**University of Waterloo**

**Department of Electrical and Computer Engineering**

**ECE 457A: Cooperative and Adaptive Algorithms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Title** | Feature Selection using Simulated Annealing | | |  |
| **Group Number** | 13 | | |
| **Team Member** | *Tian Zhu* | *20517833* | *t27zhu@edu.uwaterloo.ca* |
| *CongHao He* | *20589010* | *chhe@edu.uwaterloo.ca* | *10*  *points* |
| *Yiran Tian* | *20580120* | *y62tian@edu.uwaterloo.ca* |
| *YuCheng Liu* | *20558944* | *y698liu@edu.uwaterloo.ca* |

**[2 points] Algorithm and Problem:**

Discussing the suitability of search algorithm as it relates to the problem at hand as well as the algorithm complexity

The suitability of simulated annealing for the feature selection problem is better than that of greedy algorithm (eg. hill climbing), but worse than that of Tabu search which is more robust due to the possibility of implementing penality and constraints. Simulated annealing finds the local best solution and will jump to neighbor with a probability, due to the cost function difference between 2 solutions and temperature of the system. When the neighbor solution is better, it will always jump; if the next one is worse, it will jump with possibility of exp(dE/kT) (dE difference between cost functions; T temperature; k constant = 1).

The time complexity of SA is determined by the initial temperature, temperature decrease step and the iterations per step. There are in total of n possible neighbours, formulated by including or excluding one of the n features. The space complexity is O(n), which is the total number of features available for selection, since only current accepted solution is stored.

**[2 points] Algorithm Scalability:**

Discuss what will happen if you increase your search space.

Assume that our annealing schedule is dependant on the number of sample. As the sample increases, there may be more iterations and more element to search as the space is broader. For each iteration, the number of neighbours of the current solution also increase, which means for our feature selection problem, when our temperature is high, we will have a high change exploring all possible feature without considering the increase in the cost function. Also, the initial temperature could be set higher to accommodate bigger search space. Therefore, the time overall running time of the algorithm would be higher. However, with the temperature limit, the SA algorithm guarantee us that we will reach a solution in a controlled time, this is very important to our problem, because for the feature selection problems, as the number of feature increase, our runtime usually increase exponentially. Otherwise, if the cooling schedule and the initial temperature is not changed, search time will remain the same.

We can also scale the search algorithm to increase efficiency. This is using cooperative SA. Multiple threads can explore multiple regions of solution.

**[4 points] Results:**

Share the results achieved by the algorithm w.r.t. your problem and dataset. Comments on the solution(s) optimality.

Final solution of SA:

[

('Current', ['radius\_mean', 'perimeter\_mean', 'compactness\_mean', 'concave points\_mean', 'texture\_se', 'area\_se', 'smoothness\_se', 'concave points\_se', 'symmetry\_se', 'radius\_worst', 'texture\_worst', 'compactness\_worst', 'fractal\_dimension\_worst'], 0.9230769230769231),

('Best', ['radius\_mean', 'perimeter\_mean', 'area\_mean', 'perimeter\_se', 'area\_se', 'smoothness\_se', 'concavity\_se', 'concave points\_se', 'symmetry\_se', 'fractal\_dimension\_se', 'texture\_worst', 'compactness\_worst', 'concavity\_worst', 'concave points\_worst', 'symmetry\_worst'], 0.9790209790209791)

]

Our problem is to select a set of features that achieve best accuracy of outcomes from all the features provided. For the Simulated Annealing algorithm, the Wisconsin Breast Cancer Dataset is selected to perform the test. This dataset has 30 features, and 2^30 possible solutions are available.

The optimality of the solution depends on three important factors: initial temperature, number of iteration per temperature and the cooling schedule. When initial temperature is higher and with more iterations per step, the algorithm will be more optimal since it could explore more space during high temperature, thus it’s less possible to be trapped at local best solutions.

Overall, the optimality of SA is medium high, as it could reach the one of the absolute optimal solutions with a certain probability, but it could also stuck on local solutions and might not have a thorough view of search space. Therefore, it is not guaranteed to find the optimal solution using SA. SA could find a good solution in relatively short time.

The original SA algorithm explored the search space at first and tried to locate a local maximum solution in the end. In the result data mentioned earlier, when the algorithm terminated, the current solution scored approximately 0.923. This is obviously not the global optimum because we know from the previous algorithms that the global optimum should be at least 0.972. Such local optimal result is acceptable because simulated annealing does not guarantee global optimum result. When the temperature is high, it may attempt to escape from local optimum and tries to find global optimum. However, if in the end by chance it reached a local optimum that is not the global optimum, it will try to locate the exact local optimum. In the case of our results, only a local optimum was found.

Although simulated annealing was only designed to keep the current solution, it would be interesting if we can devote another O(n) space to keep track of the best result so far. This is not possible for actual annealing because at low temperature, particles loses the energy required to go back to the best state and even if there were enough energy, the particles do not have enough information to keep track of the best state. In simulated annealing, however, we can afford to use another O(n) space to keep track of the best solution so far and “go back in time” to that best solution instead of dwelling in the final local optimum. A tracker was implemented to keep track of the most optimized result. For every local best solutions, we compare it with our tracker and if it is better than the recorded most optimized result , then we will update the recorded best solution. The final score that the tracker observed was 0.979. This result is moderately better than the final result achieved by unmodified simulated annealing. This only required a change of from O(n) space complexity to O(2n) space complexity (which strictly speaking is still O(n)).

**[1 points] Performance Comparison [if applicable]:**

Comment on the algorithm performance and the optimality of the solution achieved compared to the results you previously achieved. This section should be used for all reports other than the Graph Search Algorithms.

Final solution of Tabu:

(['texture\_mean', 'perimeter\_mean', 'smoothness\_mean', 'compactness\_se', 'texture\_worst', 'perimeter\_worst', 'smoothness\_worst'], 0.972027972027972)

Final solution of BFS:

['area\_worst', 'compactness\_worst', 'concavity\_se', 'perimeter\_worst', 'radius\_worst', 'texture\_worst'] 97.2%

According to the computing ability of computer, above 2 maximumly ran 10 features at a time.

Final solution of SA:

[

('Current', ['radius\_mean', 'perimeter\_mean', 'compactness\_mean', 'concave points\_mean', 'texture\_se', 'area\_se', 'smoothness\_se', 'concave points\_se', 'symmetry\_se', 'radius\_worst', 'texture\_worst', 'compactness\_worst', 'fractal\_dimension\_worst'], 0.9230769230769231),

('Best', ['radius\_mean', 'perimeter\_mean', 'area\_mean', 'perimeter\_se', 'area\_se', 'smoothness\_se', 'concavity\_se', 'concave points\_se', 'symmetry\_se', 'fractal\_dimension\_se', 'texture\_worst', 'compactness\_worst', 'concavity\_worst', 'concave points\_worst', 'symmetry\_worst'], 0.9790209790209791)

]

The running time of SA is significantly smaller than that of other algorithms an thus it could run all 30 features at same time.

According to original SA, the final solution is 92.3% which is not the best one, and the final solution is worse than other 2 algorithms. Since SA could jump away from the best solution and converge to other local best solutions when cooled down.

After our SA optimization, best solutions is recorded, which is 97.9%, the final solution is the best across all 3 algorithms so far.

*N.B.: Please, note the report should be a maximum of One pages long. Please, submit your report as well as your code to the appropriate dropbox on Learn.  Your code will weigh 1 point of the grade. Please note that you can use existing libraries*