Name: Kush Kapur User-ID: ccbd24

Algorithm A: Ant Colony Optimisation

Algorithm B: Genetic Algorithm

Description of enhancement of Algorithm A:

*Instead of initialising best as random or infinite, they were initialised as best obtained from nearest neighbour algorithm, meanings less comparisons and decreased computational expense. Max and Min values for the pheromone levels were introduced. With this cap a path’s pheromone levels never drop so low that path became relevant, and levels never went too high where a path overshadows others(local minimum). Also, pheromone smoothing was added to this , based on the variable ‘stagnation’ measuring when one or more paths were nearing maximum levels and rest were nearing minimum, after which smoothing is performed to promote low level paths, and to maintain standings between paths. Genetic was incorporated into the algorithm using mutation and a greedy crossover function to create a new population from best performing ants(significant with 550 cities especially), where mutation improved the local search ability of the algorithm, by maintaining the diversity of the population. This new population generated by the genetic aspect was used after each iteration in the new improved ‘initialiseants’ function, in which the top 20% performing routes(ants) were preserved and rest initialised again with probability calculated from pheromone levels, creating a great balance between route creation from pheromone levels and breeding. More importantly it created competition between generations and preserved beneficial genes. Aspects of genetic were improved like crossover where if parent1 = parent2, parent2 was mutated. This improved local search ability of the algorithm as it ensured diversity of genetics. Finally, as calculating stagnation was expensive, it was calculated every 8th iteration to improve time-complexity. Pheromone Deposit function only deposits pheromones on edges which exist within the local best tour for every iteration, andbesttour for every 10th iteration, acting as a guide.*

*One of the drawbacks of normal my basic GA was the randomness of mutation, as it was applied to all chromosomes regardless of their fitness, hence some great chromosomes were being disrupted, and as bad chromosomes were unlikely to produce better chromosomes through crossover, they would be the ones benefitting the most from mutation. Hence, chromosomes with fitness < threshold(average fitness) had greater mutation probability, and vice versa .Mutating based on fitness improved the local search ability of the algorithm, by ensuring diversity in population, and improving bad chromosomes(proved very useful in crossover). The original crossover function was very computationally expensive, and inefficient(e.g. gene repetition which then need to be resolved(computationally expensive), inability of parent excellent gene offspring to retain), hence an improved greedy crossover was used with a quick complexity time of O(n), where using a doubly linked construction of parent 1 and 2, returns the best possible child. This had a significant improvement in runtime and performance of my algorithm. I made a further improvement in my crossover function by including a condition in which if Paren1 = Parent 2, parent 2 is mutated. This led to significant improvements because previously GA would converge too early, and would result in the same parents crossing, decreasing genetic diversity and arriving at a local minimum, however this was almost a local search modification for my algorithm and helped me reach a global minimum. Also, new population is now constructed by combining the top half of the initial and new population, to increase diversity of chromosomes and increase competition between generations, leading to better retention of superior genetics from each generation.*

Description of enhancement of Algorithm B:

***DESCRIPTION OF ALGORITHM ONLY IF THE ALGORITHM IS NOT COVERED IN LECTURES***

Description of *non-standard* Algorithm A:

*Describe any non-standard algorithms you have implemented that* ***have not been covered in lectures*** *(otherwise these boxes should be blank) You need to convince me that your implementation is indeed that of the named algorithm and you need to* ***provide a full reference to the source for your algorithm****. You should* ***include a pseudocode description****. You can vary the sizes of these boxes but not the font (Calabri), font size (11) or paragraph properties (single space), and everything should fit onto one side of A4. (You can delete these instructions.)*

Description of *non-standard* Algorithm B:

*Type here.*