UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
University of Pretoria

Artificial Intelligence
COS710

Assignment 1: Particle Swarm Optimization

Due date: 2 March 2018, at 23:59

Do the assignment described below. Note that you have to submit a pdf document, containing a technical report wherin you describe what you have done, and present and discuss your results. Remember that you can not report results for only one run of an algorithm on a problem, but you have to report results as averages and standard deviations over a sufficient number of independent runs of the algorithms. I usually use 50 independent runs. When you want to determine if one approach is better than another, the outcome of such a comparison should be based on a formal statistical test, such as the Mann-Whitney U test.

Boundary Constraint Handling Mechanisms

It has been shown empirically by Engelbrecht (see `roamingPSO.pdf`) in the assignment 1 folder on the COS710 website) that most of the particle of PSO algorithms leave the search space within the first iterations. The disadvantages if this roaming behavior have been discussed by Engelbrecht. Boundary constraint handling mechanisms have been developed to prevent particles from leaving the search space, or to at least prevent personal and neighborhood best positions to leave the search space. For this assignment you will implement and analyze some boundary constraint handling mechanisms, and some variations to them. As part of the study, you will also investigate how performance under these boundary constraints is affected for larger dimensions.

For all the implementations, choose a global PSO, where the global best position is selected as the best personal best position from the first iteration. That is, use a memory based approach. You can also simply implement a synchronous update strategy.

The boundary constraint handling mechanisms that you will implement and evaluate are the following:

- Update the personal best positions only if the new particle position is better than its current personal best position, and if the new particle position is feasible. That is, a new particle position can not become a personal best position if it violates boundary constraints.
- Clamping approach: If a particle violates a boundary constraint in a specific dimension, then clamp the corresponding decision variable at the boundary value. That is, if $x_{ij}(t+1) < x_{min,j}$ then $x_{ij}(t+1) = x_{min,j}$, or if $x_{ij}(t+1) > x_{max,j}$ then $x_{ij}(t+1) = x_{max,j}$.

- Per element reinitialization: For any decision variable of any particle that violates a boundary constraint, reinitialize that decision variable to a random position that satisfies the boundary constraints. That is, if $x_{ij}(t+1) < x_{min,j}$ or $x_{ij}(t+1) > x_{max,j}$, then $x_{ij}(t+1) \sim U(x_{min,j}, x_{max,j})$.
- Per element reinitialization and setting velocity to zero: Adapt the per element reinitialization approach above to also set the velocity of the decision variable that violates a boundary constraint to zero. The corresponding decision variable's new position will therefore not be influenced by the momentum term.
- Initialize to personal best position: Initialize the boundary violating decision variable to the corresponding personal best position. That is, if $x_{ij}(t+1)$ violates a baoundary constraint, then $x_{ij}(t+1) = y_{ij}(t)$.
- Initialize to personal best position and set velocity to zero: Adapt the intialize to personal best position strategy above to also set the corresponding velocity to 0.
- Initialize to global best position: As for the above, but $x_{ij}(t+1) = \hat{y}_j(t)$ for the boundary violating decision variable.
- Initialize to global best position and set velocity to zero: Adapt the intialize to global best position strategy to also set the corresponding velocity to 0.
- Reverse velocity: The velocity of the bounadry violating decision variable is simply reversed while that decision variable violates the boundary constraint.
- Set the bounadry violating decision variable to an arithmetic average of the corresponding personal best and global best position. That is, $x_{ij}(t+1) = \alpha y_{ij}(t) + (1-\alpha)\hat{y}_j(t)$, where $\alpha \sim U(0,1)$ (randomly selected in the range $(0,1)$).

The above will earn you at most 85% depending on the quality of your report. In order to increase your mark beyond 85%, you can implement any other boundary handling mechanism(s) that you find in the literature.

The questions that you need to answer in your report are the following:

- Which boundary constraint mechanism results in best performance with reference to solution accuracy?
- Which boundary constraint mechanism results in the highest success rate?
- Which boundary constraint mechanism is the most efficient in terms of computational effort to reach certain accuracy levels?
- Which boundary constraint mechanism is the most scalable?
- Where appropriate, which boundary constraint mechanism wastes the least amount of search effort outside of the feasible region as defined by the boundary constraints?

For your empirical analysis, chose at least 10 functions from the uploaded article, `gbestVSlbest.pdf`, of varying characteristics. In your report, motivate your choice of functions. Follow the empirical process described in this article.

Format of your Technical Report

1. Introduction:

The introduction sets the stage for the remainder of your report. You usually have very general statements here. The introduction prepares the reader for what to expect from reading your report. In general, the introduction should either contain or be a summary of your ENTIRE report.

2. Background:

A very high level discussion on the problem domain and the algorithms and/or approaches that you have used. Do not be too specific on the algorithms and approaches. This section is typically where the ”base cases” of concepts that appear throughout the remainder of your report are discussed. It is also an ideal place to refer a reader to other sources containing relevant information on the topic but which is outside the scope of your assignment. It is the perfect place for pseudo code. Remember to discuss very generally. After reading this section the marker should be able to determine whether or not you know what you're talking about.

3. Implementation:

In this section you discuss how you approached, implemented and solved your assignment choice. Mention, for example, the values set for the algorithm's control parameters, how many simulations you have run and what the characteristics for candidate solutions to your problems are. After reading this section (in addition to the background) the reader should be able to duplicate your experiments to obtain similar results to those obtained by you. This is also the section where your discussion specializes on the concepts mentioned in the background section. Be very specific in your discussions in this section.

4. Research Results:

This is the section where you report your results obtained from running the experiments as discussed in the implementation section. You have to give, at least, averages and standard deviations for the experiments/simulations. Thoroughly discuss the results that you've obtained and reason about why you obtained the results that you have. Answer questions like ”are these results to be expected?” and ”why these results occurred?” and ”would different circumstances lead to different results?”.

5. Conclusion(s):

Very general conclusions about the assignment that you have done. This section ”answers” the questions and issues that you've raised and investigated. This section is, in general, a summary of what you have done, what the results where and finally what you concluded from these results. This is the final section in your document so be sure that all the issues raised up until now are answered here. This is also the perfect section to discuss what you have learnt in doing this assignment.