**Introduction**

**Swarm intelligence (SI)** is the collective behaviour of decentralized , self organized systems, natural or artificial. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems.

SI systems consist typically of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Natural examples of SI include ant colonies. Bird flocking ,animal hearding , bacterial growth , and fish schooling.The definition of swarm intelligence is still not quite clear. In principle, it should be a multi-agent system that has self-organized behaviour that shows some intelligent behaviour.

In its common usage, “swarm” refers to a disorganized cluster of moving things, usually in- sects, moving irregularly, chaotically, somehow staying together even while all of them move in apparently random directions.

**Particle Swarm Optimaization (PSO)**

Particle swarm optimization, which has roots in artiﬁcial life and so- cial psychology as well as engineering and computer science, differs from evolutionary computation methods in that the population members, called particles, are ﬂown through the problem hyperspace. When the population is initialized, in addition to the variables being given random values, they are stochastically assigned velocities. Each iteration, each particle’s velocity is stochastically accelerated toward its previous best position (where it had its highest ﬁtness value) and toward a neighbor- hood best position (the position of highest ﬁtness by any particle in its neighborhood).

**Ant Colony Optimaization (ACO)**

In computer science and [operations research](http://en.wikipedia.org/wiki/Operations_research), the **ant colony optimization** [algorithm](http://en.wikipedia.org/wiki/Algorithm)**(ACO)** is a p[robabilistic](http://en.wikipedia.org/wiki/Probability)technique for solving computational problems which can be reduced to finding good paths through [graphs](http://en.wikipedia.org/wiki/Graph_(mathematics)).

This algorithm is a member of the **ant colony algorithms** family, in [swarm intelligence](http://en.wikipedia.org/wiki/Swarm_intelligence)methods, and it constitutes some metaheuristics optimizations. Initially proposed by Marco Dorgio in 1992 in his PhD thesis,the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.

**ACO Algorithm:**

Artiﬁcial ants, unlike their biological counterparts, move through a discrete environment deﬁned with nodes, and they have memory. When traversing from one node to another, ants leave pheromone trails on the edges connecting the nodes. The pheromone trails attract other ants that lay more pheromone, which consequently leads to pheromone trail accumulation. Negative feedback is applied through pheromone evaporation that, importantly, restrains the ants from taking the same route and allows continuous search for better solutions.

Ant System (AS) is the ﬁrst ACO algorithm proposed in literature and it was initially applied to the Travelling Salesman Problem (TSP) [5]. A general deﬁnition of the TSP is the following. For a given set of cities with known distances between them, the goal is to ﬁnd the shortest tour that allows each city to be visited once and only once. In more formal terms, the goal is to ﬁnd the Hamiltonian tour of minimal length on a fully connected graph.

AS consists of a colony of artiﬁcial ants that move between the nodes (cities) in search for the minimal route. The probability of displacing the kth ant from node i to node j is given by:

where τij and ηij are the intensity of the pheromone trail on edge (i, j) and the visibility of the node j from node i, respectively, and α and β are control parameters (α, β > 0; α, β ∈ <). The tabuk list contains nodes that have already been visited by the kth ant. The deﬁnition of the node’s visibility is application-related, and for the TSP it is set to be inversely proportional to the node’s Euclidean distance:

AS is performed in iterations. At the end of each iteration, pheromone values are updated by all the ants that have built a solution in the iteration itself. The pheromone update rule is described with the following equation:



where ρ is the pheromone evaporation rate (0 < ρ < 1, ρ ∈ ℜ), m is the number of ants in the colony, and ∆τk ij is the amount of pheromone laid on the edge (i, j) by the kth ant, and is given by:



where Lk is the length of the tour found by the kth ant, and Q is a scaling constant (Q > 0, Q ∈ ℜ).

The algorithm stops when the satisfactory solution is found or when the maximum number of iterations is reached.

Dorigo, M., Maniezzo, V. & Colorni, A. [1996]. Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics - Part B 26(1): 29–41.

**K-Means Clustering Algorithm**

This algorithm involves the technique of region-growing by dividing all the pixels of the image among clusters.The algorithm is very simple as given by the following wikipedia entry:

The K-means algorithm is an iterative technique that is used to partition an image into K clusters. The basic algorithm is:

Pick K cluster centers, either randomly or based on some heuristic

Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center

Re-compute the cluster centers by averaging all of the pixels in the cluster

Repeat steps 2 and 3 until convergence is attained (i.e. no pixels change clusters)

However,to manage the trade-off between time and efficiency of results we have run the algorithm for only a fixed number of iterations rather than waiting indefinitely for convergence.But still our number of iterations are such that the results are very efficient(and fast also).

**Disadvantages of K-Means Clustering**

Though it simple to implement and algorithm produces good results,it is heavily dependent on the initial choice of cluster points(i.e the image pixels which correspond to the intital k-clusters).So,this makes the process slightly unreliable...to make it better we approach swarm intellkigence techniques for help and come up with a unique algorithm-Bacterial Foraging Algorithm.

**Bacterial Foraging Optimization (BFO)**

Bacterial foraging optimization algorithm (BFO) has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control. BFOA is inspired by the social foraging behavior of Escherichia coli. BFOA has already drawn the attention of researchers because of its efficiency inolving real-world optimization problems arising in several application domains. The underlying biology behind the foraging strategy of E.coli is emulated in an extraordinary manner and used as a simple optimization algorithm.

**BFO Algorithm**

In BFO, the motile bacteria such as Escherichia coli propel themselves by rotating their flagella. They rotate their flagella counter clockwise to move forward rotate also called as “swimming” (or “runs”). But to move the bacteria in random direction i.e. “tumble” they rotate their flagella clockwise and then swims again. Tumbling just changes the direction of movement of bacteria. The bacteria first of all tumble in random direction to search for food. As the bacteria found the food in a particular direction it then swims toward the food in that direction. An alternation between “swim” and “tumble” enablesthe bacteria to search for food in random directions [2].

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination- dispersal. These are described as follows [2]

* Chemotaxis In the original BFO, a unit walk of the bacteria with random direction represents a “tumble” and a unit walk with the same direction in the last step indicates a “run.”
* Reproduction The fitness value of each bacterium is calculated as the sum of the step fitness during its life and all bacteria are sorted in descending order according to health status. In the reproduction step, only the first half of population survives. The surviving population is divided into two identical ones, which are then placed in the same locations at which their parents were. Thus, the total population of bacteria remains constant [2].
* Elimination and Dispersal The chemotaxis provides a basis for searching the local best solution, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. The bacteria with the best positions are kept and the remaining bacteria population is killed. The bacteria with best positions are then moved to another position within the environment [2].

[2] Dekker A (1994) Kohonen neural networks for optimal colour quantization,

Network: Computation in Neural Systems 5: 351- 367

**Application in image segmentation**

Firstly we divide all the pixels among k-clusters using the k-means clustering algorithm(we choose a value of 590 for k so that there are a lot of options for bacteria to find the optimal fitness value).Now,we apply the bacterial foraging algorithm.The output is an image wherein each pixel belongs only to the 'f'ittest' clusters and less fittter clusters contain lesser number of pixels or are eliminiated.For a cluster to be fit enough there should be minimum color distance between the pixels of the same cluster and maximum color gap with the mean color value of the other clusters.After that ,since we are interseted in breaking our image down to 3 images,we again apply k-means(by taking k=3).

**Pseudocode:**

void BF(){

for(int i = 0; i < iter ; i++){

for(int j = 0 ; j < img.width ; j++)

for(int k = 0 ; k < img.height ; k++)

move(j , k); //if you get the food then 'swim'

//else 'tumble'

reproduce();

//and also eliminate the less 'fit' clusters

}

}

**Results of bacterial-foraging based image segmentation**

Here we have compiled the snapshots of the results obtained from our optimized clustering algorithm(which eas described above).Basically,in the end we have completely partitioned our image into 3 separate images.wherein each image corresponds to the most prominent areeas/regions visible within the image.By prominent,we mean what would a child answer of you ask him "Hey,what can you see in the image?".He would most likely say(for image 1)"mountains,sky and ground",so each image correspond to one of these three prominent components.

|  |  |
| --- | --- |
|  | Time in ms : 852 |
|  | Time in ms : 1075 |
|  | Time in ms: 1369 |
|  | Time in ms: 1675 |
|  | Time in ms: 1436 |

Summary:

Average time for an image of size 256x256 comes to be 1433 ms for an initial cluster space of 590 clusters whereas for a cluster space of just 10 which gives decent output is just 806.5 ms.

**Image Processing**

**Edge detection** is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image birghtness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed *edges*. The same problem of finding discontinuities in 1D signals is known as step deduction and the problem of finding signal discontinuities over time is known as. Edge detection is a fundamental tool in image processing ,machine vision computer vision, particularly in the areas of feature detection and extraction.

Edge detection is a pre-processing step in applications of computer and robot vision. It transforms the input image to a binary image that indicates either the presence or the absence of an edge. Therefore, the edge detectors represent a special group of search algorithms with the objective of ﬁnding the pixels belonging to true edges. The search is performed following certain criteria, as the edge pixels are found in regions of an image where the distinct intensity changes or discontinuities occur (e.g. in color, gray-intensity level, texture,etc.).

**Use of ACO**

Ants use pheromone trails to mark the path to the food source. In digital images, pixels deﬁne the discrete space in which the artiﬁcial ants move and the edge pixels represent the food. The edge detection operation is performed on a set of grayscale images.

The AS algorithm is an iterative process which includes the following steps:

* Initialization:

The number of ants proportional to √N · M is randomly distributed on the pixels in the image. Only one ant is allowed to reside on a pixel within the same iteration. Initial non-zero pheromone trail value, τ0, is assigned to each pixel, otherwise the ants would never start the search.



where Imax is the maximum gray-intensity value in the image (0 ≤ Imax ≤ 255). For the

pixels in regions of distinct gray-intensity changes the higher visibility values are obtained,

which makes those pixels more attractive to ants.

* Pixel transition rule:

Unlike their biological counterparts, artiﬁcial ants have memory.Tabuk represents the list of pixels that the kth ant has already visited. If ant is found surrounded by the pixels that are either in the tabu list or occupied by other ants, it is randomly displaced to another unoccupied pixel that is not in the tabu list. Otherwise,the displacement probability of the kth ant to a neighboring pixel (i, j) is given by:



where τij and ηij are the intensity of the pheromone trail and the visibility of the pixel at

(i, j), respectively, and α and β are control parameters (α, β > 0; α, β ∈ ℜ).

3. Pheromone update rule:

Negative feedback is implemented through pheromone evaporation according to:

 and



T is a threshold value which prevents ants from staying on the background pixels hence enforcing the search for the true edges. The existence of the pheromone evaporation rate,ρ, prevents the algorithm stagnation. Pheromone trail evaporates exponentially from the repeatedly not-visited pixels.

* Stopping criterion:

The steps 2 and 3 are repeated in a loop and algorithm stops executing when the maximum number of iterations is reached.

**Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Of Iterations  No. Of Ants Multiplier  Time in ms | 5  5  218 | 7  5  299 | 5  7  474 | 10  10  2157 |
| No. Of Iterations  No. of Ants Multiplier  Time in ms | 5  5  226 | 7  5  293 | 5  7  444 | 10  10  2384 |
| No. Of Iterations  No. of Ants Multiplier  Time in ms | 5  5  215 | 7  5  304 | 5  7  512 | 10  10  1888 |
| No. Of Iterations  No. of Ants Multiplier  Time in ms | 5  5  230 | 7  5  322 | 5  7  475 | 10  10  2116 |

Summary:

For a observation sample set of 12 Images run with different number of Ants and for different number of iterations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Of Iterations | 5 | 7 | 5 | 10 |
| No. of Ants Multiplier | 5 | 5 | 7 | 10 |
| Time in ms | 218.75 | 297.5 | 453.41 | 2037.75 |

**Conclusions**

Swarm Intelligence-based techniques can be used in a number of applications. The U.S. military is investigating swarm techniques for controlling unmanned vehicles. The European Space Agency is thinking about an orbital swarm for self-assembly and interferometry.NASA is investigating the use of swarm technology for planetary mapping. A 1992 paper by Anthony Lewis and George A. Bekey discusses the possibility of using swarm intelligence to control nanobots within the body for the purpose of killing cancer tumors.Conversely al-Rifaie and Aber have used Stochastic Diffusion Research to help locate tumours.Swarm intelligence has also been applied for data mining.

In this project we successfully applied the concepts os Ant colony optimization and Bacterial Foraging to Image processing using them for two vital Image Processing techniques namely Edge Detection and Image Segmentation