

# Multi-Colony Ant Systems for Multi-Hose Routing

TGI Fernando

Dept. of Statistics and Computer Science  
University of Sri Jayewardenepura  
Gangodawila, Nugegoda, Sri Lanka

Tatiana Kalganova

School of Engineering and Design  
Brunel University  
Uxbridge, UB8 3PH, United Kingdom

## ABSTRACT

Ant System (AS) is a general purpose heuristic algorithm inspired by the foraging behaviour of real ant colonies. AS and its improved versions have been successfully applied to difficult combinatorial optimization problems such as travelling salesman problem, quadratic assignment problem and job shop scheduling. In this paper, two versions of multi-colony ant systems that are extensions to the AS are proposed for the multi-hose routing. In both versions, each colony of ants searches for an optimum path between two end points (or commodities). While each colony searches for optimum paths, they try to maximum use of other colonies paths (sharing paths, or bundling) for easy handling of multiple paths. The first version uses a single pheromone matrix for all colonies and the second version uses different pheromone matrices for each colony and a modified random propositional rule to attract ants toward foreign pheromones. The tessellated format of the obstacles was used in the algorithm instead of the original shapes of the obstacles. As a result of using this format, the algorithm can handle freeform obstacles and speed up the algorithm when checking the collision detections. The experimental results show that there is no significant difference in the quality of the solutions produced by two versions and the first version takes less computation time. Further first version needs low computer memory and one parameter lesser than of the second version.

## Keywords:

Multi-colony ant system, Hose routing, Shared paths, Bundling, Foreign pheromone, Tessellated format, Freeform obstacles

## 1. INTRODUCTION

The ant system [1, 2, 3] is a kind of a collective search algorithm inspired by the behaviour of real ants for searching foods from their nest. The originally blind ants are able to find astonishing good solutions to shortest path problems between food source and their nest. The communication between the ants is based on pheromone trails that may be laid by individual ants. An ant's tendency to choose a specific path depends on the intensity of the pheromone trail on the path, i.e. the stronger the pheromone trail a path has the higher probability that an ant will follow that particular path. Over time, the pheromone trail evaporates and it loses intensity if no more pheromone is laid down by other ants. If a large number of ants choose a specific path, the strength of pheromone of this path increases and more ants likely to follow this path. AS imitates the foraging behaviour of natural ants and allows the application of this search metaphor to the solutions of hard combinatorial optimization problems like the Travelling Salesman Problem [1, 2, 3], the Quadratic Assignment Problem [4], job-shop scheduling problem [5], and Automatic Single Hose Routing Problem in 3D space [16, 17].

This scenario can be modelled in a network (laid in the free space of the environment) as follows. Let  $G = (V, E)$  be an edge-weighted undirected graph representing a

network in which the nodes represent the terminating cities (start or end nodes) or intermediate cities. Let  $T = \{(S_c, E_c) | S_c \neq E_c \in V, c = 1, 2, \dots, n\}$  be a list of commodities, i.e. pairs of nodes in  $G$ , representing terminating points demanding to be connected by a path in  $G$ .  $T$  is said to be realizable in  $G$ , if there exist paths (disjoint or not) from  $S_c$  to  $E_c$  in  $G$ , for every  $c = 1, 2, \dots, n$ . The question of whether  $T$  is realizable was early known to be NP-complete in arbitrary graphs [19] as well as specific types of graphs (e.g. disjoint path problem) [20, 21, 22, 23, 24].

The combinatorial optimization version of the problem consists of finding the optimum set of paths between the commodities and maximising the shared (or common) length of these paths. i.e. it needs to be find a solution such that

$$\min \sum_{c=1}^n l(P_c) \text{ and } \max \sum_{c=1}^n \sum_{\substack{c'=1, \\ c' \neq c}}^n l(P_c) \cap l(P_{c'}) \quad (1)$$

where  $P_c$  and  $P_{c'} (c \neq c')$  are two paths between  $(S_c, E_c)$  and  $(S_{c'}, E_{c'})$  respectively,  $l()$  represents the length of the argument and  $P_c \cup P_{c'}$  represents common edges (or shared edges) between  $P_c$  and  $P_{c'}$ .

In this paper, two types of multi-colony ant systems (MCAS-MHR-1 and MCAS-MHR-2) are proposed for multi-hose routing; both are extensions to the ant system (AS). In MCAS-MHR-1, each colony's task is to search for an optimal path between two points such that it shares other colonies' optimal paths (bundling) as much as possible. MCAS-MHR-1 uses only a single pheromone matrix for all the colonies. Pheromone updating is based on a weighted sum of total path length and shared path length between the paths. MCAS-MHR-2 is very similar to MCAS-MHR-1, but it uses a separate pheromone matrix for each colony and adds an additional term to random propositional rule defined in the original ant system (AS) so as to ants prefer to select paths that were used by not only the same colony but also the ants of other colonies.

Previously, hose routing was addressed in various approaches. These algorithms have been developed from stationary 2D workspace and simple objects to more complex 3D environment, dynamic, multi-constraint and multi-objective problems. Methods for pipe routing can be traced back to techniques for robot path planning that have been traditionally classified into four major categories: Skeleton search (roadmap) [14], the Cell Decomposition approach [14, 25], the Potential Field method [14] and Mathematical Programming method [14, 25].

Skeleton approach involves capturing the set of feasible motions in a network of one dimensional lines and conducting a graph search of this network [26]. The Cell Decomposition approach consists of decomposing the free space into cells and connecting the start and goal configurations by a sequence of connected cells. In the Potential Field Method, a scalar math-

emathical function is constructed whose value is the minimum when the robot is at the goal configuration, and maximum near the obstacles [27]. The path from the start to the goal is determined by putting a small marble at the start and following its movement. The Mathematical Programming approach deals with computing the path as a mathematical objective function and trying to minimize it while satisfying constraints (obstacle avoidance).

Mathematical Programming [14] techniques would be further classified into deterministic and non-deterministic methods based on the search algorithms employed. Deterministic techniques guarantee the same solution for a problem when run at different times with the same starting solution, while non-deterministic techniques generate different solutions to the same problem at different runs due to the randomness involved in the solution process. Deterministic methods such as the linear and non-linear programming methods usually behave inefficiently with highly non-linear and sometimes discontinuous problems like the pipe routing and often result in a local optimum. In contrast, non-deterministic algorithms such as genetic algorithms, simulated annealing and ant colony algorithms are aimed at generating a set of globally good solutions. This feature is of practical relevance in engineering applications.

In [28], pipe routing problem was solved from a cable harness routing perspective. Initial solution is generated without considering the obstacles. The obstacles are introduced individually, thereby intensifying the complexity of the problem gradually. In [29], the research was focused on developing a pipe routing algorithm to solve for non-orthogonal routes. Simulated annealing is the optimization algorithm used to obtain the optimal route. They utilized the versatility of this non-deterministic technique to overcome limitations associated with orthogonal routing and thus generated shorter routes. In [30], the authors considered a simplified routing problem in the domain of auto-mobile assembly maintainability. Their method changes the search resolution dynamically with respect to the complexity of the environment.

In [14], a non-deterministic optimization approach based on Genetic Algorithms was proposed to generate pipe routing solution sets. Objects are represented in tessellated format and it offers huge benefits in computation as well as usage. This approach can handle 3D free-form obstacles as the algorithm uses the tessellated format.

Previously, research has been primarily focused on automatic single hose routing with the use of either deterministic or non-deterministic approaches. However, none of the above algorithms has been designed for routing multiple hoses in parallel.

The rest of the paper is organized as follows. Section 2 describes the Ant System (AS) in detail. Section 3 presents the proposed extensions to the ant system for multi-hose routing. Section 4 describes how to apply these extensions to multi-hose routing in 3D space. Section 5 presents the experimental results and finally section 6 concludes the paper.

## 2. ANT SYSTEM

Multi-colony ant systems (MCAS-MHR-1 and MCAS-MHR-2) that are proposed in this paper are based on the ant system (AS). Before discussing the multi-colony extension of the ant system, the original ant system (AS) is described [16, 17].

Real ants are able to find the shortest path between their nest and a food source. Communication between the ants

is based on a pheromone trail deposited by individual ants. An ant's tendency to choose a specific path depends on the intensity of the pheromone trail on the path, i.e. the stronger the pheromone trail a path has the higher the probability that an ant will follow that particular path. Over time, the pheromone trail evaporates and it loses intensity if no more pheromone is laid down by other ants. If a large number of ants choose a specific path, the intensity of this trail increases and more ants tend to choose that path.

When applying the Ant System to single hose routing problem [16, 17] (finding the optimal or near optimal path between two nodes in a network), a trail strength is associated with every edge, to represent the pheromone. Initially, all ants are set on the start node (nest) and they construct tours to the end node (food source). When an ant visited to a node, the ant knows the straight line distance to the end node, the trail strength (pheromone strength) on the connecting edges, and which nodes have already been visited. Based on this knowledge ants choose the next node probabilistically. Once all ants have completed a certain number of turns or moves ( $N^{turns}$ ) or a cycle, a global pheromone updating rule (global updating rule, for short) is applied (See Fig. 1): a fraction of the pheromone evaporates on all edges (edges that are not refreshed become less desirable); each ant that was able to finish a complete tour deposits an amount of pheromone on the edges which belong to its tour in proportion to how short its tour was (in other words, edges which belong to many short tours receive the greater amount of pheromone).

After global updating, the current set of ants is removed from the civilization and another set of ants starts from the start node to explore the end node. The process is iterated until the prescribed stopping criterion (e.g. 100 cycles) is met. Note that, the parameter  $N^{turns}$  is set such that most of the ants in each set are able to reach the end node.

The state transition rule used by the ant system, called random-proportional rule, is given by (2) and gives the probability with which ant  $k$  in node  $r$  chooses to move to node  $s$  [11],

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where  $\tau(r, s)$  is the pheromone level on edge  $(r, s)$ ,  $\eta(r, s)$  is the inverse of the distance from node  $s$  to the end node (heuristic information),  $J_k(r)$  is the set of neighbour nodes of  $r$  that remain to be visited by the ant  $r$  positioned on the node  $r$ ,  $\alpha$  and  $\beta (> 0)$  are parameters which determine the importance of pheromone and heuristic information, respectively.

In the ant system, the global updating rule is implemented as follows. Ants that were able to complete their tour within the number of allocated turns ( $N^{turns}$ ) allow updating pheromone levels of their visited edges according to [11],

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \sum_{k=1}^m \Delta\tau_k(r, s) \quad (3)$$

where

$$\Delta\tau_k(r, s) = \begin{cases} \frac{1}{L_k}, & \text{if } (r, s) \in \text{tour done by ant } k \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$0 < \rho < 1$  is a pheromone decay parameter,  $L_k$  is the length of the tour performed by ant  $k$ , and  $m$  is the number of ants that were able to complete their tours within the stipulated number of turns ( $N^{turns}$ ).

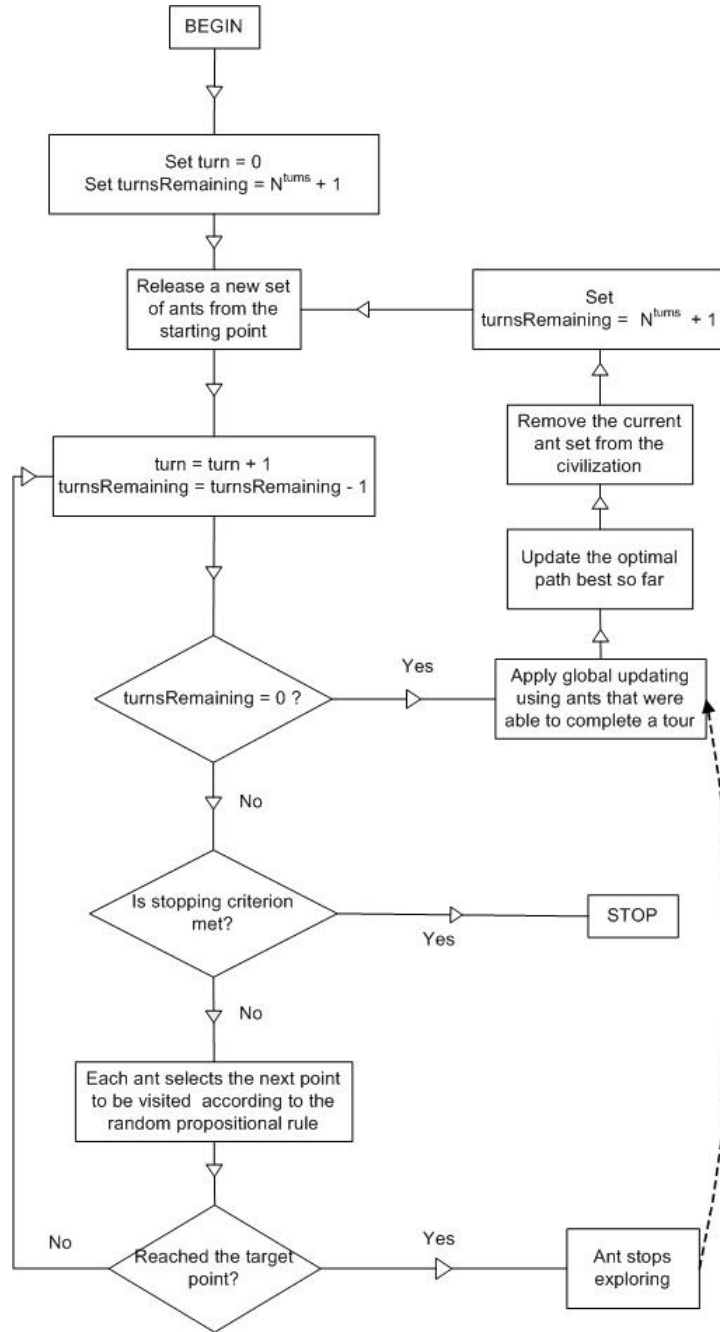


Fig. 1: Flowchart of Ant System

### 3. MULTI-COLONY ANT SYSTEM FOR MULTI-HOSE ROUTING

The multi-colony ant systems proposed in this paper uses  $n$  colonies to explore paths between  $n$  commodities. Each ant in the colony  $c$  explores paths between the start node  $S_c$  and the end node  $E_c$  by cooperating with other ants of the colony  $c$ . While the ants of the colony  $c$  walking along edges, they try to maximum use of common edges that is being used by ants of other colonies. As there are no direct communications between ants, this is obtained by the pheromone communication system of the proposed ant systems.

Version 1 (MCAS-MHR-1)

The problem which is interested in this research is to identify the paths between the commodities with maximum possible length of common edges. Therefore, in the proposed algorithm, the common edges of the paths receive a higher amount of pheromone when pheromone updating is occurred. As a result, an ant of a latter cycle encounters a shared edge, has a higher probability of choosing it than of non-sharing edge.

In this approach a single pheromone matrix is used for all the colonies. When an ant of a colony selecting the next move, it uses the random propositional rule as in the ant system defined in eq. (2). However, the most noticeable change to this algorithm is pheromone updating procedure. The pheromone updating of a path  $P_{ck}$  produced by an ant  $k$  of colony  $c$  is implemented based on

- The length of the path  $P_{ck}$ .
- The total shared length of the path  $P_{ck}$  with each path  $P_{ck'}$  produced by each successful ant  $k'$  of each colony  $c'$  ( $c' \neq c, c' = 1, 2, \dots, n$ ).

Global updating rule is implemented as follows. Ants which were able to complete their tour within the number of allocated turns ( $N^{turns}$ ) allow to updating pheromone levels of their visited edges according to the following equation:

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \sum_{c=1}^n \sum_{k=1}^{m_c} \Delta\tau_{ck}(r, s) \quad (5)$$

where

$$\Delta\tau_{ck}(r, s) = \begin{cases} F_{ck}, & \text{if } (r, s) \in \text{tour done by ant } k \text{ of colony } c \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where ( $0 < \rho < 1$ ) is a pheromone decay parameter,  $n$  is number of colonies (or number of commodities),  $m_c$  is the number of ants in colony  $c$  that were able to complete their tours within the stipulated turns  $N^{turns}$  and  $F_{ck}$  is the pheromone contribution of edges on path ( $P_{ck}$ ) produced by ant  $k$  of the colony  $c$  which is defined as follows:

$$F_{ck} = w_1 \cdot \frac{1}{L_{ck}} + w_2 \cdot \frac{s_{ck}}{L_{ck}} \quad (7)$$

where  $L_{ck}$  is the length of the path  $P_{ck}$  and  $s_{ck}$  is the total shared length of the path  $P_{ck}$  with paths produced by ants of other colonies (i.e. other than colony  $c$ ) which is defined as follows:

$$s_{ck} = \sum_{\substack{c'=1 \\ c' \neq c}}^n \sum_{k'=1}^{m_{c'}} l(P_{ck} \cap P_{ck'}) \quad (8)$$

where  $l(\cdot)$  is the length of the argument and  $m_{c'}$  is the number of ants in colony  $c'$  that were able to complete their tours within the stipulated turns  $N^{turns}$ ;  $w_1$  and  $w_2$  ( $w_1 + w_2 = 1$ ) are two weights that measure the importance of the length of the path  $P_{ck}$  and total shared length  $s_{ck}$  respectively.

## Version 2 (MCAS-MHR-2)

Unlike version 1, this algorithm uses a separate pheromone matrix for each colony (or commodity). This algorithm is designed similar to the multi-colony ant algorithm for the edge disjoint path problem described in [18]. In this approach also, an ant that encounters a pheromone trail left by an ant of the same type still have a high probability of following it. However, when it encounters an edge that was shared by other paths is more attracted than that of non-shared paths.

To implement ants' attractiveness towards foreign pheromones, the random propositional rule defined in eq. (2) is modified appropriately as follows:

$$p_k(r, s) = \begin{cases} \frac{[\tau_c(r, s)]^\alpha \cdot [\eta(r, s)]^\beta \cdot [\phi(r, s)]^\gamma}{\sum_{u \in J_k(r)} [\tau_c(r, s)]^\alpha \cdot [\eta(r, s)]^\beta \cdot [\phi(r, s)]^\gamma}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where  $\tau_c(r, s)$  is the pheromone trails left by colony  $c$  on the edge  $(r, s)$ ,  $\eta(r, s)$  is the inverse of the distance from node  $s$  to the end node ( $E_c$ ) of colony  $c$ ,  $J_k(r)$  is the set of neighbour nodes of  $r$  that remain to be visited by ant  $k$  positioned on the node  $r$ ,  $\alpha$  and  $\beta$  are parameters indicates the importance of pheromones and heuristic information respectively,  $\phi_c(r, s)$  represents the amount of pheromone trail not belonging to colony  $c$  on the edge  $(r, s)$  and is known as foreign pheromone and is defined as the sum of the pheromone trails left by all other colonies.

i.e.

$$\phi_c(r, s) = \sum_{\substack{c'=1 \\ c' \neq c}}^n \tau_{c'}(r, s) \quad (10)$$

and  $\gamma$  is a parameter that indicates the relative importance of foreign pheromone trails left by other colonies. If  $\gamma$  is set to zero, ants calculate the probability base on the problem heuristic and the pheromones of its own colony in the manner identical to original ant system [see eq. (2)]. If  $\gamma$  is increased, the probability of choosing an edge with a large amount of foreign pheromone trail is also increased and thus it tends to select edges shared with the previous paths.

Global updating rule is implemented as follows. Ants which were able to complete their tour within the number of allocated turns ( $N^{turns}$ ) allow updating pheromone levels of their visited edges according to:

$$\tau_c(r, s) \leftarrow (1 - \rho) \cdot \tau_c(r, s) + \sum_{k=1}^{m_c} \Delta\tau_{ck}(r, s) \quad (11)$$

where  $m_c$  is the number of ants in colony  $c$  that were able to complete their tours within the stipulated turns  $N^{turns}$  and  $\Delta\tau_{ck}(r, s)$  is defined as in eqs. (6), (7) and (8).

In the ant colony algorithms presented above, it is necessary to compare two paths produced by the same colony and compare the entire solution produced by all colonies in the current cycle with the previously generated best solution. Following definitions are adopted for comparing them.

### (1) Finding the best path of a colony in a cycle

As the algorithm needs to be considered the shared distance with paths produced by other colonies, it is not sufficient to consider only the length of the path produced by an ant. Thus, the pheromone contribution  $F_{ck}$  defined in (7) is used to compare two paths in the same colony. i.e.  $F_{ck} > F_{ck'}$  means the ant  $k$  produced a better path than ant  $k'$ .

### (2) Updating the total solution

From part 1, it is noticeable that how to find the best path of each colony in a cycle. The next step is to compare current solution (best paths produced by colonies) with the previous best solution produced by an earlier cycles. For this, two criteria are used; *strength.1* (reciprocal of total length of the solution)

$$\text{strength.1} = \frac{1}{L} \quad (12)$$

and *strength.2* (a weighed measure of the total length and total shared distance) which is defined as follows:

*strength.2* of a solution  $\{P_j\}, j = 1, 2, \dots, n$  is

$$\text{strength.2} = w_1 \cdot \frac{1}{L} + w_2 \cdot \frac{s}{L} \quad (13)$$

where  $L$  is the total length of paths  $P_j (j = 1, 2, \dots, n)$  and  $s$  is defined as follows:

$$s = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n l(P_i \cap P_j) \quad (14)$$

where  $l(\cdot)$  denotes the length of the argument.

The solution produced by a cycle is considered to be better than of the earlier solution if it is improved both *strength.1* and *strength.2* or was improved one of the criteria while the other criteria is remained the same.

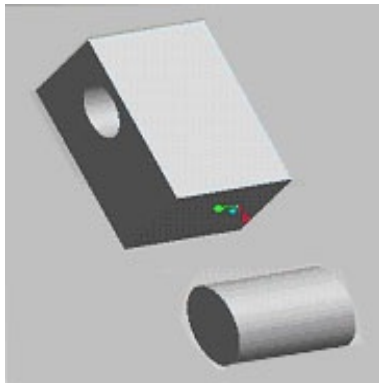


Fig. 2: Original CAD model

#### 4. MULTI-HOSE ROUTING IN 3D SPACE

The proposed algorithms (MCAS-MHR-1 and MCAS-MHR-2) were tested on randomly generated networks on 3D CAD models. The problem interested was to find the optimum layout of pipes when routing of pipes between different commodities (or different start and end points). The pipe layout is considered to be optimum when it provides the shortest total length of pipes and maximum length of shared paths.

The proposed hose routing methods briefly contain the following steps:

- (1) Assign the start and end points for each commodity.
- (2) Create a network using randomly generated points in the free space and  $n$  commodities.
- (3) Perform the hose routing algorithm (MCAS-MHR-1 or MCAS-MHR-2).

When performing the step 2, the algorithm must check the laying of a pipe between two points is collision free with objects in the CAD model. For this purpose, tessellated representation of the original 3D model is passed into the C++ collision detection library RAPID.

##### (1) The Tessellated Format

The .stl (STereoLithography) format or tessellated format [14] is an ASCII or binary file used in manufacturing. It is a list of triangular planes that describes a computer generated solid model. This is the standard input for most RAPID prototyping machines. The .stl file defines an objects surfaces as a set of adjacent triangles (Figs. 2 and 3). The file basically contains the X, Y and Z Cartesian coordinates of each vertex of the triangles, as well as the coordinates of the vectors normal to the triangles. With the tessellated format, each edge is shared only by two triangles. The tessellated model is an approximation to the real model and the accuracy of the tessellated model depends on the number of triangles used. In most CAD packages the number of triangles generated for the tessellated model can be controlled. Models used in this paper, were generated using the CAD package Pro/Engineer and its programming toolkit Pro/Toolkit.

Standard collision detection software (such as RAPID) requires polygonal models composed entirely of triangles that are an approximated model of the original model. Thus, the tessellated representation of a 3D model can be passed to the collision detection program. Also, most CAD software supports the tessellated format.

##### (2) RAPID

RAPID (Robust and Accurate Polygon Interface Detection)

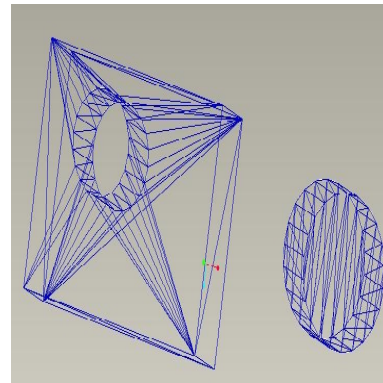


Fig. 3: Tessellated model of the original model given in Fig. 2

[15] is a C++ library developed at the Department of Computer Science, University of North Carolina, for interference detection (or collision detection) of large environments composed of unstructured models.

- It is applicable to polygon soups [15] - models that contain no adjacency information and obey no topological constraints. The models may contain cracks, holes, self-intersections, and non-generic (e.g., coplanar and collinear) configurations.
- It is numerically robust - the algorithm is not subject to conditioning problems and requires no special handling of non-generic cases (such as parallel faces).
- RAPID library is free for non-commercial use. It has a very simple user interface: the user needs to be familiar with only about five function calls.

RAPID accepts only polygonal models composed entirely of triangles, but does not require the model to have any particular structure. For example, some collision detection systems require the shapes to be well-formed solids - the surfaces must be “closed” so that there are a well-defined inside and outside.

##### (3) Hose Routing with Multi-Colony Ant Systems

Multi-hose routing with MCAS-MHR-1 and MCAS-MHR-2 is implemented in following steps.

- (a) Generate the tessellated representation of the original 3D model.
- (b) Generate a network of valid paths using randomly generated points in the free space and  $n$  commodities.
- (c) Obtain the best layout of hoses between the commodities using either MCAS-MHR-1 or MCAS-MHR-2.

In the first step, the tessellated representation of the obstacles is obtained as a text file from the CAD package. In the second step, this text file is passed to a program which incorporates the collision detection library RAPID. The following inputs are also supplied to this program:

- world size of the paths to be explored, given by the maximum and minimum of each axis coordinate -  $X^{min}, X^{max}, Y^{min}, Y^{max}, Z^{min}, Z^{max}$ ,
- number of commodities  $n$ ,
- coordinates of the each commodity  $(S_j, E_j)$  where  $j = 1, 2, \dots, n$ ,
- radius ( $r$ ) of the all hoses as they bundle together, and
- a text file containing the tessellated model of the original 3D model.

This program then generates a random network of valid paths from randomly generated points from the world and  $n$  commodities (See Fig. 4). When connecting two points, the program checks, with the aid of the C++ library, RAPID,

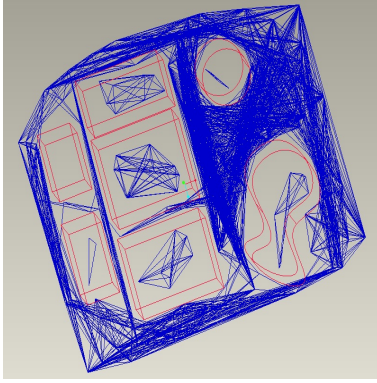


Fig. 4: Random network generated using 200 points in a CAD model

that the path between the two points is collision free (the axis of the hose cylinder lies on the line connecting the two points). For simplicity, a rectangular hexahedron is used that is centred on the line segment between the two points such that the cylindrical hoses (bundle) could be laid within it. This network data are stored in a text file for the use of the next step.

During the third step, the best layout of hoses between the  $n$  commodities is obtained using one of the proposed methods MCAS-MHR-1 or MCAS-MHR-2. The following inputs are supplied to these two methods:

- text file containing the random network of valid paths generated in the second step,
- number of commodities  $n$ ,
- coordinates of the each commodity  $(S_j, E_j)$  where  $j = 1, 2, \dots, n$ ,
- values for parameters  $\alpha, \beta, \gamma$  (only for MCAS-MHR-2), and  $\rho$ ,
- initial pheromone levels of edges (constant),
- number of ants to be released for each colony (or commodity),
- termination criteria,
- frequency at which the global pheromone update rule is applied ( $N^{turns}$ ).

## 5. EXPERIMENTAL RESULTS

MCAS-MHR-1 and MCAS-MHR-2 were implemented and their strengths and weaknesses were investigated empirically.

The parameter settings for both algorithms were: number of ants for each colony = 5, initial pheromone level on each edge = 1, pheromone decay parameter  $\rho = 0.01$ ,  $\alpha = 1$  and  $\beta = 5$ . Other parameter settings are included in the relevant experiment results.

The termination criterion for each experiment was set to 100 cycles.

All the simulations were carried out on a Pentium IV PC (Processor speed = 3.0 GHz, RAM = 512 MB) in the Microsoft Windows XP environment using Microsoft Visual C++ (.Net version).

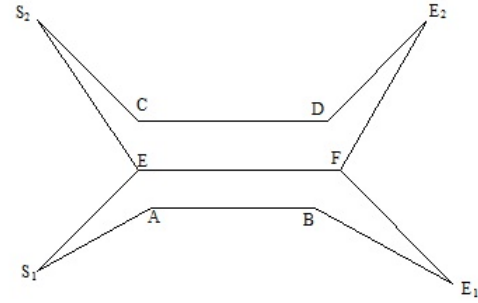


Fig. 5: Test graph 1 Finding the optimum paths between commodities  $(S_1, E_1)$  and  $(S_2, E_2)$ : Best solution is  $(S_1 E F E_1, S_2 E F E_2)$ .

Table 1. : Possible solutions for test graph 1

Solution No	Path from $S_1$ to $E_1$	Path from $S_2$ to $E_2$
1	$S_1 A B E_1$	$S_2 C D E_2$
2	$S_1 A B E_1$	$S_2 E F E_2$
3	$S_1 A B E_1$	$S_2 E S_1 A B E_1 F E_2$
4	$S_1 E F E_1$	$S_2 C D E_2$
5	$S_1 E F E_1$	$S_2 E F E_2$
6	$S_1 E F E_1$	$S_2 E S_1 A B E_1 F E_2$
7	$S_1 E S_2 C D E_2 F E_1$	$S_2 C D E_2$
8	$S_1 E S_2 C D E_2 F E_1$	$S_2 E F E_2$
9	$S_1 E S_2 C D E_2 F E_1$	$S_2 E S_1 A B E_1 F E_2$

### 5.1 Experiment 1: Demonstrating the potential of MCAS-MHR-1 and MCAS-MHR-2 on a test graph

The purpose of this experiment is, demonstrating the potential of the proposed two algorithms MCAS-MHR-1 and MCAS-MHR-2 on a graph whose optimal solutions between the given commodities are already known.

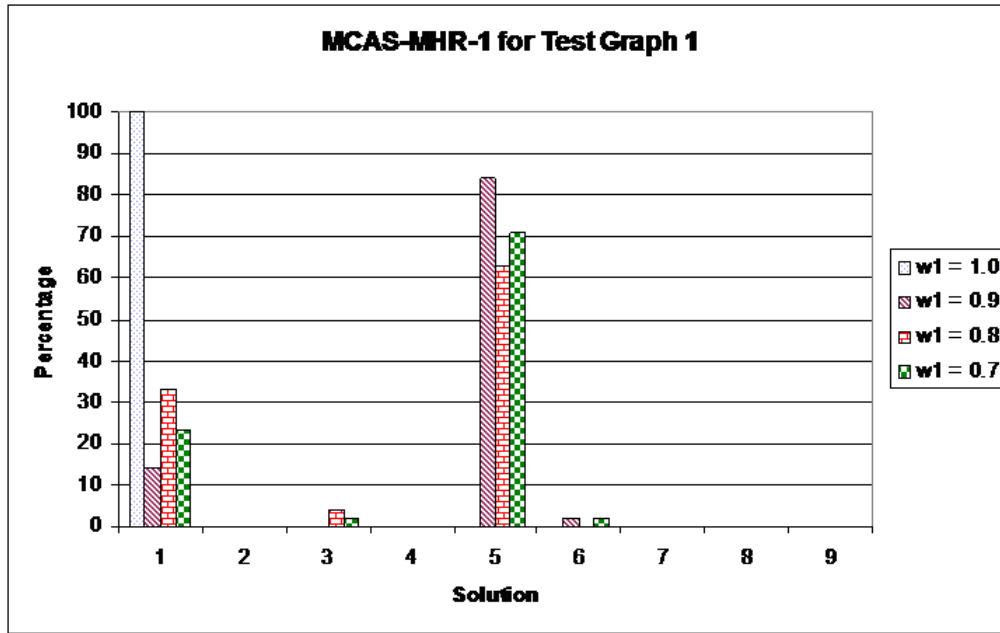
The test graph 1 (See Fig. 5) is used to find the best shared paths between the commodities  $(S_1, E_1)$  and  $(S_2, E_2)$ . There are 9 possible solutions for this simple graph (See TABLE 1). When considering the shortest paths between the two commodities  $(S_1, E_1)$  and  $(S_2, E_2)$ , the choice would be the solution 1. However, if paths need to be shared as much as possible while the total length of the paths is minimizing, algorithms must select the solution 5.

TABLE 2 shows the percentage of runs that reach to the each solution over 100 runs after 100 cycles on different values of  $w_1, w_2$  and  $\gamma$  (only for MCAS-MHR-2).

These results shows that both algorithms reach to the disjoint solution (solution 1) when the shared length between two paths are not considered for pheromone updating (See TABLE 2, Fig. 6 and Fig. 7). However, both algorithms reach to the solution 5 more than 70% of runs (except one case) when there is a contribution from the shared length between two paths for the pheromone updating (See Fig. 6 and Fig. 7). Highest percentage obtained for MCAS-MHR-1 is 84% and it is happened when  $w_1 = 0.9$  and  $w_2 = 0.1$ . Comparing with MCAS-MHR-1, MCAS-MHR-2 gives slightly better results in most of the  $\gamma$  values. For example, when  $w_1 = 0.9$  and  $w_2 = 0.1$ , MCAS-MHR-2 reaches to the solution 5, 91%, 88%, 89% and 93% of the runs for the  $\gamma$  values 1, 2, 4 and 5 respectively. Therefore, it possible to conclude empirically, performances of MCAS-MHR-2 are better as a result of using additional information (foreign pheromones).

**Table 2. : Percentage of each solution found on test graph 1 for different values of  $w_1, w_2$  and  $\gamma$  (for MCAS-MHR-2 only)**

MCAS-MHR-1																																				
(w1, w2)	(1, 0)									(0.9, 0.1)									(0.8, 0.2)									(0.7, 0.3)								
Solution	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
(%)	100	0	0	0	0	0	0	0	0	14	0	0	0	84	2	0	0	0	33	0	4	0	63	0	0	0	0	23	0	2	0	71	2	0	0	0
MCAS-MHR-2																																				
(w1, w2)	(1, 0)									(0.9, 0.1)									(0.8, 0.2)									(0.7, 0.3)								
Solution	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
$\gamma = 0$	100	0	0	0	0	0	0	0	0	23	0	0	0	77	0	0	0	0	3	0	0	0	95	2	0	0	0	23	0	2	0	71	2	0	2	0
$\gamma = 1$	100	0	0	0	0	0	0	0	0	9	0	0	0	91	0	0	0	0	0	0	0	0	94	6	0	0	0	16	0	2	0	78	2	0	2	0
$\gamma = 2$	100	0	0	0	0	0	0	0	0	12	0	0	0	88	0	0	0	0	15	0	2	0	79	2	0	2	0	4	0	2	0	92	2	0	0	0
$\gamma = 3$	100	0	0	0	0	0	0	0	0	18	0	0	0	80	2	0	0	0	20	0	0	0	78	0	0	2	0	14	0	2	0	78	6	0	0	0
$\gamma = 4$	100	0	0	0	0	0	0	0	0	6	0	3	0	89	0	0	0	0	8	0	0	0	92	0	0	0	0	6	0	2	0	90	2	0	0	0
$\gamma = 5$	100	0	0	0	0	0	0	0	0	3	0	0	0	93	2	0	2	0	5	0	2	0	93	0	0	0	0	18	0	0	0	82	0	0	0	0



**Fig. 6: Percentage of runs of MCAS-MHR-1 that reaches to the each solution of test graph 1**

## 5.2 Experiment 2: Applying MCAS-MHR-1 and MCAS-MHR-2 on a complex 3D model

The purpose of this experiment is, applying proposed two algorithms MCAS-MHR-1 and MCAS-MHR-2 on a complex 3D environment and comparing the results of the two algorithms. Both versions of algorithms were tested on following Pro/Engineer 3D model (See Fig. 8) using different values of  $w_1, w_2$  and  $\gamma$  (only for MCAS-MHR-2) for random networks of 200 points and 400 points. Other parameters of the algorithms were fixed as mentioned in the beginning of this section. Two algorithms were used for finding the best shared paths of 4 commodities. All simulations were carried out for 100 cycles.

For each trial, final path length of each commodity, connecting points of each path, time spent on running the trial (in seconds), total shared length of paths,  $strength_1$  and  $strength_2$  (defined in eqs. (12), (13) and (14) were recorded in text files.

Let's take  $\{Q_i^{(j)} | i = 1, 2, 3, 4\}$  is the best solution produced in the  $j$ th trial ( $j = 1, 2, \dots, 100$ ). Here  $Q_i^{(j)}$  means path between  $i$ th commodity of the best solution of the  $j$ th trial. Further, let's take  $\{R_i | i = 1, 2, 3, 4\}$  is the overall best solution among the best solutions  $[\{Q_i^{(j)} | i = 1, 2, 3, 4\}, j = 1, 2, \dots, 100]$  produced in 100 trials and  $t_i (i = 1, 2, \dots, 100)$  is the computation time of the  $i$ th trial.

Tables 3 - 6 summarize the following descriptive statistics obtained for the two algorithms over 100 trials after 100 cycles for different values of  $w_1, w_2$  and  $\gamma$  (MCAS-MHR-2 only) values for 200 point and 400 point networks.

—Average total length (Average total length of the paths of the best solutions over 100 trials)

$$\bar{Q} = \frac{\sum_{j=1}^{100} Q_j}{100}$$



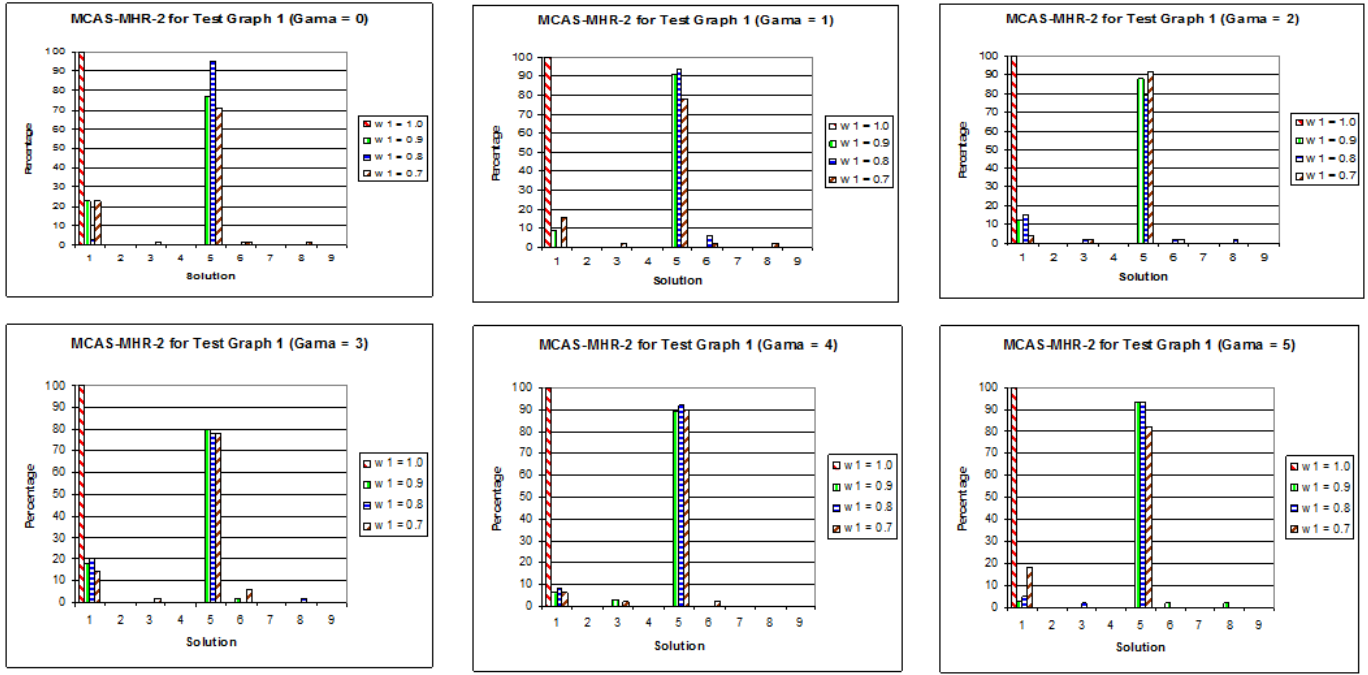


Fig. 7: Percentage of runs of MCAS-MHR-2 that reaches to the each solution of test graph 1

where

$$Q_j^\Sigma = \sum_{i=1}^4 l(Q_i^j)$$

- SD total length (Standard deviation of the total lengths of the paths of the best solutions over 100 trials) -  $SD(Q_j^\Sigma)$
- Averaged shared length (Average shared length of the best solutions over 100 trials)

$$\bar{S} = \frac{\sum_{j=1}^{100} S_j^\cap}{100}$$

where

$$S_j^\cap = \sum_{i=1}^4 \sum_{\substack{k=1 \\ k \neq i}}^4 l(Q_i^{(j)} \cap Q_k^{(j)})$$

- SD shared length (Standard deviation of shared lengths of the best solutions over 100 trials) -  $SD(S_j^\cap)$
- $P_1$  - % avg. shared length (Percentage of the average shared length to the average total length)

$$P_1 = \frac{\bar{S}}{Q} \cdot 100$$

- Total length of the overall best solution

$$R^\Sigma = \sum_{i=1}^4 l(R_i)$$

- Total shared length of the paths of the overall best solution

$$R^\cap = \sum_{i=1}^4 \sum_{\substack{k=1 \\ k \neq i}}^4 l(R_i \cap R_k)$$

- $P_2$  - % shared length of the overall best solution (Percentage of the shared length of the paths of the overall best solution to

the total length of the paths of the overall best solution)

$$P_2 = \frac{R^\cap}{R^\Sigma} \cdot 100$$

- Average time per trial

$$\bar{T} = \frac{\sum_{i=1}^{100} t_i}{100}$$

The best solution out of 100 runs is selected using the *strength<sub>2</sub>* [see eqs. (13) and (14)]; i.e. the solution with highest value of *strength<sub>2</sub>* is selected as the best solution. Here  $l(\cdot)$  is the length of the argument and  $R_i \cap R_k$  is the common edges between the paths  $R_i$  and  $R_k$ .

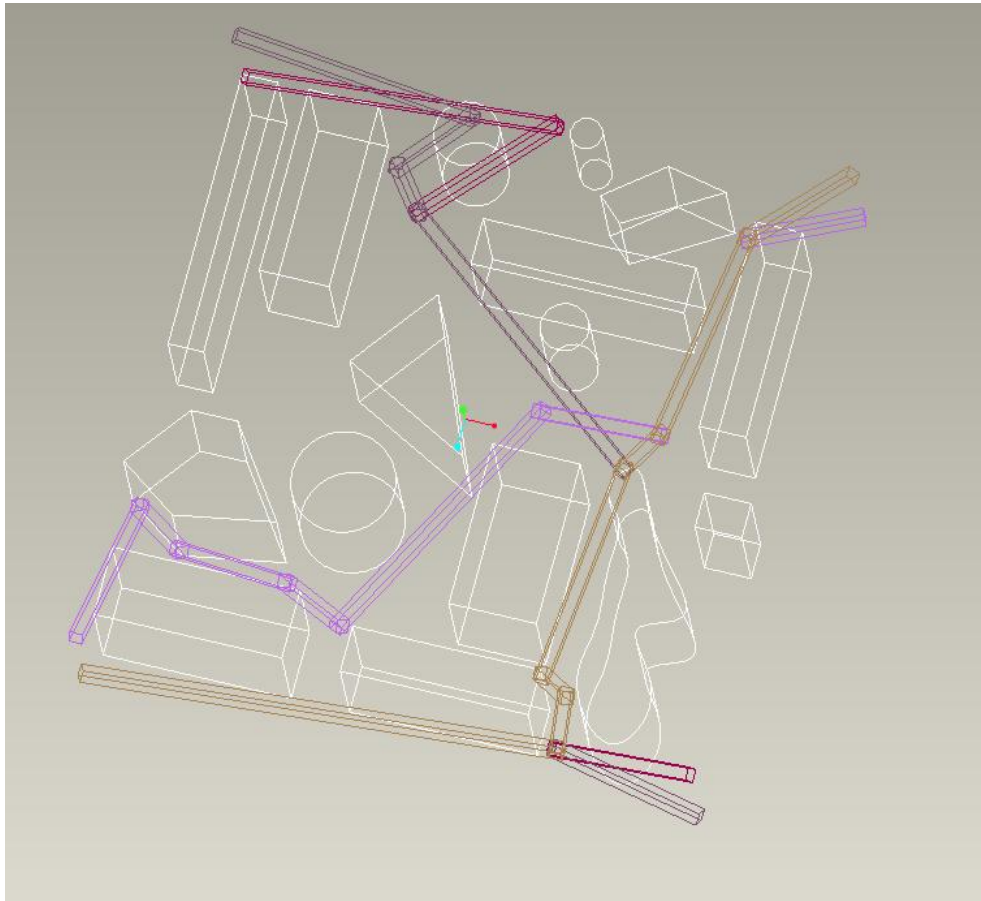
Fig. 9 shows how the average total lengths and the average shared lengths of MCAS-MHR-1 are changed over the different values of weights. Average shared length increases with  $w_1$  decreases. However, average total length also increases with low values  $w_1$ .

Fig. 10 and Fig. 11 show how the average shared lengths and average total lengths of MCAS-MHR-2 differ over the different values of  $w_1, w_2$ , and  $\gamma$ . As in the MCAS-MHR-1, average shared length and average total length increase with  $w_1$  decreases. It can be noted that for each value of  $\gamma$ , average shared length and average total length increase as  $w_1$  decreases.

It is noticeable that in both algorithms, average total length is increases as  $w_1$  decreases; thus the designer of the algorithm must select the values for  $w_1$  and  $w_2$  carefully. Low values of  $w_1$  implies that both total length and shared length of the solution increase and high values of  $w_1$  implies that both total length and shared length decrease. Hence the designer should select  $w_1$  and  $w_2$  such that it balances the optimality of both total length and shared length of paths.

When comparing the average shared lengths for the different  $\gamma$  values for specific weights (See Fig. 10), it is noted that





**Fig. 8: Tested Pro/Engineer Model**

$\gamma = 2$  gives highest average shared length or closer to highest average shared length in most of the cases of both 200 points and 400 points networks. Therefore, MCAS-MHR-2 with  $\gamma = 2$  is selected to compare with MCAS-MHR-1.

Fig. 12 compares the average shared lengths of MCAS-MHR-1 and MCAS-MHR-2 (with  $\gamma = 2$ ) against the different weights for the 200 and 400 points networks. For the 200 points network, this shows that MCAS-MHR-2 produces a significant average shared lengths for weights  $w_1 = 0.9999$  and  $0.999$  when comparing with the values of MCAS-MHR-1. However, it is not shown a significant difference for average shared lengths for MCAS-MHR-1 and MCAS-MHR-2 for the 400 points network. However, when comparing the average total lengths (see Fig. 13), there is no much difference between the two versions.

Total length and shared length of the overall best solutions of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) are compared in Fig. 14 and Fig. 15. In both of the 200 and 400 points networks, MCAS-MHR-1 gives higher shared values than MCAS-MHR-2 (4 out of 6). But when comparing the total lengths MCAS-MHR-2 produces less total length values in most cases.

When comparing the computational times of the two algorithms (see Fig. 16), computation time of MCAS-MHR-1 is less for more complicated network (400 points network).

When comparing the memory requirements, since MCAS-MHR-2 uses a single pheromone matrix for each colony and it needs more memory than of MCAS-MHR-1. Thus, the memory

requirement of MCAS-MHR-2 grows with the number of commodities used in the algorithm.

## 6. CONCLUSION

In this paper, two versions of multi-colony ant systems (called MCAS-MHR-1 and MCAS-MHR-2 respectively) have been introduced for routing multiple hoses/pipes in parallel. Two versions use a separate colony for each commodity. First version uses a single pheromone matrix for all colonies whilst the second version uses a separate pheromone matrix for each colony. Thus, ants in the second version are able to smell different pheromones individually laid by ants in the other colonies. But ants in the first version cannot recognize pheromones individually as all pheromones laid by ants on different colonies on an edge are summed up to a single value. Further a modification is introduced to the random propositional rule in version 2 to attract ants towards edges that were used by ants of other colonies. When pheromone updating, both methods evaluate the quality of a solution based on not only the total lengths of paths but also the shared length of paths.

Aim of this work is to apply these methods to multi-hose routing with maximum use of common edges. Initially, two methods have been tested on a simple test graph (See Experiment 1) using two commodities. In Experiment 2, two methods have been applied for multi-hose routing in a complex 3D CAD model using two randomly generated networks of size 200 points and 400 points and 4 commodities.

**Table 3. : Descriptive statistics of MCAS-MHR-1 for 200 points network**

	$\bar{Q}$	$SD(Q_j^x)$	$\bar{S}$	$SD(S_j^y)$	$P_1$	$R^z$	$R^\wedge$	$P_2$	$\bar{T}$ (Sec.)
$w_1 = 1.0, w_2 = 0.0$	3568.9	30.6	119.6	192.5	3.35	3490.32	148.50	4.26	56.27
$W_1 = .9999, w_2 = .0001$	3584.3	37.4	238.7	264.3	6.66	3595.12	887.32	24.68	57.03
$W_1 = .999, w_2 = .001$	3735.6	126.4	588.7	230.4	15.76	3960.28	1209.94	30.55	63.66

**Table 4. : Descriptive statistics of MCAS-MHR-1 for 400 points network**

	$\bar{Q}$	$SD(Q_j^x)$	$\bar{S}$	$SD(S_j^y)$	$P_1$	$R^z$	$R^\wedge$	$P_2$	$\bar{T}$ (Sec.)
$w_1 = 1.0, w_2 = 0.0$	3342.36	36.38	58.23	110.85	1.74	3238.01	241.34	7.45	128.61
$W_1 = .9999, w_2 = .0001$	3347.85	42.47	82.11	115.43	2.45	3238.44	241.34	7.45	129.52
$W_1 = .999, w_2 = .001$	3480.53	96.66	338.30	195.07	9.72	3577.07	828.75	23.17	128.28

In both algorithms, importance of the total length and shared length must be identified; accordingly respective weights ( $w_1$  and  $w_2$ ) must be selected. It is difficult to select pre-defined values for  $w_1$  and  $w_2$  for the all routing problems. Best way to get a better result is to run the algorithms with different values of  $w_1$  and  $w_2$  and select the appropriate solution from solutions produced over different values of  $w_1$  and  $w_2$ .

According to the results found, MCAS-MHR-2 performs well with  $\gamma = 2$  and gives better shared lengths compared to other  $\gamma$  values.

When comparing overall best solutions in different runs, MCAS-MHR-1 gives better shared lengths whilst MCAS-MHR-2 produces better total lengths in most cases.

When comparing the average shared lengths, MCAS-MHR-2 performs slightly better than of MCAS-MHR-1. Obviously this is a result of use of foreign pheromones in random propositional rule used in MCAS-MHR-2. However, MCAS-MHR-2 uses an additional parameter ( $\gamma$ ) and needs a separate pheromone matrix for each of the commodity; hence increases the memory requirement of the algorithm as the number of commodities increases.

Computational times for the two algorithms do not show a much difference for the simple network (200 points network). However, MCAS-MHR-1 takes less computational time for the 400 points network.

According to the results found in both versions, there is no significant difference in quality of the solutions between the two versions: MCAS-MHR-1 and MCAS-MHR-2. Thus, MCAS-MHR-1 is recommended for this type of problems as it takes less computation time, lesser no of parameters and needs low memory requirements.

Both of these algorithms are run on a single PC and next step is to implement each colony on different PCs (grid computing) and speed up the algorithms. Further, these two versions use a classical approach of multi-objective optimization (weighted sum approach) and another possibility of improving them is to use of Pareto optimization techniques.

## 7. REFERENCES

- [1] Dorigo, M., Maniezzo, V., & Colnari, A. (1996). The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13.
- [2] Colnari, A., Dorigo, M., & Maniezzo, V. (1991). Distributed Optimization by Ant Colonies. In proceedings of ECAL91 - European Conference on Artificial Life, Elsevier Publishing, pp. 134-142.
- [3] Stutzle, T. & Hoos, H. (1997). MAX-MIN Ant System and Local Search for the Travelling Salesman Problem, IEEE International Conference on Evolutionary Computation, pp. 309-314.
- [4] Maniezzo, V., Dorigo, M., & Colnari, A. (1994). The Ant System Applied to the Quadratic Assignment Problem. Technical Report IRIDIA/94-28, Universit Libre de Bruxelles, Belgium.
- [5] Dorigo, M., Maniezzo, V., & Colnari, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics - Part B, 26 (1), pp. 29-41.
- [6] Gambardella, L., & Dorigo, M. (1996). Solving Symmetric and Asymmetric TSPs by Ant Colonies. In IEEE Conference on Evolutionary Computation (ICEC'96), IEEE Press, pp. 622-627.
- [7] Bullnheimer, B., Hartl, R.F., & Strauss, C. (1997). A New Rank-based Version of the Ant System - A Computational Study. Technical report, University of Viena, Institute of Management Science.
- [8] Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). From natural to artificial swarm intelligence. New York: Oxford University Press.
- [9] Corne, D. Dorigo, M., & Glover, F. (Eds.). (1999). New ideas in optimization. Maidenhead, UK: McGraw-Hill.
- [10] Gambardella, L.M., & Dorigo, M. (1999). Ant algorithms for discrete optimization. Artificial Life 5: Massachusetts Institute of Technology. pp. 137-172.
- [11] Gambardella, L.M., & Dorigo, M. (1997). Ant Colony System: A cooperative learning approach to the travelling salesman problem. Evolutionary Computation, IEEE Transactions. pp. 53-66.
- [12] Sim, K.M., & Sun, W.H. (2003). Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions. IEEE Transactions, MAN, and CYBERNETICS -

**Table 5. : Descriptive statistics of MCAS-MHR-2 for 200 points network**

Weights	$\gamma$	$\bar{Q}$	$SD(Q_i^r)$	$\bar{S}$	$SD(S_i^r)$	$P_i$	$R^z$	$R^\wedge$	$P_i$	$\bar{T}$ (Sec.)
$w_1 = 1.0, w_2 = 0.0$	0	3567.02	30.70	133.25	202.09	3.74	3488.68	0.00	0.00	56.10
	1	3568.54	29.46	75.27	153.89	2.11	3489.08	518.03	14.85	56.20
	2	3572.95	31.52	92.53	166.40	2.59	3390.99	0.00	0.00	55.94
	3	3562.44	33.30	120.53	178.93	3.38	3460.98	104.40	3.02	55.56
	4	3573.30	28.56	148.12	215.85	4.15	3514.67	0.00	0.00	55.47
	5	3567.67	35.66	121.38	188.76	3.40	3461.55	43.69	1.26	55.21
$w_1 = .9999, w_2 = .0001$	0	3588.27	43.37	239.78	251.01	6.68	3493.48	800.54	22.92	56.26
	1	3587.12	45.09	259.04	233.30	7.22	3532.09	688.93	19.50	56.34
	2	3591.61	42.89	293.69	248.87	8.18	3574.13	910.93	25.49	56.11
	3	3587.27	43.32	282.10	247.06	7.86	3572.71	990.17	27.72	55.86
	4	3581.76	37.96	236.63	227.05	6.61	3554.92	740.03	20.82	55.73
	5	3593.44	38.78	310.84	243.90	8.65	3550.06	1004.85	28.31	55.56
$w_1 = .999, w_2 = .001$	0	3747.75	156.75	612.55	250.36	16.35	3812.51	1296.03	33.99	56.09
	1	3735.79	128.62	616.77	229.78	16.51	4065.98	1369.88	33.69	55.82
	2	3730.34	114.60	650.97	267.58	17.45	3609.02	1128.69	31.27	55.00
	3	3741.96	107.70	615.63	238.94	16.45	3774.27	1141.45	30.24	55.29
	4	3737.44	111.86	617.79	273.20	16.53	4084.38	1395.68	34.17	55.21
	5	3740.83	145.18	615.56	250.67	16.46	3548.82	1016.95	28.66	55.72

**Table 6. : Descriptive statistics of MCAS-MHR-2 for 400 points network**

Weights	$\gamma$	$\bar{Q}$	$SD(Q_i^r)$	$\bar{S}$	$SD(S_i^r)$	$P_i$	$R^z$	$R^\wedge$	$P_i$	$\bar{T}$ (Sec.)
$w_1 = 1.0, w_2 = 0.0$	0	3339.39	38.59	34.80	81.36	1.04	3239.44	0.00	0.00	135.04
	1	3342.62	40.62	41.78	92.60	1.25	3186.64	241.34	7.57	133.79
	2	3345.17	35.52	24.50	70.98	0.73	3261.64	0.00	0.00	135.93
	3	3342.08	35.89	44.93	94.09	1.34	3261.90	0.00	0.00	134.79
	4	3340.79	37.23	53.81	96.21	1.61	3234.37	241.34	7.46	135.75
	5	3335.33	45.24	31.06	78.62	0.93	3197.35	412.80	12.91	135.47
$w_1 = .9999, w_2 = .0001$	0	3343.75	40.04	70.17	105.89	2.10	3264.22	380.03	11.64	135.97
	1	3355.58	48.88	101.50	142.10	3.03	3244.43	516.50	15.92	135.30
	2	3349.84	43.95	92.01	125.33	2.75	3267.76	402.77	12.33	134.09
	3	3353.93	45.21	85.96	138.72	2.56	3232.26	241.34	7.47	136.08
	4	3351.63	42.97	107.48	160.30	3.21	3334.68	456.43	13.69	136.87
	5	3346.46	45.10	82.42	146.13	2.46	3196.40	241.34	7.55	135.22
$w_1 = .999, w_2 = .001$	0	3466.34	102.51	289.11	194.28	8.34	3528.09	768.50	21.78	135.19
	1	3472.51	89.87	346.19	195.18	9.97	3542.69	857.55	24.21	134.47
	2	3471.20	92.79	348.81	185.79	10.05	3482.27	747.83	21.48	133.85
	3	3470.18	114.92	319.52	197.79	9.21	3532.38	1015.93	28.76	134.56
	4	3477.17	102.54	355.33	187.84	10.22	3612.18	988.48	27.37	135.05
	5	3477.33	98.62	349.55	196.05	10.05	3475.93	840.10	24.17	133.75

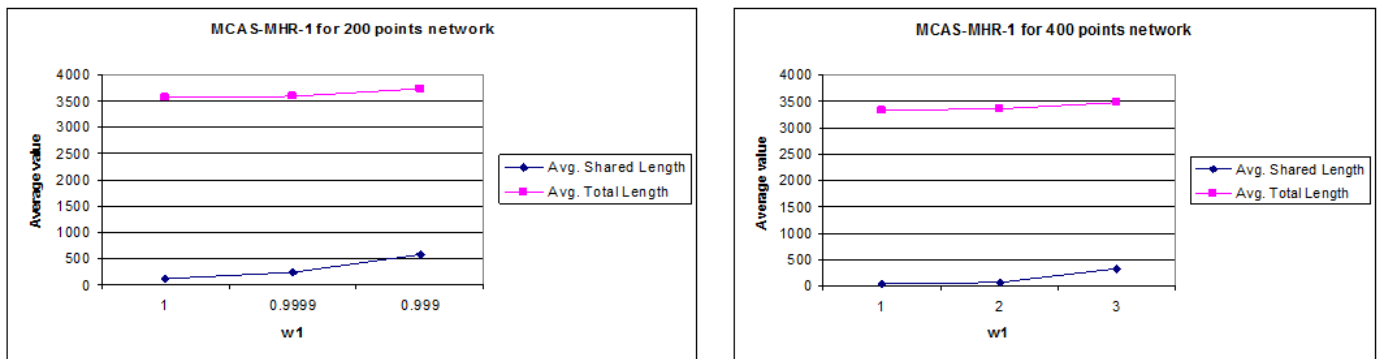


Fig. 9: Average values of total length and shared lengths of MCAS-MHR-1 of the 200 and 400 points networks against different values of weights

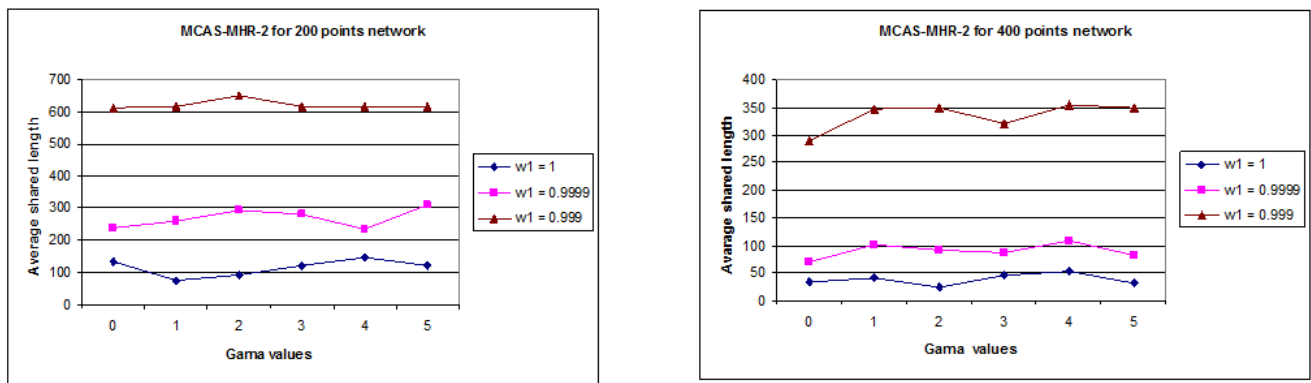


Fig. 10: Average shared lengths of MCAS-MHR-2 of the 200 and 400 points networks against different values of  $\gamma$  values

University Press. ISBN 0-19-513159-2.

- [14] Sandurkar, S., & Chen, W. (1998). GAPRUS - Genetic algorithms based pipe routing using tessellated objects. The journal of computers in industry.
- [15] Gottschalk, S., Lin, M.C., & Manocha, D. RAPID (Robust and Accurate Polygon Interface Detection). Available: <http://www.cs.unc.edu/geom/OBB/OBBT.html>
- [16] Thantulage, G., Kalganova, T. & Fernando, W.A.C. (2006). A Grid-based Ant Colony Algorithm for Automatic 3D Hose Routing. IEEE Congress on Evolutionary Computation, CEC 2006, Vancouver, Canada, Jul., 2006. pp. 48-55.
- [17] Thantulage, G., Kalganova, T. & Wilson, M. (2006). Grid Based and Random Based Ant Colony Algorithms for Automatic Hose Routing in 3D Space. Transactions on Engineering, Computing and Technology, Volume 14, International Journal of Applied Science, Engineering and Technology (IJASET), Enformatika, ISBN 1503-5313, ISBN 975-00803-3-5, Aug., 2006. pp. 144-150.
- [18] Nowe, A., Verbeeck, K. & Vrancx, P. (2004). Multi-type Ant Colony: The Edge Disjoint Paths Problem. Ants 2004, LNCS 3172, Springer-Verlag Berlin Heidelberg, 2004. pp. 202-213.
- [19] Karp, R. (1972). Complexity of Computer Computations. Chapter - Reducibility among Combinatorial Problems. Plenum Press, 1972. pp. 85-213.
- [20] Kramer, M. & Leeuwen, J.V. (1984). Advances in Computing Research, Volume 2: VLSI Theory. Chapter - The Complexity of Wire-routing and Finding Minimum Area Layouts for Arbitrary VLSI Circuits. Jai Press, 1984. pp. 129-146.
- [21] Middendorf, M. & Pfeiffer, F. (1993). On the Complexity of the Disjoint Path Problem. Combinatorica 13, 1993. pp. 97-107.
- [22] Nishizeki, T., Vygen, J. & Zhou, X. (2001). The Edge-disjoint Paths Problem is NP-Complete for Series-Parallel Graphs. Discrete Applied Mathematics 115, 2001. pp. 177-186.
- [23] Marx, D. (2004). Eulerian Disjoint Paths Problem in Grid Graphs is NP-Complete. Discrete Applied Mathematics 143, 2004. pp. 336-341.
- [24] Vygen, J. (1995). NP-Completeness of Some Edge-Disjoint Paths Problems. Discrete Applied Mathematics 61, 1995. pp. 83-90.
- [25] Ahuja, N. & Hwang Y.K. (1992). Gross Motion Planning - A Survey. ACM Computing Surveys, 24(3), 219-291.
- [26] Aurenhammer, F. (1991). Voronoi Diagrams - A survey of fundamental geometric data structure. ACM Computing Survey, 23(3) (Sept.), 345-405.
- [27] Koditschek, D. E. (1989). Robot Planning and Control via potential functions. Robotics Review, Vol 1, MIT Press, Cambridge, Mass.
- [28] Conru, A.B. & Cutkosky, M. R. (1993). Computational Support for Interactive Cable Harness Routing and Design. Advances in Design Automation, DE- 65(1), 551-558.
- [29] Szykman, S. and Cagan, J. (1995). Synthesis of Optimal Non-orthogonal Routes. Advances in Design Automation, DE- 82(1), 431-438.
- [30] Chang, H. & Li T-Y. (1995). Assembly Maintainability Study with Motion Planning. IEEE International Conference on Robotics and Automation, 1012-1019.

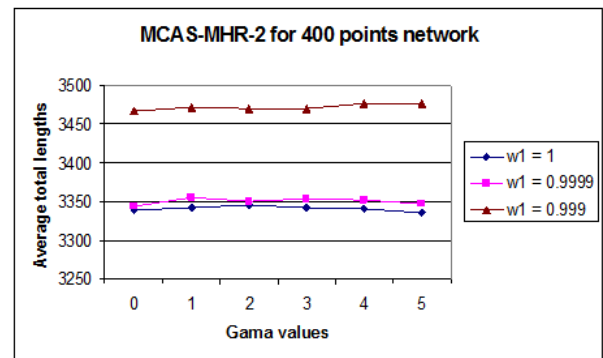
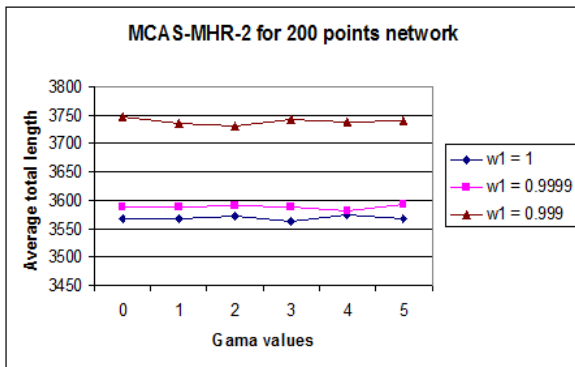


Fig. 11: Average total lengths of MCAS-MHR-2 of the 200 and 400 points networks against different values of  $\gamma$  values

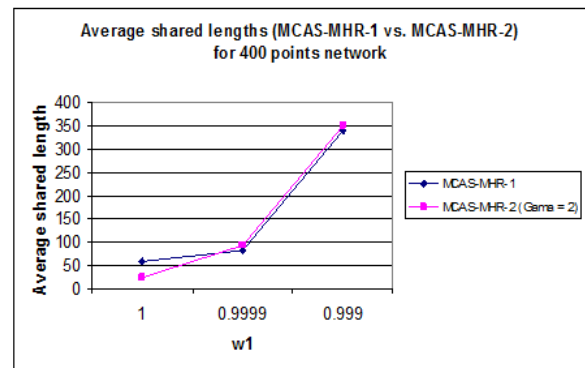
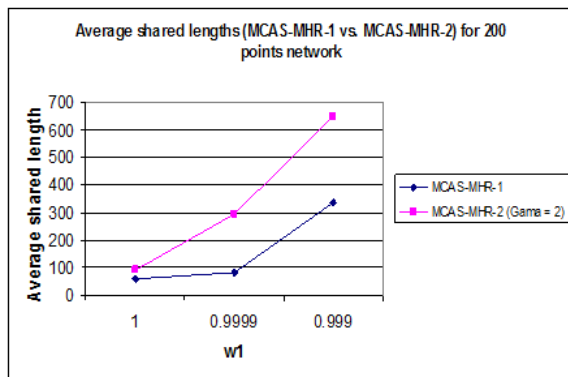


Fig. 12: Comparison of average shared lengths of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) of the 200 and 400 points networks against different values of weights

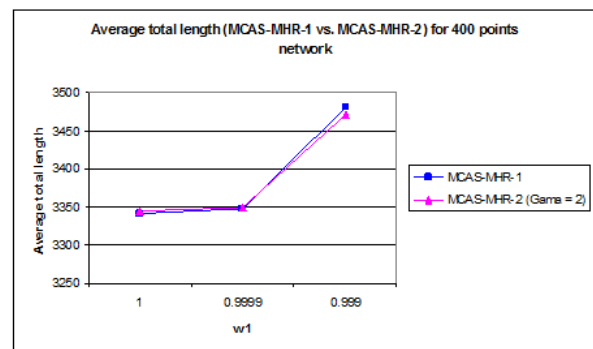
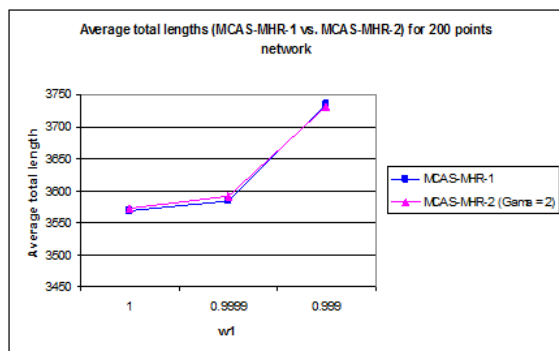


Fig. 13: Comparison of average total lengths of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) of the 200 and 400 points networks against different values of weights

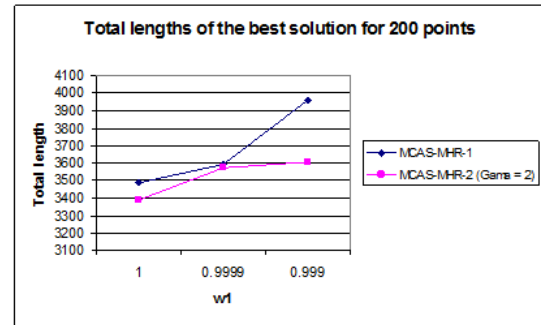
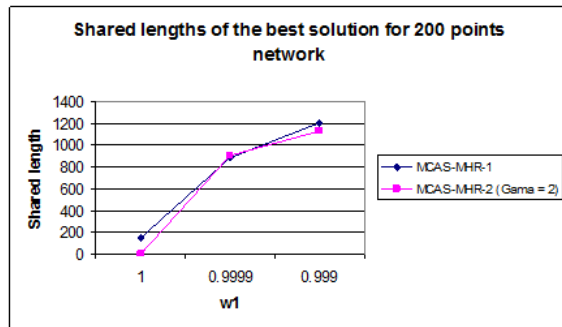


Fig. 14: Comparison of shared lengths and total lengths of the best solution of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) of the 200 points network against different values of weights

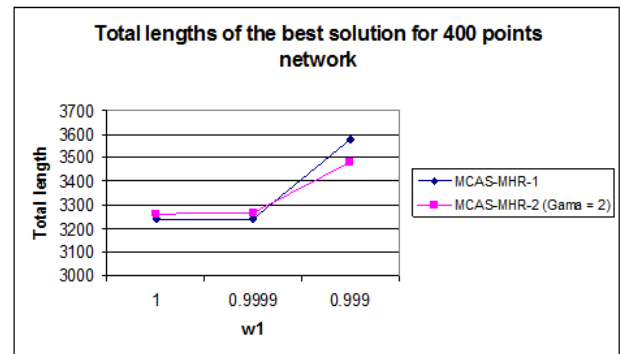
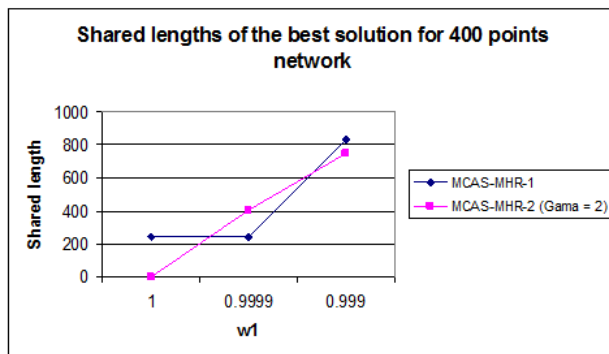


Fig. 15: Comparison of shared lengths and total lengths of the best solution of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) of the 400 points network against different values of weights

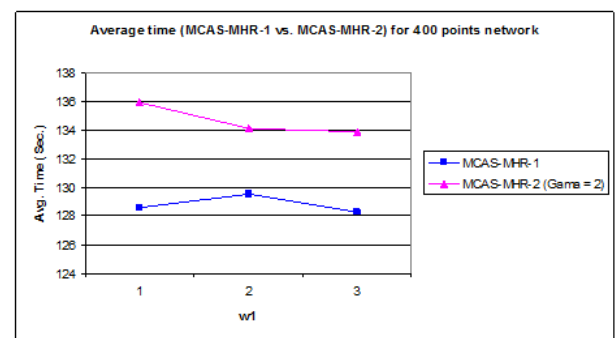
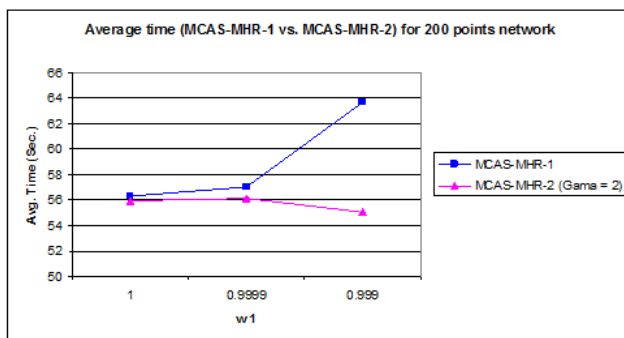


Fig. 16: Average time per trial of MCAS-MHR-1 and MCAS-MHR-2 ( $\gamma = 2$ ) for the 200 and 400 points networks against different values of weights