Mark Demore, 2d Lt

CSCE686 – Dr. Lamont

Homework 8

1. ***Problem Domain Requirements Specification form:***

- domains, D

input Di - Graph G(X,Γ), X:locations. Γ: weighted vertex link set (cost); Set S(W,V), W: item weight, V: item value

output Do – Set of sets(R,L), R:route for each plane, L: load for each plane

- I(x); input conditions on input domain satisﬁed; x in X, link in Γ, set S

- O(x,z); output conditions on output/input domain satisﬁed; i.e.,

a feasible/optimal solution with respect to the input domain   
-- all x assigned  
-- max V (total value)

-- no *wp >* W (max weight)  
-- min C (total cost)

***Algorithm domain requirements specification form:***

* Name: stochastic-search genetic algorithm
* Domains: Ds is a set of satisfying solutions-a population; the population size n is the cardinality of Ds
* Operations:

I(x); x in Ds; x is a possible solution from population

O(x,z); x in Ds, z in Ds; z is a satisfying solution

***Algorithm domain design specification form:***

* Name: stochastic-search-ga(Ds)
* Domains: Di is set of algorithm-internal solutions, Ds is a set of satisfying solutions
* Imports: ADT set, list, real/integer/character
* Initialization of feasible solutions -> Ds; Di empty
* Operations I(x); x in Ds

O(x,z); x in Ds, z in Ds; condition on z being a satisfying solution

* Next-solution-generator -> x for x in Ds, Ds->Di
  + Recombination(crossover) x->y with crossover probability
  + Mutation x->y with mutation probability
  + Feasibility(y) -> Boolean [if true union(y,Di)] ‘genotype’
* Fitness/objective function mapping f(x) of each x in Di ‘phenotype’
* Selection Di -> Ds using f(x) as criteria, x in Di
* Axioms:

***Algorithm domain function specification form:***

* Function stochastic-search-ga(Ds)
* Initial condition: generate feasible Dinitial -> Ds, Di empty, pc, pm
* Body
  + - While not time/generation termination do ss-ga loop:
      * Next-state solution/population Ds, Ds->Di; do for each x in Ds, size n
        + Crossover(x) = y with pc
        + Mutation(x) = y with pm
        + If feasibility(y) then union (y, Di) -> Di
      * Fitness calculation f(x) for each x in Di
      * Selection(di) -> Ds based upon f(x), x in Di
    - End ss-ga while loop
    - Find optimal z in Ds
    - END function

***algorithm domain intermediate speciﬁcation form: (iterative)***

*• Heuristics: distance to next airdrop location, value of load item added*

*• Data structures: input – graph: set of nodes (locations), set of edge weight (cost between each location), set of items weight and value, set of planes with max weight; output – list of sets (route for each plane, and load for each plane)*

***algorithm domain function speciﬁcation form: (iterative)***

*•*  Function ss-ga(initial, Expand, Goal, Cost, Heuristic, crossover, mutation)

q <- New-Priority-Queue()

Insert (initial, q, Heuristic(initial))

**while** generation limit not reached

**do** current <- Extract-Min(q)

crossover(x) = y with pc

mutation(x) = y with pm

**if** Goal(current) then **return** solution

**for** each next in Expand(current)

**do** Insert (next, q, Cost(next) + Heuristic(next))

return failure

The art of the genetic algorithm is that it improves over each iteration through ‘natural selection’ using the metaheuristic.

1. Talbi’s exercise 3.8 has complexity dependent on the input size because the selection mechanism is allowed to select a particular number of individuals from the population and is thus dependent on the combination of population size and permissible selections. Exercise 3.9 has a solution of simply O(n) because it make use of stochastic uniform sampling selection and would be directly related to the number of individuals permissibly selected.

**References:**

[1] <https://www.geophysik.uni-muenchen.de/~igel/downloads/inviigenetic.pdf>

[2] <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>

[3] <https://en.wikipedia.org/wiki/Genetic_algorithm>

[4] <http://www.cs.ucc.ie/~dgb/courses/tai/notes/handout12.pdf>

[5] <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>