Universidade Federal do Ceará
Disciplina: Inteligência Computacional
Professor(a): Guilherme Barreto and Ajalmar
Estudante: Rubem Vasceconcelos Pacelli Matrícula: 519024

August 10, 2022

Homework 03 - Metaheuristic

A metaheuristic is a high-level problem-independent algorithmic framework arisen in the numerical optimization field that provides a set of guidelines or strategies to develop a problem-dependent heuristic (a partial search algorithm) solution, providing sufficiently good results to a given optimization problem [1]. Heuristic solutions are attractive when

- The classical methods are too slow or fail in finding the global solution.
- The computational capacity is limited or unaffordable.
- There is incomplete or imperfect information.

The main drawback is that we cannot guarantee that the global optimal solution is found using heuristic methods as they only sample a subset of solutions that is otherwise too large to be completely explored. That being said, the main goal of the metaheuristic is to efficiently explore the search space in order to find a near-optimal solution for nonspecific problems.

In this homework, we will focus on particle swarm optimization (PSO), which is a method from the natured-inspired metaheuristic branch.

Let us define

$$\mathbf{x}_i^{(m)} = \begin{bmatrix} x_1^{(m)} & x_2^{(m)} & \cdots & x_D^{(m)} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^D, \tag{1}$$

and

$$\mathbf{v}_i^{(m)} = \begin{bmatrix} v_1^{(m)} & v_2^{(m)} & \cdots & v_D^{(m)} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^D, \tag{2}$$

as the ith particle position and its velocity at the instant m, respectively, where D is the length of the solution and $i \in \{1, 2, \cdots, I\}$. The Global Random Search (GRS) is a PSO algorithm that explores both the cognitive component of the individual experience and the social component to update $\mathbf{x}_i^{(m)}$ and $\mathbf{v}_i^{(m)}$. The recursive equations are given by

$$\mathbf{v}_i^{(m+1)} = w\mathbf{v}_i^{(m)} + c_1\mathbf{r}_1 \odot \left(\mathbf{p}_i^{(m)} - \mathbf{x}_i^{(m)}\right) + c_2\mathbf{r}_2 \odot \left(\mathbf{g}^{(m)} - \mathbf{x}_i^{(m)}\right) \tag{3}$$

and

$$\mathbf{x}_i^{(m+1)} = \mathbf{x}_i^{(m)} + \mathbf{v}_i^{(m+1)}, \tag{4}$$

where w is the inertia parameter, c_1 and c_2 are the coefficient accelerators, $\mathbf{r}_1$ and $\mathbf{r}_2$ are constant vectors whose values come from the sampling of a random process with a uniform distribution between 0 and 1, $\mathbf{p}_i^{(m)}$ is the best position of the ith particle, $\mathbf{g}^{(m)}$ is the best global position, and $\odot$ denotes the Hadamard operator.

The best individual and global positions are defined based on a cost function, which is defined by the problem. For this homework, we will use the data set shown in Figure 1.

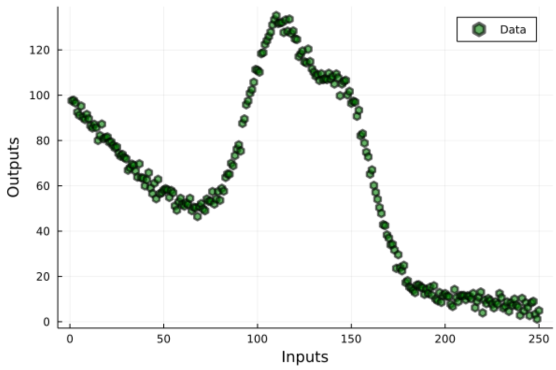

Figure 1: Scatterplot of the residuals.

The goal is to find the coefficients of the k-order polynomial function, that is,

$$y(n) = a_0 + a_1 x(n) + a_2 x^2(n) + \cdots + a_k x^k(n), \tag{5}$$

where $x(n)$ is the nth dataset sample. Hence

$$\mathbf{x}_i^{(m)} = \begin{bmatrix} a_0^{(m)} & a_1^{(m)} & \cdots & a_k^{(m)} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{k+1}. \tag{6}$$

The cost or objective function is given by

$$J_i^{(m)} = \sum_{n=1}^{N} e_i^2(n), \tag{7}$$

where N is the length of the dataset and $e_i^{(m)} = y(n) - \hat{y}_i(n)$ is the residual error, being $\hat{y}_i(n)$ the output value of the polynomial curve when it is used $\mathbf{x}_i^{(m)}$ as the set of parameters and $y(n)$ the result of the 6-order polynomial least squares[1].

The Algorithm 1 summarizes the GRS method, where the stop criteria is iterating over the dataset N_i times. Moreover, the inertia parameter, w, is initialized at a high value ($\simeq 0.9$) and is gradually decreased to its minimum value ($\simeq 0.4$), making the GRS

[1]The value of $k = 6$ was defined based on the results achieved by the polynomial regressor.

algorithm vary from exploring[2] to exploiting[3], respectively. The final output is the global best solution found by the swarm.

Algorithm 1: Global random search

Input: $D, J(\cdot), I, N_i, c_1, c_2$

Output: $\mathbf{g}^{(m)}$

Data: $\{x(n)\}_{n=0}^{N}$

$\mathbf{x}_i^{(m)}, \mathbf{v}_i^{(m)}, w \leftarrow$ Initialize

$\mathbf{p}_i^{(m)} \leftarrow \mathbf{x}_i^{(m)}$

forall $m \in \{1, , \cdots, N_i\}$ **do**

  forall $i \in \{1, , \cdots, I\}$ **do**

    $\mathbf{r}_1, \mathbf{r}_2 \leftarrow$ sample it from $\sim U(0,1)$

    $\mathbf{v}_i^{(m+1)} \leftarrow w\mathbf{v}_i^{(m)} + c_1\mathbf{r}_1 \odot \left(\mathbf{p}_i^{(m)} - \mathbf{x}_i^{(m)}\right) + c_2\mathbf{r}_2 \odot \left(\mathbf{g}^{(m)} - \mathbf{x}_i^{(m)}\right)$

    $\mathbf{x}_i^{(m+1)} \leftarrow \mathbf{x}_i^{(m)} + \mathbf{v}_i^{(m+1)}$

    $J_i^{(m+1)} \leftarrow \sum_{n=1}^{N} e_i^2(n)$

    if $J_i^{(m+1)} < J_i^{(m)}$ **then**

      $\mathbf{p}_i^{(m+1)} \leftarrow \mathbf{x}_i^{(m+1)}$

    else

      $\mathbf{p}_i^{(m+1)} \leftarrow \mathbf{p}_i^{(m)}$

    if $J_i^{(m+1)} < J_j^{(m)}\ \forall j \in \{1, 2, ..., I\}, j \neq i$ **then**

      $\mathbf{g}^{(m+1)} \leftarrow \mathbf{x}_i^{(m+1)}$

    else

      $\mathbf{g}^{(m+1)} \leftarrow \mathbf{g}^{(m)}$

return $\mathbf{g}^{(m)}$

The same process was repeated M times. Unfortunately, the heuristic process could not to find a reasonable solution for this problem of any independent run, as shown in Figure 2. The value of the coefficients also vary with the independent realizations.

[2]The exploration phase where the algorithm tries to follow different paths in order to explore and space search and in better solutions.

[3]Is the phase where the algorithm becomes more conservative in the sense that it tends to follow the same direction.

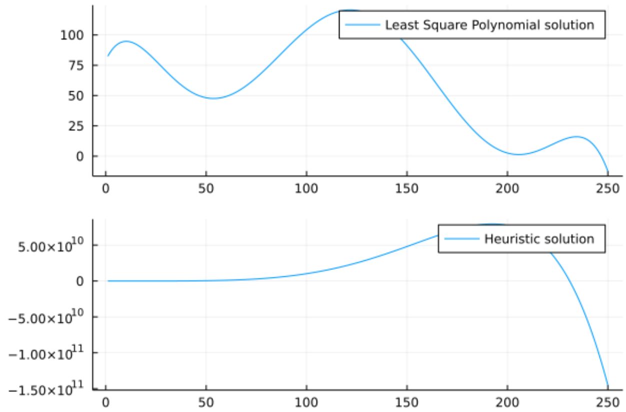

Figure 2: result of the heuristic regression.

References

[1] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Chapter 10 - metaheuristic algorithms: A comprehensive review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications* (A. K. Sangaiah, M. Sheng, and Z. Zhang, eds.), Intelligent Data-Centric Systems, pp. 185–231, Academic Press, 2018.