NatAlgReport

Name: Cheng-Han Yang User-ID: zgcs95

Two-letter code for your chosen NatAlgReal algorithm: AB

You have found a minimum value of 1.1102230246251565e-16 and a minimum point of [1.1244146279096368e-08, 3.673180269280689e-10, 3.881890406390699e-09].
You have found a minimum value of 0.0 and a minimum point of [4.72204359222054e-09, 7.533546462459736e-09, 1.713304443376589e-08].
(The elapsed time of both minimums found was 60.0 seconds)

*The minimum found by the enhanced version is either 0 or a number around 10**-16, and the parameters are N = 30, M = 10, num_cyc = 300000, and lambbda = 8. The results generated by the vanilla version are around 0.001, with the same parameters as the enhanced version.*

Enhancement of employed bee phase: (The partner mentioned below means the food source used not being the current food source to calculate the values of near-neighbours)
To avoid the local minimums, I add more randomness while exploring neighbourhoods. In the vanilla version, the employed bee only takes one variable to change and only generates one near-neighbour. The number of generated near-neighbours in the enhanced version depends on the limitation number. As the objective function takes three variables, there are 7 combinations of changing variables, excluding three variables remaining the same. Therefore, a list of near-neighbours will be generated according to the limitation number, and the number of near-neighbours is the limitation number+1. The size of the list of near-neighbours at most 7. For example, if a food source has 3 as the limitation number, the generated list of near-neighbours has 4 different near-neighbours, and then the greedy selection is performed among the food source and all the near-neighbours in the list. In addition, whether these three variables will be changed depends on the binary code of the limitation number, and each variable's partner is different. For instance, if the limitation number is 3. The near-neighbours will be generated based on 001, 010, 011, and 100. The binary code indicates which variable should change or not, and 1 means the variable in that position should be changed. As the variables are presented as a, b, c, and take 011, for example, it means variables b and c should be changed to generate a near-neighbour of the current food source.

Roulette Wheel:
The tournament selection replaces the roulette wheel selection. Suppose N the number of food sources is too large. In that case, the difference between the fitness between different food sources will be reduced, which makes the roulette wheel more random instead of the food source with better fitness having a higher chance to be selected.

The greedy selection is applied in the scout bee phase. Since the employed bee phase has been enhanced to generate more diverse near-neighbours, it is easier for a current food source to find better near-neighbours. It is unnecessary to replace food sources with a new random food source with lower fitness. If this random food source gradually decreases but is still stuck in a local minimum, the computational time will be wasted. Also, greedy selection prevents the problem resulting from the limitation number not being appropriately set. If the limitation number is too low, many food sources might be replaced with random food sources with lower fitness.

(continued over)

Name: Cheng-Han Yang User-ID: zgcs95

Two-letter code for your chosen NatAlgDiscrete algorithm: AB

The chosen question is the largest clique problem, so the food source mentioned below means a set of vertices.

There are two ways to generate near-neighbours of food sources:
The first way to generate a near-neighbour is to save the entire set of vertices and runs through the vertices that do not belong to that current food source to check whether it forms a proper clique with a current food source, and a vertex does not belong to its current food source. If a proper clique is found with a vertex and a current food source, the current food source will be replaced by that proper clique. If a proper clique is not found, the limit number will be added by 1. This process is used in the employed bee and onlooker bee phases.

The second way to generate a near-neighbour is by removing a random number of vertices of the targeted food source. The number of removed vertices ranges from 1 to half the targeted food source's size, meaning at least half the vertices will remain. After removing the vertices, the unselected vertices will be chosen randomly and added to the targeted food source. This newly generated food source is the size of the current food source + 1. Then this newly generated food source will be checked to determine whether it is a proper clique. If so, the current food source will be replaced with this new food source. This process is used in the scout bee phase.

Roulette Wheel:
Like the real value function problem, the roulette wheel selection is replaced by tournament selection to solve the problem of N being too large.

Adjusted Scout Bee Phase:
The second way of generating near-neighbour only happens in the scout bee phase. The second way is done because there is no way to generate a larger clique and save all the vertices in the current food source. Therefore, removing some vertices is needed. If the limitation number of a current clique is over the lambda and the size is n, a clique with n + 1 size will be generated in the second way generating clique. That n+1 clique will be checked to determine whether it is a proper clique. If it is a proper clique, it will replace its current clique.

The parameter of the limitation number:
As mentioned above, each vertex that does not belong to the current food source is checked to determine whether it forms a proper clique with the current food source. Accordingly, the only difference in the limitation number is 0 or not. If a limited number of a current food source is above 0, a potentially better clique will be generated and checked whether it is a proper clique. Additionally, the parameter of limitation number only serves as the threshold of the number of changing variable sets. As mentioned above, there are up to 7 combinations of changing variables as the function takes three variables. For instance, if the limitation parameter is 3, there will only be at most 3 combinations of changing variables.

Greedy Selection:
Greedy selection is removed because of the discretization methodology. A current food source is only replaced in all three phases if a larger and proper clique is found. Otherwise, the limitation number is increased by 1.