Interleaving between Ant Colony Optimization and Tabu Search for Image Matching

Ghusoon S. Basheer
College of Computer sciences and Mathematics
University of Mosul, Iraq

Received on: 01/08/2006 **Accepted on: 23/01/2007**

ABSTRACT

Image matching plays an important role in many applications such as multi-modality medical imaging and multi-spectral image analysis. The role of matching is to integrate multiple sources of object information into a single image. The matching problem consists of determining the unknown transform parameters required to map one image to match the other image(20). Different non – traditional methods are used for solving this kind of problem. Among these methods are the Genetic Algorithms, Neural Networks & Simulating Annealing.

Swarm Intelligence (SI) algorithms take their inspiration from the collective behavior of natural, for example, ant colonies, flocks of birds, or fish shoals, a particularly successful strandant colony optimization (ACO)(1). **Ant Colony Optimization** is a population-based general search technique, proposed by Dorigo(1992,1996), for the solution of difficult combinatorial problems(4). The studies show that, in nature, the ant colony is able to discover the shortest paths between the nest and food sources very efficiently, such a deposit substance is called ***pheromone*** during talking and another ants can smell it, if one of ants find a short path, it feedback on the same path and the value of pheromone on this path increases and a another ants gradually chose this path.(22)

Tabu search is one of the best known heuristic to choose the next neighbor to move on. At each step, one chooses the best neighbor with respect to specific function (23).

The basic idea in this paper is using Ant Colony Optimization(ACO) & Tabu Search(TS) as a success strategy for matching two images. The suggestion algorithm evaluation is a good promising solution, by providing an optimal algorithm which is executed by optimal time and coast, I believe that there is no prior research conjoining the two topics in this way. The program is written in Matlab language (6.5).

Keywords: Ant Colony Optimization, Tabu search, image matching.

دمج بين مستعمرة النمل المثلى والبحث المجدول لمطابقة الصور

غصون بشير
كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ الاستلام: 2006/08/01 **تاريخ القبول: 2007/01/23**

الملخص

تؤدي مطابقة الصور دورا فعالا في عدد من التطبيقات المهمة مثل الصور الطبية ذات الأشكال المتعددة وفي تحليل صور المنظار الطبي. وتتركز مسألة المطابقة في تحديد متغيرات الحركة غير المعروفة بحيث تتم مطابقة إحدى الصور المجهولة مع صورة محددة معلومة. استخدمت العديد من الطرائق غير التقليدية لمعالجة مسألة المطابقة وأكثرها شيوعا الخوارزمية الجينية (Genetic Algorithm)، الشبكات العصبية (Neural Networks) والمحاكاة الكاذبة (Simulating Annealing)(20).

استوحت خوارزميات ذكاء الأسراب(Swarm Intelligence) فكرتها من سلوك المجاميع في الطبيعة كمستعمرات النمل، أسراب الطيور وصفوف الأسماك، والخوارزمية الأكثر نجاحا هي الخوارزمية المحاكية لسلوك مستعمرات النمل, خوارزمية النمل المثلى(Ant Colony Optimization)(1)، اكتشفها(1992-1996)Dorigo, استخدمت في حل مسائل الأمثلية المعقدة. بينت الدراسات إن مستعمرات النمل قادرة على إيجاد الطريق الأقصر مابين الوكر والغذاء بكفاءة عالية وسبب ذلك وضع النمل مادة على الطرق التي تسير فيها بحيث تستطيع بقية النملات شمها, فعندما تصل واحدة من النملات إلى الغذاء سالكة طريق اقصر من الطرق الأخرى فإنها تعود من نفس الطريق وبإفرازها لمادة الفيرومون ذهابا و إيابا ترتفع نسبته على الطريق الأقصر, بعد فترة تسلك كل المستعمرة نفس الطريق.(21).

تعد خوارزمية بحث الـTabu من الخوارزميات المعروفة التي تختار أفضل حل مجاور للحل الحالي, في كل خطوة يتم اختيار أفضل حل من بين عدة حلول(23).

تستخدم الفكرة المقدمة في هذا البحث خوارزمية النمل المثلى(ACO) والبحث المجدول (TS) بوصفه إستراتيجية بحث لمطابقة الصور، تم الحصول على حلول جيدة وبفضاء بحثي صغير وبزمن قياسي. يجب التأكيد على أن فكرة الدمج بين الخوارزميتين واستخدامها في مطابقة الصور لم يقدم في بحث سابق .تمت كتابة البرنامج بلغة(Matlab 6.5).

الكلمات المفتاحية: خوارزمية النمل المثلى، خوارزمية البحث الممنوع، مطابقة الصور.

1. Introduction:

It is necessary to distinguish between two different classes of problems: **P** (or **polynomial**) problems whose complexity is determined by a polynomial function, and **NP** (**non-polynomial**) problems which are not determined by polynomials. In other words, the number of steps that are necessary to solve an instance of the NP-Problems increases exponentially) with the size of the instance itself. Consequently, it is not possible to solve such problems exactly in an acceptable time with the algorithms and computing power which are available nowadays(21). Search-based image matching problem methods utilize an iterative procedure to improve the initial guess of the unknown transform parameter, this can be done by searching in very large search space, so this problem is NP-Complete. Intelligently methods are especially appropriate for the optimization in large search space, which are unsuitable for exhaustive search procedure, it trades-off between the exploration of the search space and the exploration of the best solutions found so far(20). The basic idea is using Ant Colony System (ACS)(one of intelligently methods) as a population based search strategy to evolve promising starting solutions (affine transformation).

2.Related Work:

Ant Colony Optimization used by many researches for finding the optimal solutions in NP-Complete problems .Morillo,Fernandez& Orduna (17) proposes a new implementation of the Ant Colony Optimization(ACS) search method for solving the partitioning problem iv DVE systems, they have evaluated the proposed implementation for both small and large DVE systems and compared the results with the ones provided by the Linear Optimization Technique(LOT), the partitioning method that provides the best solutions for DVE systems, ACS methods able to find solutions better than the ones provided by LOT method.

Cord´on(5) proposed new hybrid algorithm to solve real world application, such as Fuzzy rule learning and several Bioinformatics problem by analyzing the possibility of hybridization of different metaheuristic, namely Evolutionary computing and Ant Colony Optimization. In (21), Socha, Sampels & Manfrin presents two Ant Algorithms to solving a simplified version of a typical University Course Timetabling problem: Ant Colony System and MAX-MIN Ant System. The main difference between them is in the way of pheromone updating. The MAX-MIN ACO System (MMAS) performs better than ACS on all instances tested. Five metaheuristics algorithm (Genetic Algorithm, Simulating Annealing, Tabu Search, Iterated Local Search and Ant Colony Optimization) were evaluated and then compared with two Ant Algorithm. It becomes clear that for small class problem the Ant Algorithm proves to be very efficient, while in medium instance of problem, the Simulating Annealing better than MMAS, in the large instance MMAS better than any Algorithm.

Dorigo& Gambardella (10) in their paper introduce Ant Colony System (ACS) and applied it to the traveling salesman problem (TSP). A set of cooperating agents ***ANTS*** cooperates to find good solutions to TSPs. Ants cooperate using an indirect form of communication mediated by pheromone they deposit on the edges of the TSP graph while building solutions. They study ACS by running experiments. The results show that

ACS outperforms other nature-inspired algorithms such as simulated annealing and evolutionary computation. In (7), the researchers explore the addition of a look-ahead mechanism to ACO algorithm. In constructing a candidate solution for scheduling problem, ANTs use a transition rule that incorporates complete information on past decisions **(*the trail*)** and local information (***visibility*)** on the immediate decision that is to be made. They will present a series of numerical experiments which allow us to distinguish the contributions of each of the elements of the augmented ACO for industrial scheduling. Christodoulou, S.(4) in his paper presents a methodology to arrive at optimal truss designs using Ant Colony Optimization (ACO) algorithms. The paper outlines the suggestion of possible implementation strategy for solving optimal truss designs (geometrical configuration and member characteristics). The ACO seems to be functioning well in searching for optimal truss topology solutions, it was able to improve the initial topology, in terms of total truss weight and resulted in a modified topology of reduced weight prior to reaching an indeterminate structure.

3. Ant Colony Optimization:

Ants first evolved around 120 million years ago, take form in over 11,400 different species and are considered one of the most successful insects due to their highly organized colonies, sometimes consisting of millions of ants. Computer scientists began researching the behaviors of ants in the early 1990’s to discover new routing algorithms. The result of these studies is ***Ant Colony Optimization*** (***ACO***) and in the case of well implemented ACO techniques, optimal performance is comparative to existing top-performing routing algorithms(2). In (ACO) meta-heuristic a colony of artificial ants cooperate in finding good solutions to difficult discrete optimization problems. Cooperation is a key design component of ACO algorithms: The choice is to allocate the computational resources to a set of relatively simple agents(artificial ants) that communicate indirectly by stigmergy. Good solutions are an emergent property of the agents’ cooperative interaction.(6)

1. Advantage of Ant Colony Optimization :

The important aspect of real ants’ foraging behavior that is exploited by artificial ants is the coupling between the **autocatalytic(positive feedback)** mechanism and the **implicit evaluation** of solutions. By implicit solution evaluation we mean the fact that shorter paths (which correspond to lower cost solutions in artificial ants) will be completed earlier than longer ones, and therefore they will receive pheromone reinforcement more quickly. Implicit solution evaluation coupled with autocatalysis can be very effective: the shorter the path, the sooner the pheromone is deposited by the ants, the more the ants use the shorter path. If appropriately used, autocatalysis can be a powerful mechanism in population-based optimization algorithms (6).

2.Disadvantage of Ant Colony Optimization:

a. Although Ant Colony Optimization(ACO) is an interesting and promising result, it remains clear that as well as other metaheuristics, in many cases cannot compete with specialized local search methods. A current trend is therefore to associate with the metaheuristic a local optimizer, giving birth to so-called hybrid methods(13).

b. Ants build solutions applying a probabilistic decision policy to move through adjacent states. As for real ants, the artificial ants' policy makes use of local information only and it does not make use of lookahead to predict future states. Therefore, the applied policy is completely local, in space and time.

4.Using ACO for Optimization Problems:

There are two kinds of optimization problem: **Static problems** & **dynamic problems**.

1. ***Static problems*** are those in which the characteristics of the problem are given once and for all when the problem is defined, and do not change while the problem is being solved. A paradigmatic example of such problems is the classic traveling salesman problem , in which city locations and their relative distances are part of the problem definition and do not change at run-time.

2. ***dynamic problems*** are defined as a function of some quantities whose value is set by the dynamics of an underlying system. The problem changes therefore at run-time and the optimization algorithm must be capable of adapting online to the changing environment. The paradigmatic example discussed in the following of this section is network routing(6). The solution construction is biased by the pheromone trails which change at run-time, the heuristic information on the problem instance and the ants' private memory (4).

4.1 Pheromone Evaluation :

Dorigo M &. Di Caro G.(6), proves that an experiment typically lasts approximately one hour, it is plausible to assume that the amount of pheromone evaporated in this time period is negligible .

In this paper we make the assumption that the amount of pheromone on a branch is proportional to the number of ants that used the branch in the past. This assumption implies that pheromone evaporation is not taken into account. There are two types of evaluation of pheromone:

First standard and called **Local Pheromone Values**, Such each ante updates the value of pheromone locally during pass on that path or edge ,and every ante or node has probability calculate as follows:

Probability of node **:Local Evaluation: Pij = [Tij] [ŋij] / ∑[Tih][ŋih]**

Ŋij: Heuristic value.

Pheromone value : Tij**.**

Second called **Summation Evaluation**, in this type updating of pheromone value depends on the later dissection without neglecting the value of local pheromone value , and calculate as follows:

Weighted Summation Evaluation:Pij=∑[γ.Tkj].[ŋij] / ∑(∑[γ.T])[ŋih]

if **γ** > 0 this determines the percentage effect for the later pheromone value. If **γ** =1, there is no effect and this is similar to the local pheromone value.(16)

5.Tabu Search:

Tabu search is a mathematical optimization method, belonging to the class of local search techniques. Tabu search enhances the performance of a local search method by using memory structures, is designed to be useful and accessible to researchers and practitioners in management science, industrial engineering, economics, and computer science**(**8**).**

Tabu search (TS) goes beyond the classical design to provide a method that is dramatically changing our ability to solve problems of practical significance(12).

Until some stopping criterion has been satisfied(8):

1.Use a local search procedure to iteratively move from a solution x to solution x' in the neighborhood of x.

2.modify the neighborhood structure of each solution as the search progresses to explore regions of the search space that would be left unexplored by the local search procedure .

3.The new neighborhood determined .
Do

1. Advantage of TABU :

1.cycle avoidance which also saves time.

2.in ducking vigor in search(diversification exploration): guide search to more promising or new regions of search space(18).

2. Disadvantage of TABU:

1.Current iteration is prevented from undoing the last move or current iteration is prevented from repeating last previous solutions.
2.computationally more expensive(18).

6.Image Matching:

The most important factors in multi point squares image matching which play an important rule in the accuracy, reliability and efficiency of matching results are: consideration of the type of constraint, number of matching points and the decided weight for constraints(19). Image matching leads to finds corresponding points in two images to estimate the difference between them, and the question in this state: what is the criteria or conditions which must be satisfied to decide if two images are matching or not(3). A range image is a picture in which each pixel value encodes not the intensity of light reflected in a certain direction but rather the distance (or range) of the nearest surface in that direction. This type of imagery, therefore, provides direct, explicit geometric information which is useful in many applications(18).

In this paper, we define border of two images as coordinate on the surface (x & y axis) and use Euclidean law***** for calculating the distance between all points in two images and determined the rang of matching between them .

***Euclidean law:** $D = \sqrt{(\Sigma \left| d_1(i,j) - d_2(i,j) \right|^2)}$

d1,d2:distance transformation of the respective image

7.Suggestion Algorithm :

The idea is dynamically applying a heuristic search method that provides a good result for matching the two images. Since the matching algorithm must achieve multiple iterations of comparisons between the two images, the search method must be as faster as possible, which means a minimum number of iterations (minimum search space).

Therefore, we proposed the use of Ant Colony Optimization (ACO) and Tabu Search (TS) an evolutionary computation algorithms. The minimum number of comparisons between borders of two images is the shortest path which ACO and TS satisfied it. We assume the edge for two images is determined and this paper achieved matching between them. The basic idea for matching is making one of two images stable and modifying the second image by angles of rotation and translation until matching satisfied or numbers of determined iterations end without matching .

Assumption there are ten ants which are in different spaces and these ants pass through several intermediate nodes to reach the food (matching) and as follows:

1.Initially,we generate array (3*10) of random integer values(each row consists of three values: the first value represents the values for rotating θ, the two second values represent the translating values (on the x and y axis)).

2.We generate ten antes by changing the coordinates of the first image (modified image). (each node uses each row in this array for generating ten new intermediate nodes).

3.Each node(ant) from there ten generates ten intermediate nodes(new modified images) using initial array and passes through search path.

4.We divided the search path into stages, each stage represents two pass through Tabu search. Between each two stages there is ACO search.Table (1) displays the differences between ACO and TABU search.

5.In each pass through Tabu and ACO search there are multiple modified images, by using **Euclidean law** we calculate multiple differences between two images (original(stable) image and modified images),and depending on the values of comparisons we decided the values of heuristic functions and then the pheromone values for each node, as shown in figure(1).

Stage I → ***ACO Search*** → **stage i+1** → ***ACO Search*** → **….** **stage i+n-1**
tabu Search ***tabu Search*** ***tabu Search***

Figure (1) The stages of search

We note in figure (1) ,search starts with Tabu search (stage I)(each stage consists of two pass),then followed by ACO search ,the output nodes from ACO consider the input nodes to Tabu ,this continues until reaching the n (n: maximum number of iterations).

Suggestion Algorithm:
-Create initial array of (3*10)integer values.
-Using the initial array for generating ten nodes by modifying one of two images by the values in this array.
-While not matching or not reaching the maximum numbers of iterations:
-For each node in search space:

T
A
B
U

***Generate new ten nodes by modifying current node by initial array(100 intermediate nodes).**
***For each ten nodes :select the best neighbor to be the next nodes .**

***Update the heuristic and pheromone values for the chosen node.**
-From this best ten nodes(result of tabu search) generate 100 intermediate nodes using initial array.
-Check the pheromone values for all nodes in search space
A
C
O
-If pheromone value for one node twice pheromone value for other:
-Remove the node which have minimum value from search space
-Update pheromone values for other nodes
End while

ACO search	TABU search
1.The search in local space(in each 10 nodes). 2.walking of each 10 nodes independent. 3.choosing the best node depends on the heuristic value for each node. 4.no removable nodes. 5.updates the heuristic values.	1.The search in global space(in all nodes in the problem). 2.walking of nodes is related to each other. 3.choosing the best node depends on the pheromone value for each node. 4.after several iterations, there is removable nodes. 5.updates the pheromone and heuristic values

Table(1) : display the differences between ACO and TABU search

7.1 Determining the Number of Nodes :

The **number of total nodes in this problem** was undetermined and changes during the run time, also we note that multiple of nodes will be removed during ACO search, because these two reasons we consider this problem as **Dynamic Combinational Optimization Problem .**

We assume the number of initial nodes(ten) and consider it a starting point for search (each node represents ant walk from nest to food).

7.2 Evaluation of Heuristic Values:

Heuristic value is different from one problem to another. Here we want to minimize the search space and reducing the number of iterations for comparisons. For this goal, we calculate the heuristic values depending on the convergence of each new image(modified image) to the original image(stable image) by evaluating the differences between all points in two images and calculating the heuristic values as follows(Heuristic value and differences inversely proportional):
Heuristic value = 1/difference between all points in the two images .

7.3 Pheromone Evaluation:

The pheromone values for each node in search space depend on the heuristic value for this node. Therefore we use Heuristic value for each iteration as a pheromone value divided by 1000 to easier calculation , as follows:

Pheromone value = pheromone value + heuristic value /1000.

8. Practical Representation:

1. Generating initial values for rotation and translation:

In the beginning, we generate array with dimension(3*10), Each row represented the θ for rotation and translation value on x & y axis. Each node uses all rows in this array. That means each node generates ten nodes .

a. Rotation:

Before rotation image2 (b2 array), we translate the value of θ into ***RADIUS ANGLES*** by multiplying it by (2 * π), then use it for rotating the all pixel of image 2 (b2) as follows:

X1 = X * COS Θ _ Y * SIN Θ

*** COS Θ Y1 = X * SIN Θ + Y**

X1: x axis for new point.

X : x axis for original point.

Y1: y axis for new point.

Y : y axis for original point.

SINΘ :sin theta for rotation.

COSΘ: sin theta for rotation.

We build **ROT** function which calculates rotation θ. In this function, we rotate each pixel in image 2) by the value r (r: value of θ),

b. Translation:

The coordinate which is produced from ROT function is rotation coordinate. Now this coordinate adding with values of TEMP(value for translate x & y axis) for getting the final values for rotation and translation coordinates.

FINALB2 (X) = NEW X 1 + TEMP1

FINALB2 (Y) = NEW Y 1 + TEMP2

FINALB2(X): x axis for final array after rotation and translation.

. FINALB2(Y): y axis for final array after rotation and translation

NEWX1: x axis from rotation.

NEWY1: y axis from rotation.

TEMP1: result from initial value of x axis.

TEMP2: result from initial value of y axis.

The representation of the new boundary for new image after rotation and translation by using values from array FINASLB2 (new node).

2 . Reading the Image :

We assume that the borders of two images(binary image) which must match are known.

3 . Determining the initial nodes(antes):

We consider that we have ten nodes for satisfying the matching (ten antes walking from nest to food). These ants are generated by modifying one of two images by using the values in initial array (using values for rotating and translating the coordinates of the image).

4. Determining the intermediate nodes:

In each stage of search we generate the intermediate nodes by modifying the initial ten nodes by using the same values in initial random array, this means that each node will generate ten other new nodes in each stage, in Tabu search, the current node choose the best neighbor node from the new ten nodes to be the next node. When moving to the next node we update the pheromone value for this node (adding constant number to the counter of pheromone value for this node) .

5. Representing the image as values in array:

The borders of two images is integer values, we translate this value into two arrays (two dimensions arrays) which represent the x and y axis for each pixel in two images called b1(boundary for original image)and b2(boundary of modified image).

Note: We rotate and translate b2 array and compare it with b1.

c. Building the (Evaluation Function) EVF :

By using the **Eqledian law***, we calculate the square of differences between the distance for all pixels of first image (b1 array) and second image(b2 array after rotation and translation (FINALB2 array)) as follows:

(FINALB2 (I , X) – B1 (I , X)) ^2 DIF(I , X) =

FINALB2 (I , Y) – B1 (I , Y))^2 (DIF(I , Y) =

DIF(I,X): different for x axis in index I.

DIF(I,Y): different for y axis in index I.

y axis for array B1in index I. **B1(I,Y):**

FINALB2 (I,X):x axis for array FINALB2 in index I

FINALB2 (I,Y):y axis for array FINALB2 in index I

B1(I,X) :x axis for array B1in index I

***Euclidean law:** $D = \sqrt{(\sum \left| d_1(i,j) - d_2(i,j) \right|^2)}$

d1,d2:distance transformation of the respective image

After this, we find the square root for summation of the difference in two axis as follows:

$DIF(I) = \sqrt{(DIF(I,X) + DIF(I,Y))}$

6. Heuristic and pheromone values:

After calculating the differences between b1 and 100 finalb2(b2) array(each node generates five node). Now, we must calculate the heuristic values for each node. This is achieved depending on the idea that heuristic value inversely proportional with summation of all differences between two images(all pixels of two images), as follows:

H =1 / DIF

H : heuristic value

DIF :value for differences for that node.

When one of the 100 nodes is chosen, then the pheromone value must be increased in these nodes , the pheromone value is increased using the following formula:

PH=PH + H /1000

We calculate ph as a rate for heuristic value (for easier calculation).Where Heuristic value for this node in that iteration.

7. Representing the Tabu Search:

After calculating the heuristic value for each nodes, we used Tabu search algorithm for choosing the best neighbor node to be the next node, depending on the Heuristic value, (node with higher heuristic value was choosen).

8. Representing the ACO Search:

After two steps in Tabu Search, ACO search will begin, The results from Tabu search are only five nodes. We build SERCH function to locate the nodes with higher pheromone value from this node and compare the pheromone values of each one with other nodes. In case which the pheromone value of one node is twice the value of pheromone value for another node ,we remove the node with lowest pheromone value and update the pheromone value for the other nodes.

9. Results:

As shown in table(2), we generate ten random integer values for rotation and translation of each node(image) in search space, the first field of table represents the values for translating the coordinates of each point in the borders of the image, the second field represents the angles for rotation .

In order to control the number of nodes in each iteration, we generate ten initial ants(nodes), this ten nodes are generated by translating and rotating the borders one of two images by the values in table(2) for obtaining new ten images with new borders, we calculate evaluation function for each new node by **Euclidean law** and determining the heuristic and pheromone value for each one as shown in table(3).

In each iteration, each node in search space, generates 10 intermediate nodes using the same values in table(2). The search starts with two stages of Tabu search which don't remove any node from search space, but choosing the best neighbor(depending on the heuristic and pheromone values) from others to be the next node.

As shown in table(3), in the first stage of Tabu search, from 100 intermediate nodes the chosen nodes are (**1,12,25,32,41,53,44,55,61,73**), by using table(2), these ten nodes generate another 100 intermediate nodes and these nodes pass through a second stage of Tabu search to select the best node which is(**2,15,21,32,44,52,65,74,85,95**).

After each two paths of Tabu search, there is ACO search which checks the pheromone values for each node in a search space and determines **not promising nodes** and removes it(decreases the size of search space). We can note from the table that the removing of nodes in ACO search depends on the different between maximum and minimum values of pheromone.

In general, decreasing the number of nodes in search space leads to speed up the matching. In the field **pheromone value** the percentage increases in pheromone values at each iteration.

In first iteration of ACO, the nodes:(74,81,94) have minimum value of pheromone ,therefore this node will be removed from search space (not promising nodes), then the input node to the next Tabu search is 7 nodes only and the intermediate nodes is 70.By Tabu search, from this 70 nodes we determine best 7 nodes **(3,15,21,32,45,53,64**), by using table(2),these ten nodes generate another 70 intermediate nodes and these nodes pass through a second stage of Tabu search to select the best 7 nodes which are **(1,13,25,34,4455,65**).

In the second iteration of ACO ,the removable nodes are(**24, 33, 44, 54, 63**),then the intermediate node in next stage is 20 nodes only.

The role of ACO search in decreasing the size of search space very clear and important in speed up the convergence to the solution of the problem.

Figure (2)and (3) show the results from the converge of the modified image to original image after few iterations .

10.Conclusion and Future Work:

In this paper , we have proposed a new implementation of the Ant Colony Optimization system search method with TABU search for solving the image matching problem. We have evaluated the proposed implementation for several images. Each node(ant) in search space corresponds to one generated image. These new generated images have been evaluated using different heuristic and pheromone function.

The ACS is able to find solutions for matching problem. The reason for the performance improvement achieved with ACS method is the inherent flexibility of ACS method. This flexibility allows it to dynamically find good paths in changing environments. The results show that well-designed ant algorithm may successfully compete with other metaheuristics in solving such highly constrained problems as the image matching.

As a future work to be done, we plan to propose an implementation of the ACS method that allows to provide certain quality of service for solving image segmentation and registration.

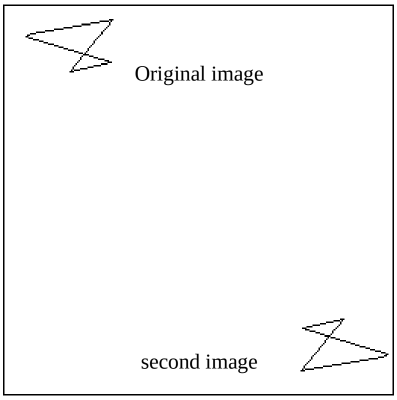
Initial Coordinates for translation(x,y)	Initial theta for rotation
(100,20)	10
(44,90)	50
(12,72)	94
(65,56)	190
(90,70)	16
(33,20)	100
(55,40)	75
(20,88)	66
(122,88)	90
(66,40)	170

Table(2) –displays initial values for rotation and translation to modify first image(this is generated randomly).

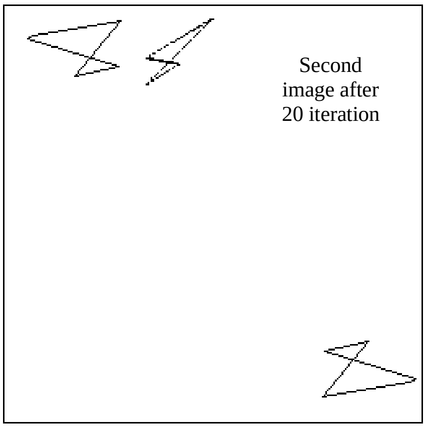
Iteration number	Type of search	Total No. of initial nodes	Total no.of intermedia-te nodes	Choi-ces nodes	Evaluation Value (dif)	Heuristic value of choices nodes (h=1/dif)	Pheromone value of choices nodes (ph=ph+h/1000)	Remov-able nodes

First iteration of Tabu	**Tabu**	10	100	1	**1.0e+005 * 1.4403**	**1/1.0e+005 * 1.4403**	**0.0069**	
				12	**1.0e+005 * 3.6884**	**1/1.0e+005 * 3.6884**	**0.0027**	
				25	**1.0e+005 * 3.9213**	**1/1.0e+005 * 3.4213**	**0.0026**	
				32	**1.0e+005 * 4.0305**	**1/1.0e+005 * 4.0305**	**0.0025**	
				41	**1.0e+005 * 4.2995**	**1/1.0e+005 * 4.2995**	**0.0023**	
				53	**1.0e+005 * 4.5045**	**1/1.0e+005 * 4.5045**	**0.0022**	
				44	**1.0e+005 * 4.7084**	**1/1.0e+005 * 4.7084**	**0.0021**	
				55	**1.0e+005 * 4.9252**	**1/1.0e+005 * 4.9252**	**0.0020**	
				61	**1.0e+005 * 5.7540**	**1/1.0e+005 * 5.7540**	**0.0017**	
				73	**1.0e+005 * 5.9787**	**1/1.0e+005 * 5.9787**	**0.0017**	
Second iteration of Tabu	**Tabu**	10	100	2	**1.0e+005 * 3.8191**	**1/1.0e+005 * 3.8191**	**0.0032**	
				15	**1.0e+005 * 3.9719**	**1/1.0e+005 * 3.9719**	**0.0025**	
				21	**1.0e+005 * 4.2253**	**1/1.0e+005 * 4.2253**	**0.0024**	
				32	**1.0e+005 * 4.4797**	**1/1.0e+005 * 4.4797**	**0.0022**	
				44	**1.0e+005 *4.9216**	**1/1.0e+005 * 4.9216**	**0.0020**	
				52	**1.0e+005 * 5.1651**	**1/1.0e+005 * 5.1651**	**0.0019**	
				65	**1.0e+005 * 5.5423**	**1/1.0e+005 * 5.5423**	**0.0018**	
				74	**1.0e+005 * 5.7363**	**1/1.0e+005 * 5.7363**	**0.0017**	
				85	**1.0e+005 * 5.9390**	**1/1.0e+005 * 5.9390**	**0.0017**	
				95	**1.0e+005 * 6.4486**	**1/1.0e+005 * 6.4486**	**0.0016**	
First iteration of ACO	**ACO**	10	100	3	**1.0e+005 * 0.8415**	**1/1.0e+005 * 0.8415**	0.0035	
				15	**1.0e+005 * 4.0918**	**1/1.0e+005 * 4.0918**	0.0024	
				22	**1.0e+005 * 4.4958**	**1/1.0e+005 * 4.4958**	0.0022	
				32	**1.0e+005 * 4.7665**	**1/1.0e+005 * 4.7665**	0.0021	
				42	**1.0e+005 * 4.9469**	**1/1.0e+005 * 4.9469**	0.0020	
				55	**1.0e+005 * 5.2321**	**1/1.0e+005 * 5.2321**	0.0019	**74**
				64	**1.0e+005 * 5.5653**	**1/1.0e+005 * 5.5653**	0.0018	**81**
				74	**1.0e+005 * 5.7913**	**1/1.0e+005 * 5.7913**	**0.0017**	**94**
				81	**1.0e+005 * 6.2612**	**1/1.0e+005 * 6.2612**	**0.0016**	
				94	**1.0e+005 * 6.9078**	**1/1.0e+005 * 6.9078**	**0.0014**	
Third iteration of Tabu	**Tabu**	7	70	3	**1.0e+005 * 1.0445**	**1/1.0e+005 * 1.0445**	**0.0039**	
				15	**1.0e+005 * 3.8088**	**1/1.0e+005 * 3.8088**	**0.0026**	
				21	**1.0e+005 * 4.4164**	**1/1.0e+005 * 4.4164**	**0.0023**	
				32	**1.0e+005 * 4.9425**	**1/1.0e+005 * 4.9425**	**0.0020**	
				45	**1.0e+005 * 5.2313**	**1/1.0e+005 * 5.2313**	**0.0019**	
				53	**1.0e+005 * 5.5726**	**1/1.0e+005 * 5.5726**	**0.0018**	
				64	**1.0e+005 * 5.7156**	**1/1.0e+005 * 5.7156**	**0.0017**	
Fourth iteration of Tabu	**Tabu**	7	70	1	**1.0e+005 * 0.9149**	**1/1.0e+005 * 0.9149**	**0.0040**	
				13	**1.0e+005 * 3.6997**	**1/1.0e+005 * 3.6997**	**0.0027**	
				25	**1.0e+005 * 4.1841**	**1/1.0e+005 * 4.1841**	**0.0024**	
				34	**1.0e+005 * 4.9562**	**1/1.0e+005 * 4.9562**	**0.0020**	
				44	**1.0e +005 * 5.3127**	**1/1.0e +005 * 5.3127**	**0.0019**	
				55	**1.0e+005 * 5.6666**	**1/1.0e+005 * 5.6666**	**0.0018**	
				65	**1.0e+005 * 5.8334**	**1/1.0e+005 * 5.8334**	**0.0017**	
Second iteration of ACO	**ACO**	7	70	1	**1.0e+005 * 0.9379**	**1/1.0e+005 * 0.9379**	0.0047	
				15	**1.0e+005 * 3.9187**	**1/1.0e+005 * 3.9187**	0.0026	
				24	**1.0e+005 * 4.7138**	**1/1.0e+005 * 4.7138**	**0.0021**	**24**
				33	**1.0e+005 * 4.9059**	**1/1.0e+005 * 4.9059**	**0.0020**	**33**
				44	**1.0e+005 * 5.0669**	**1/1.0e+005 * 5.0669**	**0.0020**	**44**
				54	**1.0e+005 * 5.3098**	**1/1.0e+005 * 5.3098**	**0.0019**	**54**
				63	**1.0e+005 * 5.5502**	**1/1.0e+005 * 5.5502**	**0.0018**	**63**
Fifth iteration of Tabu	**Tabu**	2	20	3	**1.0e+005 * 1.0169**	**1/1.0e+005 * 1.0169**	**0.0098**	
				11	**1.0e+005 * 3.5201**	**1/1.0e+005 * 3.5201**	**0.0028**	

Table (3)-displays the results of Tabu and Aco search for several iterations.



Figure(2):shows the two images



Figure(3):shows the modified image after 20 iterations

REFERENCES

[1] Blum, C. (2005) “Ant Colony Optimization and Swarms intelligence” ,MIT,Press, Cambridge, MA.

[2] Botley, L. (2006) “Artificial intelligence applied to network load balancing using Ant Colony Optimization”, in IEEE Transaction on Evolutionary Computation.

[3] Bouchouicha, M. and Ben K.Helifa, M.(1999) **A Genetic algorithm approach for image matching** ,University de Toulon et du Var. , Laboratories SIS-AI.

[4] Christodoulou, S.(2005) “Optimal Truss design using Ant Colony Optimization”, Department of Civil and Environmental Engineers University of Cyprus,P.O.BOX 20537, 1678 Nicosia, Cyprus.

[5] Cord´on O. (2005) “Hybridizing Evolutionary Computation and Ant Colony Optimization. Applications to Fuzzy Rule Learning and Bioinformatics Problems”. Dept. of Computer Science and Artificial Intelligence. E.T.S.I. Informatics, University of Granada.18071-Granada. Spain.

[6] Dorigo, M Di Caro G. (2001) Ant Algorithms for Discrete Optimization, IRIDIA, Universities Libre de Bruxelles,Brussels, Belgium.

[7] Gagn´e, C. & Gravel M. &Wilson L. Price (2001) “A look-ahead addition to the ant colony optimization met heuristic and its application to an industrial scheduling problem”, Department d’Informatique et de Math´ematique, Universities du Qu´ebec `a Chicoutimi 555, Boul. de l’Universit´e, Chicoutimi, Qu´ebec, Canada, G7H 2B1.

[8] Glover, F. & Laguna M.(1997) “Tabu Search”, University of Colorado at Boulder, Hardbound , ISBN 0-7923-9965-x,July 1997,408 pp.

[9] Grandchamp, E. & Charvillat V.(2001) “Metaheuristics to Design Satellite Constellation”, Alcatel Space Industries, 26, Avenue J.F. , Champollion, BP 1187,F-31037 Toulouse cedex, France.

[10] Gambardella, L.& Dorigo M.(1997) “Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem”, University Libre de Bruxelles ,Belgium, Accepted for publication in the IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997. In press.

[11] Grandchamp, E. and Charvillat V.(2001) "Meta heuristics to Design Satellite Constellation" ,Department of Computer Science and Engineering .The Chinese University of Hong Kong. 1998.

[12] Laguna, M. and Glover F.(1996) "What's Tabu Search",(adapted from: Handbook of Combinatorial Optimization, Panos Pardalos and Ding-Zhu Du (Eds.), Kluwer, 1998; Colorado Business Review, vol. LXI, no. 5, September 1996).

[13] Maria Gambardella, L. & Dorigo M. (1999) "Ant Colony Optimization: ants inspired systems for combinatorial optimization", IRIDIA, Universities Libre de Bruxelles,Brussels, Belgium.

[14] Matthew, D. Russell and John, A. Clark and Stepney, S.(1999) "Making the Most of Two Heuristics: Breaking Transposition Ciphers with Ants", Department of Computer Science, The University of York.

[15] Merkel, D. ; Middendorf, M. and Schmeck, H. (2000) "Pheromone Evaluation in ant colony optimization", Institute for applied Computer Sciences and Formal description methods(AIFB), University of Karlsuhe,D-67135 Karlsuhe ,Germany.

[16] Meshoul, S. and Batouche, M. (2002) "Ant Colony System with Extremal Dynamics for Point Matching and pose estimation", Batouche. Computer vision Group, LIRE labrotory.,2002.

[17] Morillo, P. ,Fernandoez, M. and Orduna, J. M.(2003) "An ACS-Based Partitioning Method for Distributed Virtual Environment Systems", Department of information ,University of Valencia.

[18] Nasraoui, O.(1999) "Tabu Search", slides for CECS 694:topics in combinational optimization , University of Louisville.

[19] Samadzadegan, F.& Azizi A. M. Petrou & M. Mirmehdi (1996) "**Range Data Analysis**", Lecture Notes in Computer Science, Springer Verlag, Germany.

[20] Sammoud1, O., Sorlin2 S., Solnon2 C. and Gh´edira1 K.(2000) "A Comparative Study of Ant Colony Optimization and Reactive Search for Graph Matching Problems", LIRIS, CNRS UMR 5205, bˆat. Nautibus, University of Lyon I 43 Bd du 11 novembre, 69622 Villeurbanne cedex, France.

[21] Simunic, K. & Loncaric S.(1997) "A Genetic Search based Partial Image Matching", Faculity of Electrical Engineering and computing, University of Zagrreb, Unska 3,1000 Zagreb, Croatia.

[22] Socha, K. , Sampels M. & Manfrin M.(2003) "Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of- the Art", IRIDIA, University of Libre de Bruxelles, CP 194/6, Av.Franklin D. Roosevelt 50, 1050 Bruxelles, Belgium.

[23] Wagner1 S. ; Affenzeller1 M. and KhalilI.(2000)"AGENT-BASEDROBLE M SOLVING:THE ANT COLONIES METAPHOR", Institute of Systems Science Department of Telecooperation Systems Theory and Information Technology Johannes Kepler University Linz Johannes Kepler University Linz Austria, Austria.

[24] Yang, Y. Petrovic S. and Graham Kendall (2002) "A tabu search approach for graph-structured case retrieval", In IOS Press, editor, Proc. of the Starting Artificial Intelligence Researchers Symposium (STAIRS 2002), pages 55–64, 2002.

25) د.خطيب, محمد(2005) "خوارزمية النمل" ، الدرس الرابع،كلية الهندسة المعلوماتية في سوريا,CSC-SY .

26) د.خطيب, محمد(2005) "خوارزمية النمل" ، الدرس الخامس،كلية الهندسة المعلوماتية في سوريا,CSC-SY .

27) د.خطيب، محمد(2005) "خوارزمية أسراب الطيور", كلية الهندسة المعلوماتية في سوريا,CSC-SY .