger datasets: 100 nodes and we are looking specifically for a clique if size 17. The size of the search space is selecting 17 unordered nodes from 100 = C(100,17) = $6*10^{18}$. Consider even larger problems. They can easily be contrived so you know the solution.

##

SUM OF SUBSETS PROBLEM (C/J p. 233)

Consider a set of n positive integers. Determine whether or not there is a subset of integers whose sum is exactly equal to K, and if it exists indicate the numbers included in the subset.

The following data sets are too small, but may serve as a model to generate larger ones.

1. K=767, n=20, items={61, 20,15,93,1,13,32,37,91,73,25,56,7,34,22,41,44,6,14,82}
2. K=252, n=20, items={61, 20,15,93,1,13,32,37,91,73,25,56,7,34,22,41,44,6,14,82}
3. K=100, n=5, items={33,66,5,7,2}

Variation: A secondary constraint

Among all solutions (sum=K), pick the subset with the fewest (or most) numbers.

Chromosome Representation	Binary string to select or not each integer (Some may be infeasible)
Fitness Function	Several discussed below
Min or Max Problem	Minimization

Let SUM be the sum of the values in the subset depicted in a chromosome

binary string
If we require SUM = K then
all chromosomes are infeasible except for solutions

What do we do if sum is close to K, but not over? What if it is slightly over?
What to use for fitness? How about |Sum – K|

What about a secondary consideration of the fewest numbers in the subset?
Consider: $\alpha * |SUM - K| +$ number of one bits, where $\alpha > 1$ and adjusted for the problem size.

If you select this problem to solve use the SUM <= K variation below

Another variation is to find the subset with largest sum <= K

Now chromosomes are feasible as long as the sum<= K

Fitness: instead of |Sum – K|, how about $|Sum - K| + K/10 * \beta$, where $\beta = 0$ if feasible
$\beta = 1$ if infeasible

Now if we use the minimum number of values as a secondary condition, consider
$|Sum - K| + K/10 * \beta +$ number of one bits, where $\beta = 0$ if feasible
$\beta = 1$ if infeasible

Discuss the construction of datasets to test your GA such as: 95 unique values in the range from 20 to 2000, and then intermix 1, 4, 7, 3, 2 among these numbers. K = 17

##########

WEIGHTED MAX CUT (C/J p. 210) easy to make this a weighted min cut problem as well.

$G = (V,E)$ is a weighted undirected graph: a function associates positive integer weights with the edges in E. Partition the vertices in V into disjoint subsets V_1 and V_2 so as to maximize the total weight of the edges from E that have one endpoint in V_1 and the other in V_2. Note when all edge weights are one, the Weighted Max Cut Problem becomes the **Max Cut Problem**, which is:
Given an undirected graph $G = (V,E)$, partition the vertices of V into disjoint subsets V_1 and V_2 so as to maximize the number of edges from E that have one endpoint in V_1 and the other in V_2.

Variation: $|V_1| = |V_2|$.

Chromosome Representation	Binary string to select separating the vertices into V_1 and V_2
Fitness Function	Maximum Weight
Min or Max Problem	Maximization

If you select this problem to solve, use the $|V_1| = |V_2|$ variation

In this case the number of nodes needs to be even and every chromosome needs to have the same number of ones as zeroes. If not, I suggest a fixup either randomly or by some heuristic.
Possible contrived dataset: consider 30 nodes. Select ahead of time which 15 nodes are to be in V_1 and V_2. Then randomly connect edges in V_1 and V_2 with probability 60% and weights in the range 5..10. Then connect all of the other nodes with probability 30% and weights randomly in the range say 3..8.

##########

BIN PACKING (C/J p. 226) **(See the Bin Packing Handout)**

A Finite set of packages, P, is packed into a set of B bins. Each bin has the same capacity, and each package has positive size ranging from 1 unit to the size of each bin. Determine the packing such that all of the packages are packed into the minimum number of bins. You should use one of the following packing strategies: first fit, next fit, or best fit. (These strategies will be discussed in class)

Chromosome Representation	Permutation of the 1..N Packages
Fitness Function	Use first fit, next fit, or best fit to minimize the number of bins
Min or Max Problem	Minimization

Use the following to construct possible datasets: 8,2 7,3 6,3,1 3,3,2,2 5,4,1 4,4,2 etc.
12 bin problem: make package sizes using the above values (twice and in random order)
Larger contrived datasets are easy to generate.

##########

GRAPH BISECTION PROBLEM (GPB)

also called **MIN-CUT** or **MINIMUM BISECTION (C/J p. 209)**

The graph bisection problem partitions the vertices of an undirected weighted graph into two equal sized sets of vertices such that the weight of the edges between the two sets is minimized.

Chromosome Representation	Binary string for the nodes (Some may be infeasible)
Fitness Function	Several discussed below
Min or Max Problem	Minimization weighted Cut

What if we decided to make all strings feasible, and not worry about how to handle infeasible strings in the fitness function? How to make every string feasible:

1. Randomly switch zero's to ones or visa-versa to produce 50% of each set.
2. Apply some evolutionary pressure: For example suppose we have too many ones. Pick a small number of ones, say 3 or 4, randomly and switch one of them based on best resulting fitness

What if we consider infeasibles:

Possible fitness could be Weight + |No. 1's – No. 0's| * Factor * Weight
Factor is typically between zero and one.

Another possible fitness could be Weight + |No. 1's – No. 0's| * Factor
This time Factor would be greater than or equal to 1.

See Weighted Max Cut problem for possible contrived datasets only with minimum cut.

##########

SET COVERING (C/J p. 223) **(See the Set Covering Handout)**

See handout for a description of this problem. We will discuss this problem in detail in class.

Chromosome Representation	Binary Strings selecting columns (some strings are infeasible)
Fitness Function	minimize the cost for a covered set
Min or Max Problem	Minimization

How to distinguish among the infeasible solutions?
Perhaps by the number of rows not covered.
Suggestions for fitness of infeasibles: A + B * (number of rows not covered)

1. A is the max value (cost) in the population among feasibles, and B is a small factor
2. A is the actual cost, and B is a large factor to make infeasibles worse than feasibles

Discuss Contrived datasets in class.

##
PACKAGE PLACEMENT PROBLEM or **OPTIMAL LINEAR ARRANGEMENT**
(C/J p. 200) **(See Tutorial II Handout, page 14)**
See handout for a description of this problem. We will discuss this problem in detail in class.

If you select this problem you are to solve the 2-dimensional PPP, described in class

Chromosome Representation	Permutation 1..N of the packages (all feasible)
Fitness Function	minimize the cost of the placement
Min or Max Problem	Minimization

Discuss possible contrived datasets.

##
INDEPENDENT SET PROBLEM (C/J p. 194-5) **(See Yaser Paper)**
In a graph, G = (V,E) find the largest independent set, k (k <= |V|).
An independent set is a subset V' $\subseteq$ V
Such that |V'| = k and such that no two vertices in V' are joined by an edge in E.

Binary Strings representing candidate vertices for the independent set Some not feasible

Max independent set or not. Or how close?

Chromosome Representation	Binary Strings selecting the Independent Set (some strings are infeasible)
Fitness Function	maximize the number of nodes in the set
Min or Max Problem	Maximization

How to use infeasibles. How close is the chromosome to an Independent Set?

How to generate graphs randomly? Contrived?

##########

GEOMETRIC CONNECTED DOMINATING SET PROBLEM (C/J p. 219) (Yaser)

Given a set of points, P, in a plane and a constant, B. Find the minimum K such that P' $\subseteq$ P with |P'| = K and such that all points P – P' are within Euclidean distance B of some point in P', and such that the graph G = (P',E), with an edge between two points in P' if and only if they are within distance B of each other and connected.

Binary String representing the candidate set of points, P'
Not all feasible

Minimize K

Chromosome Representation	Binary Strings selecting points in P' (some strings are infeasible)
Fitness Function	minimize K for a legal connected Dominating Set
Min or Max Problem	Minimization

##########

THE CAPACITATED K-CENTER PROBLEM (See Yaser paper)

The capacitated k-center problem is fundamentally a facility location problem, where we are required to locate *k* facilities in a graph, and assign the other vertices to these facilities, in order to minimize the maximum distance from a vertex to the facility to which it is assigned. In addition, each facility can be assigned at most *L* vertices. The capacitated k-center problem is defined as follows: given an edge-weighted graph *G = (V, E)* find a subset *S* of *V* with size at most *k* such that each vertex in *V* is "close" to some vertex in *S*. In addition, each vertex in S cam be assigned at most L vertices. More formally, the objective function is defined as follows:

$$\min_{S \subseteq V} \max_{u \in V} \min_{v \in S} d(u,v)$$ where d is the distance function.

As an application, one may wish to install *k* fire stations and minimize the maximum distance (response time) from every location (at most L) to its closest fire station.

Chromosome: Consider a Binary String representing the vertices in V'.
Must be exactly P one bits in every chromosome. So make all chromosomes feasible or penalize infeasibles.
How do we assign the remaining vertices to V-S to the vertices in S?

Variation: Consider the chromosome to be a permutation of the vertices 1..N.
How would this work?
How about the first *k* vertices belong to S. How do we assign the rest of the vertices?

##########

THE P-MEDIAN PROBLEM (research paper by Correa, Steiner, Freitas, Carnieri)
(See Yaser Paper)

Assume all vertices of a graph are potential medians, the P-median problem is defined as follows: Let G = (V,E) an undirected graph, where V are the vertices, and E are the edges, and P is some constant less than |V|. The goal is to find a set of vertices V' $\subseteq$ V (median set) with |V'| = P such that the sum of the distances between each of the remaining vertex in V-V' (demand set) and its nearest vertex in V' is minimized.

Chromosome: Binary String representing the vertices in V'.
Must be exactly P one bits in every chromosome.
So make all chromosomes feasible or penalize infeasibles.
Problem: How do we assign the remaining V-V' vertices to vertices in V'?
Goal: Minimize sum of distances

Variation: Capacitated K-Center problem (above)
Capacitated P-Median problem

##########

THE MINIMUM VERTEX COVER PROBLEM

A *vertex cover* of a graph is a set of vertices that touches every edge in the graph. The Minimum Vertex Cover Problem is to find the smallest vertex cover in a given graph (smallest set of vertices that cover the graph). Note this is similar to the Traveling Tourist Problem.

Chromosome Representation	Binary Strings selecting nodes in Vertex cover (some strings are infeasible)
Fitness Function	minimize the number of nodes in the Vertex Cover
Min or Max Problem	Minimization

##########

RURAL POSTMAN PROBLEM (C/J p. 213) **(See Tutorial II, page 30)**

Given a weighted graph G = (V, E) and E' $\subseteq$ E. Find the minimum length path that includes every edge in E'.

##########

THE TRAVELING TOURIST PROBLEM (See Yaser paper)

The traveling tourist problem is defined as follows: Given a graph *G = (V, E)* find the shortest walk visiting a subset of vertices, such that each vertex is either visited, or has at least one of its neighbors visited. For example, the vertices of the graph could correspond to attractions a tourist might like to see, and an edge between two vertices denotes visibility of one attraction from another. The shortest such walk would guarantee that the tourist sees all points of interest. This problem is closely related to the geometric connected dominating set problem. The difference is that we are not only looking for a connected subset that covers all of the points, but also we are looking for the shortest connected subset that cover all the points. Note this is similar to the Minimum Vertex Cover Problem.

Chromosome Representation	Binary Strings selecting the subset of vertices (some strings are infeasible)
Fitness Function	minimize the distance of the tourist tour
Min or Max Problem	Minimization

##########

THE GENERALIZED TRAVELING SALESMAN PROBLEM (See Zimple paper)

"The Generalized Traveling Salesman Problem (GTSP) is a modification of the Traveling Salesman Problem in which nodes are partitioned into clusters and exactly one node from each cluster is visited in a cycle. It has numerous applications, including airplane routing, computer file sequencing, and postal delivery". Source: Part of the abstract from " The Generalized Traveling Salesman Problem: A New Genetic Algorithm Approach" by John Silberholz and Bruce Golden, Springer, 2007.

How would you set this problem up as a GA?

Yellow | Green | Tan | Orange | Purple

					1							1				1				1											1	

9*6*5*7*8 = 15,120. This represents Combinations. Must order now! to make Permutations

Datasets of
20 pairs $2^{20} = 10^{6}$
50 pairs $2^{50} = 10^{15}$
20 triples $3^{20} = 10^{9}$
40 quads $4^{40} = 10^{24}$
50 quintet $5^{50} = 10^{34}$

100 clusters-random 2..6 in each cluster $\sim 4^{100} = 10^{60}$

This is the number of Combinations. Must consider all Permutations of each Combination!
Consider an extension of the chromosome that contains the order of the clusters

4	**1**	**5**	**3**	**2**

What does this mean for crossover and mutation operators?