Universidade Federal do Ceará

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

August 11, 2022

Homework 03 - Metaheuristic

In the optimization theory field, a metaheuristic is a problem-independent algorithmic methodology that provides a set of guidelines to develop a problem-dependent heuristic 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 will be found by using heuristic methods since they only sample a subset of the solution space. This subset is called search space, and a partial search algorithm is utilized to find a heuristic solution. 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 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.

Vectors $\mathbf{x}_i^{(m+1)}$ and $\mathbf{v}_i^{(m+1)}$ are limited to the range $[\mathbf{x}_l, \mathbf{x}_u]$ and $[-|\mathbf{x}_u - \mathbf{x}_l|, |\mathbf{x}_u - \mathbf{x}_l|]$, respectively, where $\mathbf{x}_l$ and $\mathbf{x}_u$ contains the elements of the lower and upper bounds of the search space.

For this homework, we will use the data set shown in Figure 1.

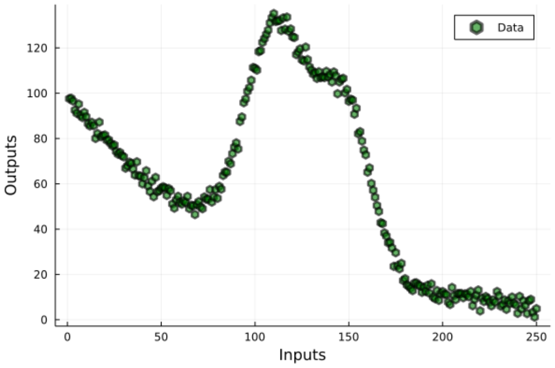


Figure 1: Scatterplot of the residuals.

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 and $k = 6$[1] is the polynomial function order. 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 best individual and global positions are defined based on a cost function, which is defined by the problem. The cost or objective function of this problem 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 $e_i(n) = y(n) - \hat{y}_i(n)$ 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.

The Algorithm 1 summarizes the PSO 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 reasonable 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.

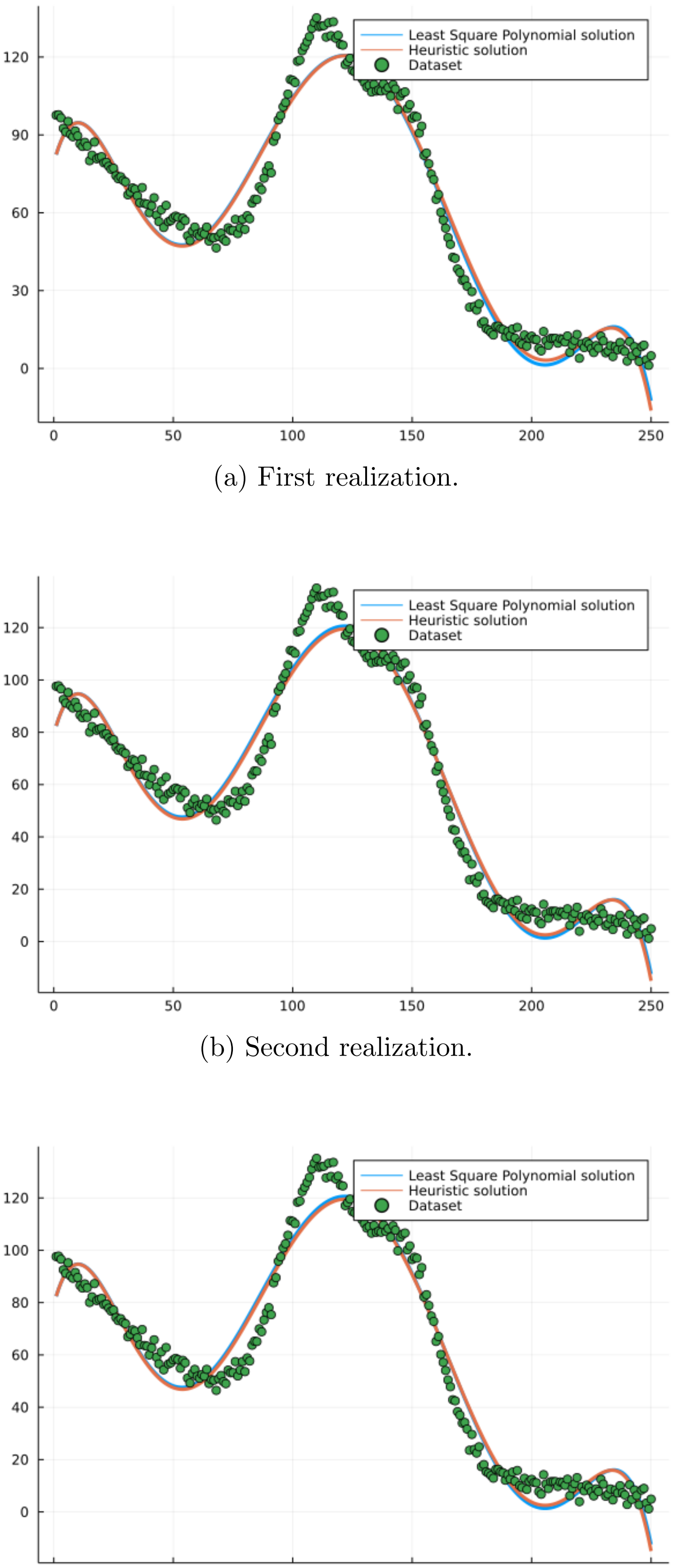
Algorithm 1: Particle Swarm Optimization

```
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 $\sim U(0,1)$
        $\mathbf{v}_i^{(m+1)} \leftarrow w\mathbf{v}_i^{(m)} + c_1\mathbf{r}_1 \odot \left(\mathbf{p}_i^{(m)} - \mathbf{x}_i^{(m)}\right) + c_2\mathbf{r}_2 \odot \left(\mathbf{g}^{(m)} - \mathbf{x}_i^{(m)}\right)$
        $\mathbf{v}_i^{(m+1)} \leftarrow$ limit it to $[-|\mathbf{x}_u - \mathbf{x}_l|, |\mathbf{x}_u - \mathbf{x}_l|]$
        $\mathbf{x}_i^{(m+1)} \leftarrow \mathbf{x}_i^{(m)} + \mathbf{v}_i^{(m+1)}$
        $\mathbf{x}_i^{(m+1)} \leftarrow$ limit it to $[\mathbf{x}_l, \mathbf{x}_u]$
        $J_i^{(m+1)} \leftarrow \sum_{n=1}^{N} e_i^2(n)$
        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. The result of the curve fitting is shown in Figure 2 for three 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.



(a) First realization.

(b) Second realization.

(c) Third realization.

Figure 2: result of the curve regression through PSO.

The value of the coefficients also varies with the independent realizations. Notwithstanding, these values are relatively close to the best solution. For comparison, the best solution is

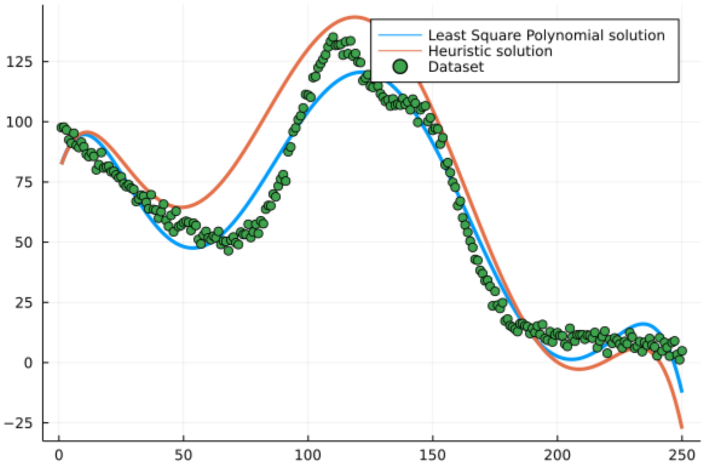
$$\mathbf{x}_o = \begin{bmatrix} 79.2057 \\ 3.44317 \\ -0.230983 \\ 0.0045354 \\ -3.70951.10^{-5} \\ 1.34374.10^{-7} \\ -1.79008.10^{-10} \end{bmatrix}, \tag{8}$$

while the solution found by the PSO is

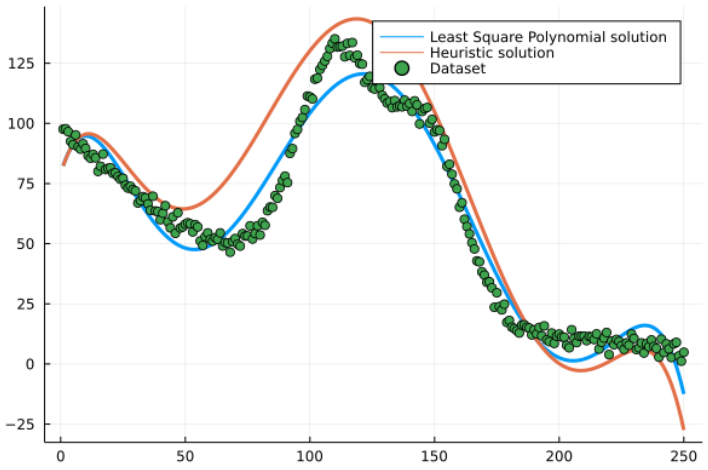
$$\mathbf{g} = \begin{bmatrix} 79.2057 \\ 3.44607 \\ -0.221978 \\ 0.00442822 \\ -3.67436.10^{-5} \\ 1.34206.10^{-7} \\ -1.79343.10^{-10} \end{bmatrix} \tag{9}$$

.

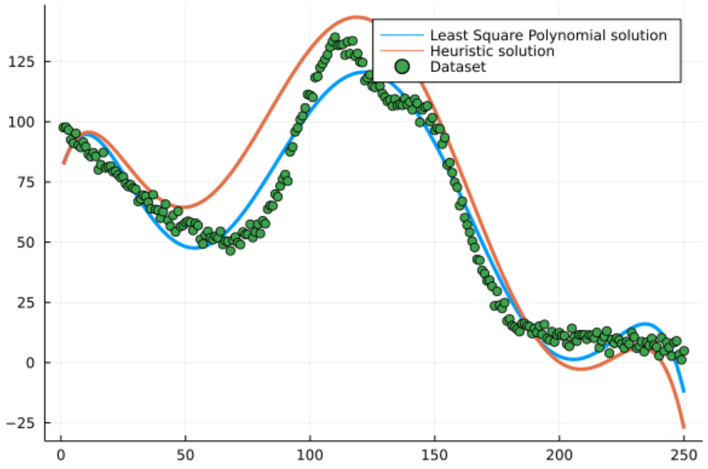
The Global Random Search (GRS), which is a nonnature-inspired metaheuristic technique, could achieve similar results, as shown in Figure 3.



(a) First realization.



(b) Second realization.



(c) Third realization.

Figure 3: result of the curve regression through GRS.

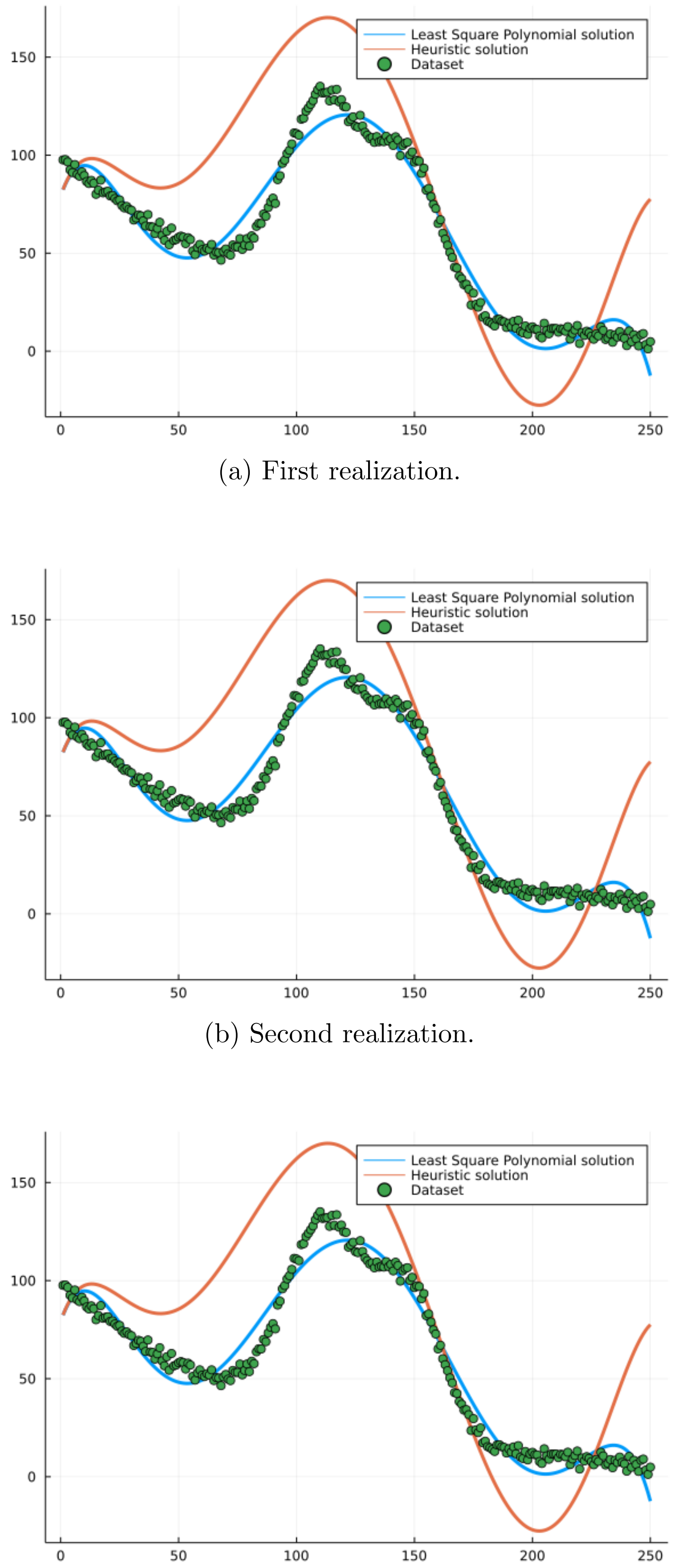
Despite the good results, both PSO and GRS are very sensitive to the search space range. In other words, they only achieve reasonable results when the search space is very close to the best solution. Otherwise, the best solution can differ significantly from a reasonable result.

Although the PSO is much more costly than the GRS, the final results suggest that the swarm approach outperforms the individual random search method.

For the following cost function

$$J_i^{(m)} = \sum_{n=1}^{N} |e_i(n)| \,, \tag{10}$$

The results of the curve fitting are shown in Figure 4.



(a) First realization.

(b) Second realization.

(c) Third realization.

Figure 4: result of the curve regression through PSO.

These results are significantly worse than the squared value.

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.