UNIVERSITY OF JOHANNESBURG

FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**IT00107** OPTIMISATION
CAMPUS	**APK**
EXAMINATION	**NOVEMBER, 2016**

DATE 18 NOVEMBER 2016	**SESSION** 8:30 – 10:30
ASSESSOR(S)	DR. GJ VAN NIEKERK
INTERNAL MODERATOR	NONE
EXTERNAL MODERATOR	DR. WJC VAN STADEN (UNISA)
DURATION 2 HOURS	**MARKS** 100

NUMBER OF PAGES: 4 (cover page included)

INSTRUCTIONS
ANSWER ALL THE QUESTIONS
WRITE NEATLY AND LEGIBLY
NUMBER IN SEQUENCE

REQUIREMENTS
NONE

Section A: Evolutionary Algorithms

1. (5 marks) Discuss how the size of the population for an Evolutionary Algorithm should be chosen and the effect that this size has on the overall algorithm.

2. (5 marks) A specific Genetic Algorithm (GA) implementation starts with a very high mutation rate. After each iteration the current mutation rate is doubled if at least half the offspring is better than both parents, otherwise the mutation rate is halved. Discuss whether this approach is viable or not. Clearly motivate your answer.

3. (4 marks) Propose a multi-parent ($n > 2$) version for the uniform crossover operator to produce multiple offspring from the n parents. (Consider using a diagram in your explanation.)

4. (5 marks) Discuss the merits of using the following approach to calculate dynamic strategy parameters for candidate i using Evolutionary Programming (EP):

$$\sigma_i(t) = |f(\hat{\mathbf{y}}(t)) - f(\mathbf{x}_i(t))|$$

   Where $f : \mathbb{R}^{n_x} \to \mathbb{R}$ denotes the fitness function; $\hat{\mathbf{y}}(t)$, the best candidate at time t and $\mathbf{x}_i(t)$, the i^{th} candidate solution at time t.

5. (15 marks) The bounded knapsack problem is a problem in combinatorial optimization: Given a set of k item types, each with a weight and a value, determine the number of each item type (to a maximum n, for each type) to include in a collection so that the total weight is less than or equal to a given limit, say m and that the total value is maximised.

   Discuss how this problem can be solved using a Genetic Algorithm. Clearly elaborate on the **representation scheme**, the **fitness function** and any specifics regarding the reproduction and selection operators.

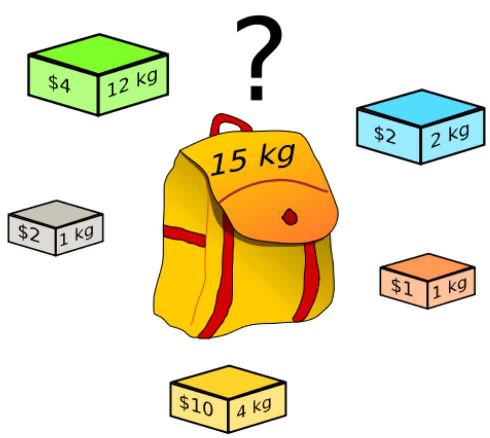

Figure 1: An example of the knapsack problem

Section B: Particle Swarm Optimisation

1. (2 marks) Briefly explain why Particle Swarm Optimisation Algorithms are better suited for use in continuous-valued search spaces than Genetic Algorithms.

2. (3 marks) Velocity clamping can be made dynamic to balance exploration/exploitation better. Explain how this can be done.

3. (5 marks) Briefly explain how the Cooperative Split PSO (CPSO) algorithm calculates its fitness function.

4. (4 marks) Consider two fitness functions $f : \mathbb{R}^{n_x} \to \mathbb{R}$ and $g : \mathbb{R}^{n_x} \to \mathbb{R}$. The function f is computationally expensive, but its results are highly accurate for similar solutions. Function g gives inaccurate results for similar solutions, but is computationally inexpensive. Discuss which of the two should be used.

5. (6 marks) The standard predatory-prey PSO uses the following velocity update rule for a single predator:

$$\mathbf{v}_p(t+1) = \mathbf{r}(\hat{\mathbf{y}}(t) - \mathbf{x}_p(t))$$

   where $\mathbf{v}_p$ and $\mathbf{x}_p$ are the velocity and position of the predator, $\hat{\mathbf{y}}(t)$ the best particle's position and $\mathbf{r}$ controls the speed with which the predator catches up with the prey.

   The velocity of the prey becomes:

$$\mathbf{v}_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_{ij}(t) - x_{ij}(t)) \\ + c_3 r_{3j}(t) D(d)$$

   where d is the Euclidean distance between prey particle i and the predator. $D(d)$ quantifies the influence that the predator has on the prey.

   Explain how the algorithm can be adapted to use multiple predators in the algorithm instead of one. Discuss the possible effects this could have on the algorithm.

6. (3 marks) Can a PSO be regarded as a form of Cooperative Co-Evolution? Motivate your answer clearly.

7. (3 marks) Briefly explain why an atomic swarm tends to perform better than a charged and a neutral swarm.

8. (10 marks) Consider some function, $D(x_0, x_1, x_2, x_3)$ that returns some real value. Now, given a starting point (0, 0), an initial set of 4 parameters x_0, x_1, x_2, x_3 and an initial direction (say upwards), an agent is moving around in a spiral-like pattern using the following simple algorithm:

   1. Determine the distance, d, according to the function, $D(x_0, x_1, x_2, x_3)$.
   2. Draw a line from the current position, in the direction the agent is pointing, d units long.

3. Discard x_3 and let $x_n = x_{n-1}$ for $n > 0$. For example x_3 will get the value of x_2 and so on. Finally, let $x_0 = d$, i.e. the last value returned in step 1.
4. Turn right.
5. Back to step 1.

The task is now to find the initial set x_0, x_1, x_2, x_3, where $x_n \in \mathbb{R}$ that will result in the agent walking the longest path, where d always remain within a given range $d_{min} \leq d \leq d_{max}$. The moment the agent strays out of the range, the drawing process stops.

Show how this can be solved with a PSO algorithm.

Section C: Ant Algorithms

1. (3 marks) In a bridge experiment with two paths of **equal** lengths from the source to destination, ants will eventually converge to one of the paths. Explain why.

2. (2 marks) Briefly discuss the importance of the forgetting/evaporation factor in the pheromone trail depositing process.

3. (10 marks) The travelling salesman problem (TSP) asks the following question:

   > "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

   Using the Ant System (AS), show how the TSP can be solved. Focus on the specifics of the TSP (like the heuristics value), rather on reiterating the specifics of the AS algorithm.

4. (15 marks) You are provided by a number of $n \times m$ black and white bitmaps representing alphabetic characters of various fonts. If a bitmap position is '1', that position in the bitmap is filled (black), while '0 indicates an unfilled (white) position.

   Briefly explain how the Lumer-Faieta clustering algorithm can be used to cluster all the symbols of the same type together (i.e. all the As, all the Bs, etc.). Place special emphasis on this specific problem, rather than elaborating on the theoretical aspects of the theoretical algorithm.