UNIVERSITY OF GLOUCESTERSHIRE

This is a peer-reviewed, post-print (final draft post-refereeing) version of the following published document and is licensed under All Rights Reserved license:

Nwaki, Michael Menonye, Zhang, Shujun ORCID logoORCID: https://orcid.org/0000-0001-5699-2676, Bechkoum, Kamal ORCID logoORCID: https://orcid.org/0000-0001-5857-2763 and Liewe, David (2016) ACO-based discrete and continuous optimisation algorithm for optimising multi-level truss topological design. In: 5th international conference on Bionics Engineering ICBE 2016, 21st to 24th June 2016, Ningbo Campus, University of Nottingham, Ningbo, China.

EPrint URI: https://eprints.glos.ac.uk/id/eprint/3475

Disclaimer

The University of Gloucestershire has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

The University of Gloucestershire makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

The University of Gloucestershire makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

The University of Gloucestershire accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.

An ACO-based discrete and continuous optimisation algorithm for optimising multi-level truss topological design

Michael Menonye Nwaki[1], Shujun Zhang[1*], Kamal Bechkoum[1], David Liewe[1]

[1] *School of Computing and Technology, University of Gloucestershire, the Park, Cheltenham, GL50 2RH, UK*

Abstract: Topological design truss problems are known to be difficult to solve to optimality, partly due to the inaccuracies of computational strategies to evaluate the impact of external forces acting on a specific edge (truss), which can have significant resultant effect on other trusses, through the individual vertices (node) and the magnitude of these resultant effects are difficult to accurately estimate due to the damping effect of the vertices (nodes). Besides, the largeness of the search space can be an issue, which can only be resolved with superior computational power and strategies. The complexities of these problems becomes exponentially larger when it involves multi-level hierarchies.

This paper proposes a variant of ant-colony metaheuristic algorithm (ACO). The proposed algorithm includes a local search metaheuristic. In this algorithm, the ACO will be used exploitatively to identify high performance area of the search field, through the continuous interactions between the search agents (ants) and then using the local search algorithms, such as hill-climber to localise the search solutions. The feasibility of the structure of each combinatorial solutions, i.e. the discrete options and the corresponding continuous design variables will be evaluated using the Grubler's criteria (degree of freedom), to determine the kinematic stability i.e. the statistical and dynamic balancing of a solution (structure).

From the computer simulation results, it has been shown that the proposed algorithm can be used simultaneously, in searching across multi-level hierarchies for solving trust topological design problems with both the discrete and continuous design parameters within a given set of constraints. It is effective and more efficient than the existing algorithms such as GA for the similar problems.

Keywords: ACO, optimisation algorithm multi-level-hierarchies, truss topological design

1 Introduction

Evolutionary computation is a subfield of computational intelligence, which deals with mechanism ability to exhibit or facilitate intelligent behaviours in a complex and changing environment.[1,2] In dealing with real world design and optimization problems, evolutionary computation has relied on nature-inspired biological behaviours of creatures and their behavioural patterns, which has been reengineered to solve complex design problems to optimality.

One of the biggest challenges of our time, is the ability to create highly responsive systems, compact in design, simple in operation, large-variations of options and configurations, easy to maintain and economical in every aspect (both materials and processes).These criteria puts a lots of design pressure on system designer and on production processes.

Corresponding authors: Shujun Zhang, E-mail: szhang@glos.ac.uk

In multi-level truss topological design, the relationship between components and sub-components at different levels of configurations of a whole-system will be the subject of this study. A typical example of this kind of design arrangement is the topological design schemata, where the number of discrete design options could grow exponentially, based on the inter-relationship between the main unit of an engineering system, its sub-units and the components of its sub-units.

Navigating through these levels of hierarchies, for each discrete configurations and sampling the corresponding continuous search spaces simultaneously can be a daunting challenges to most algorithms. The strength of the ant colony metaphor (ACM) base algorithms, developed by Dorigo et al has had wide approval in dealing with complex engineering and optimisation systems[2,3]. The ACM provides a means of implementing concurrent searches through the use of multi-agents cooperation and interactions during exploration.

The application of the ACM are demonstrated with two experiments: for each discrete configuration, the optimal solutions of the continuous search spaces are continuously evaluated to identify the design options that best matches the constraints and the feasibility of the structure/components. The implementation of ACM-variants used in this experiments with other variants of algorithms has proven to give better result than those used previously.

In the past, genetic-based algorithms and strategies have been used to explore the search spaces of truss optimisation problems, by generating series of suboptimal solutions, that gets refine over generations and iterations of run. One of the disadvantages of these form of optimal solution generations is the complicated mathematical models and heavy computational power required to obtain such results. The age of advancement in computer power technology has eased the consideration of such technique as viable ways of attaining feasible solutions for truss optimisation problem that are generally known to have very large phenotypic and genotypic search spaces. In addition, topological design problems are known to have very large search spaces.[4]These search spaces usually contain series of local multi-modal-minima, which are usually very attractive for the GA's and therefore could easily be trapped in these local optima, as shown in Figure 1.[5,6]

Furthermore, an unstructured problem domain i.e. one that has landscapes is deceptive with changing topologies during the evolutionary process. For these sorts of problems, the conventional method is not effective and efficient to find optimal solutions, when compared to using ACO.[7]

This paper will be focused on proposing and design of an ACO-based discrete and continuous optimisation algorithm for optimising multi-level truss topological design. Following the introduction, Section 2 deals with the formulation of objective function of the truss sizing optimization problem. Section 3 presents the use of ACO-variants for solving discrete and continuous design problems with hierarchy levels. Section 4 propose a method for applying hill climbing algorithms for fine tuning optimal solutions. Section 5 provides results of the computer-based soft experiments and comparative analysis with the other algorithms for a case study of seven-bar truss design. Section 6 draws a conclusion and provides the scope for further work on this algorithm.

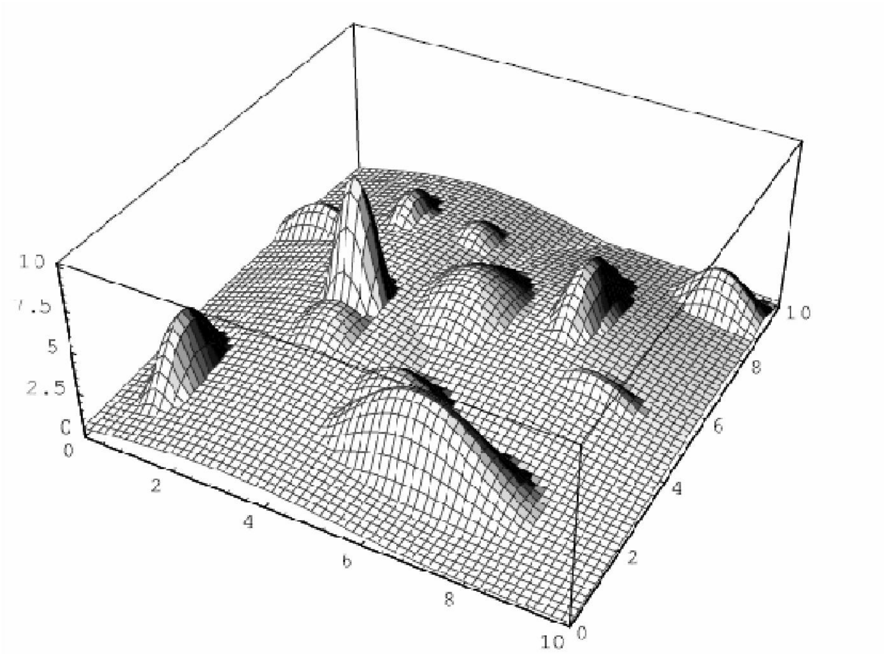

Figure 1 Three-dimensional view of the search space

2.0 Case for the use of ants-system algorithms for truss optimization

2.1 Problem discussion

Even though ants are assumed to be blind, yet they still communicate with each other, as well as with the environment[8]. A specific ant family known as the Lasius Niger uses the pheromones as a means of communicating certain findings to one another. Other behaviours observed with this ant's family, which makes them stand-out among other spices for solving such complicated design problems include the followings:

- *The randomness of search among each search agents*
  The search agents (individual ants within the family) takes responsibility of ensuring the survival of the whole colony. This is achieved through individuals and collective efforts of family members that randomly samples the entire landscapes in search of food-sources (division of labour) and coordinate their efforts to bring the food to the nest. This explains why the ant-system works well on dynamic systems and on graphs with changing topologies, which is of interest to design and engineering problems
- *The trust among each individual as well, as among other family members*
  Family member are welled trusted members, and therefore each agents respond effectively to the signals given by the other member. The idea of trust is very essential in software design and explains why access-restrictions are necessary to prevent intruders from corrupting the individual agents (software) - protocol design.

- *The degree of thoroughness of the search by the ant (exploration/exploitation)*
  It has been identified that the paradigm between exploration and exploitation of a search space for the use of specialist software for archiving these tasks.[9] In a highly diversified random method, an individual ant participates actively in exploring the entire search space. When the food-source is located, the attractiveness of the deposited pheromones along the trail ensures that other family member join forces in exploiting the solution. This characteristic is very important since one-software cannot explore and exploit

effectively the entire search spaces in polynomial time. As cooperation among the ants family is essential for the survival of the entire enclave, the cooperation among the different software is important for efficiency and reliability

- *The sense of judgments among individuals in abandoning a non-promising trail.*
  The simple fact that ants are able to follow the path with higher intensity of pheromone deposits is a natural and admirable sense of judgments, this explains why a non-promising area of search could easily be abandoned and all resources from the trail are re-committed to high performance regions within the search space.

- *Balance between cost and solutions*
  The economic consequence of a solution is another area, where the ants demonstrate some degree of superior judgments among lots of software's. When the distance from the nest to the food source is very long, the deposited pheromone on the trail quickly evaporates between successive tours. This action could be examined from two perspectives:
  - There is an assumption that when the cost of finding a good solution (could be computational cost or etc.) exceeds the reward, the solution may not be worth exploiting; hence the pheromones on the trail evaporates and consequently no other ants will explore that path.

  - In contrast, therefore, if every optimal solution is abandoned because of the cost of bringing food to the nest, it comes impossible to find good solution. So there must be a way of balancing cost and the quality of the solution. Otherwise modifying the problem space will be the only-way out.

2.2 Design complexity of the problem

Describing a system that comprises of both the discrete and continuous design variables is somewhat difficult in nature. The difficulties are mainly associated with the discreteness of each configuration option, which can only be described by a large continuous search spaces. When the discrete level increases, an exponential increase in the continuous search spaces is expected, sampling these continuous search spaces concurrently for each discrete configuration can be somewhat be challenging for any computational machines. ACO tends to have a balance that ensure that each agents simultaneously computes each of the continuous search spaces of their respective discrete configurations and relates the data to a centralised repository.

3.0 Topological truss design problems

3.1 Introduction

In order to understand the complexities of topological design problems, it is imperative to examine the complexity posed using ant colony metaphor to find optimal design solutions for a simple truss optimization problem as shown in Figure 2. The aim is to solve topological design problems as if it were the travelling salesman problem (TSP).

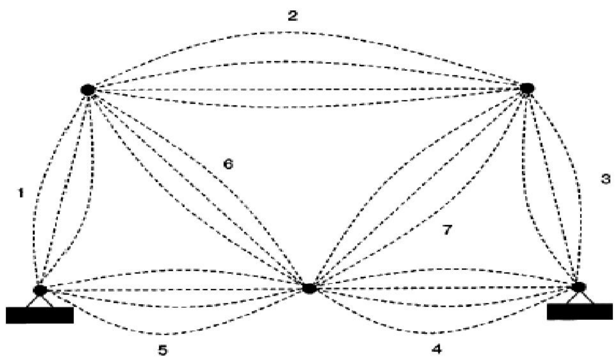

Figure 2 Pheromones paths laid by ants as they locate each food source

The motivation for applying ant-colony metaphor for solving topological design problems is based on the recorded success of the ant-colony metaphors for solving discrete or continuous design problems for the obvious reasons stated above.[3,8]

Figure 2 shows a typical 7-bar trusses that consists of a number of edges, represented by a numerical values between 1 and 7. The inter-connectivity device between two edges, m and n are known as the vertexes (5 vertexes in total). As ants (s) wonders in search of food-source from their nest, they make probabilistic decisions, based on the numbers of options available to them, with regards to directions and locations that deemed promising where they might find some food source. As they journey in search of food, they communicate with other ants through the deposition of stigmergy (pheromones) on the path.

As shown in Figure 2, during this iterative process, pheromones (dotted-lines) is a representation of series of constructed solutions by the ants as they move-from the present edge m to the next edge n.

Pheromones are chemical substance with the capability of evaporation, depending on the intensity of the follow-up ants. In the case of truss optimisation problem, a solution can only be said-to-be feasible if it can withstand the forces at play and gives the user a sense of security, that is that such a product will not crumble under fatigue (Grubler's criterion)

3.2 Mathematical modelling of an truss optimisation problem

Similar to the TSP; the objective of this paper is to find the optimum topological design (TOD) with the objective of minimizing materials (volumes) cost. The stress in materials under tension and displacement due to the acting forces can be used to check if the design is strong or not. The objective function of minimising materials volume can be expressed as:

$$\text{Min}\, W = \rho \sum_{m=1}^{7} A_{nm} \tag{1}$$

Where: W is weight;

ρ is material density;

A_{nm} is area of the current truss m in relation to the next truss n;

3.3 The seven-bar truss problem

The seven-bar truss-line diagram, as shown in Figure 3, is a transposition of the 7-bar pheromones diagram in Figure 2. The seven-bar truss line diagram consists of two reaction forces, which are rigidly bolted to the earth and well supported at point A and B. At the top of the structure, are two equal and opposite horizontal forces $\mathbf{P}_1$ and $\mathbf{P}_2$. A vertical compressive downward force ($\mathbf{P}_3$) acts on the base of the construction, resulting in a harmonic deflection of the edges concerned.

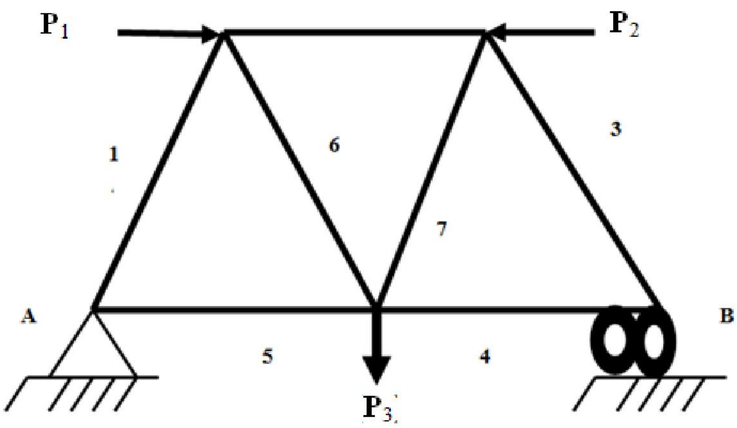

Figure 3 Free-body line diagram for Seven-bar truss

The resultant effect of this vertical compressive force ($\mathbf{P}_3$) is a maximum deflection at the middle-length of the base truss A-B.

3.4 Formulation of constraints:

Due to the action of the exerted forces on the body of the structure, the structure mechanism may be subjected to deflective action (compressive); the life span of such structure will depend on the intensity of stresses caused by these forces.

As a check on each edge, the maximum stress on a specific edge (m) will depend on the formula in Equation (2).

$$\sigma_m = \frac{F_m}{A_m} \leq \sigma_{all} \qquad (2)$$

Where: σ_m is normal stress on edge m;

F_m is normal force on edge e;

A_m is the area of truss m;

σ_{all} is the maximum allowable stress on member m (for m =1 to 7).

Since the objective of this section is to deal with the structural optimization of the seven-bar structure, which in principles will address the following under mentioned items, such as: topological optimization, which refers to the variations of the element node (vertexes), being the connectivity between the current edge (m) and the next edge (n).The shape optimization refers to the movement of the node that can affect the shape of the structure without compromising its topology and node-connectivity. The size optimization will deal with the variations in the cross-sectional areas of edges (m, n).

The slenderness ratios ($\lambda_{m,n}$) and displacements ($\delta_{m,n}$) are two very important factors that can be used to determine the feasibility of the generated ants solutions, hence it is imperative that these factors are taken into considerations.

Subject to:

$$g_m = \frac{\sigma_m}{\sigma_{all}} - 1 \leq 0 : for\ m = 1, \dots \dots . . n \quad (3)$$

$$g_{\lambda m} = \frac{\lambda_m}{\lambda_{all}} - 1 \leq 0 : for\ m = 1, \dots \dots . . n \quad (4)$$

$$g_{\delta m} = \frac{\delta_{m,n}}{\delta_{all}} - 1 \leq 0 : for\ m = 1, \dots \dots . . n \quad (5)$$

Where: g_m is stress value per unit edge;

$g_{\lambda m}$ is slenderness ratio;

$\delta_{\delta m}$ is displacement due to forces at play;

F_{c_i} is compression forces;

σ_m is computed stress per each edge;

σ_{all_i} is the maximum allowable stress on each edge.

λ_m is the slenderness ratio.

3.5 Conditions for efficient ant systems:

Bilchev and Parmee identified that in order to have an efficient ants- algorithms, the location of the nest is an important factor.[9] The nest is a symbolic colony (home) which inhabits the ant's family. The nest should be located in an area that is closer to a constant-food source supply. This will prevent the ants from having to explore a large unconstraint search spaces in search of food. Therefore, the location of the nest will always be an important factor in solving design problems within polynomial time. If we consider the simple cantilever problem (32 x 20 cells), which is essentially a simple design problem in today's design environment but the search space could be as large as 2^{640} design choices. Kim and de Weck argued that of these 2^{640} design choices, only less than 5% is physically meaningful.[4] In reality, it is almost impossible to explore the entire design choices within polynomial time. Kim and de Weck proposed the idea of using a dedicated software (ants) to locate the fractions (5%) with physical meaning from the vast hypothetical design choices. There have been suggestions that a "niching-algorithm" could be employed to scan the search spaces in order to identify high performance regions where the nest could be located.

Another important factor is the identification of the stopping criteria for the ants-algorithm, Bilchev et al proposed a variant of ant-colony algorithm for searching continuous design spaces.[9] It was proposed that if after a successive number of iterations, a specific trail continuous to generate values that is below a stated threshold, such trail will be evaporated and the resources redistributed to a more promising trail (path). Dorigo et al holds on to the analogy that ACO algorithm could not work well without evaporation of solutions that are sub-optimal.[3] The evaporation of a solution can only occur when a specific path does not attract more ants (stagnation). The evaporation is a condition resulting from fewer ants moving towards a trail. It was noted that excessive numbers of ants in a trail could lead to less efficient computational system. This situation is noticeable when the result obtained after a number of iterations is not significantly improved. In TSP problem, this paradigm will introduce the need for a trade-off between the numbers of ants and the number of vertexes Dorigo.[10]

3.6 Construction of solutions (path)

As the ants transverse through the trusses, stochastic decisions are made by each ant base on the number of choices at their disposal regarding the direction of the next (n) move (m,n). An Ant (k)has the option of moving from a current edge (m) to the next edge (n) base on the state transition rule of the probability stated in Equation (6) below:

$$T_{m,n}{}^{k} = \frac{[\tau_{mn}]^{\alpha}[\eta_{mn}]^{\beta}}{\sum_{t\epsilon N_m^K}[\tau_{ml}]^{\alpha}[\eta_{ml}]^{\beta}} \qquad if\ n \epsilon N_m^k \qquad (6)$$

Where:

$T_{m,n}{}^{k}$ is transition probability of the k-th design variable, transiting from (m) to(n);
τ is quantity of pheromone deposit between the edges m and n ;
$\eta = {}^{1}/_{\delta}$ is the inverse of the distance $\delta(m, n)$;
N_m^k is the feasible environment of ant k and edge m;
α is relative factor of importance of pheromone trail;
β is relative factor of the heuristic value.

Note that this conditions exist to be true of the relationship between the values of the alpha (α) and the beta (β) respectively. When the value of alpha (α) tends towards zero:

$$\alpha\ (alpha) \ \dashrightarrow\ 0\ \{pheromones\ deposit\ are\ negligibe, ant\ tends\ to\ take\ alternative\ quick\ route\ \}$$

Consequently when the heuristic value ofβ (beta) tends towards zero,

$$\beta\ (beta) \dashrightarrow\ 0\ \{heuristic\ value\ are\ disregarded, causing\ risk\ of\ algorithm\ stagnation, similar\ to\ GA\ being\ trapped\ in\ local\ optimum\}$$

This two conditions must be addressed to avoid them happening. A similar trends that occurs using the genetic algorithms (GA) to solve truss optimization problem; a situation that occurs when GA is trapped into a local optimum and needs the re-introduction of some strengths, in form of mutation to energise the algorithms to move out of the trapped position.[11]

At initialisation stage, a matrix P is generated, representing the numbers of design variables encoded in the problem domain. This matrix (P) is initialised with a value of zero for each of the design variable. These values is used to indicate the strength of the pheromone (τ_0) upon each of the edges. The assumption is that pheromone level is very negligible or almost none in existence at the start of the iteration.

As the artificial ants (k) traverse along the path (edges) from the start-node, the state-transition rule is applied based on the Equation (6). After an instant tour has taken place (transition) each ants (k) automatically updates the pheromones level (updates Matrix P) from the initial pheromones (τ_0) level to a new updated value using Equation (7), which is popularly known as the local updates:

$$\tau_m = (1-\rho)\tau_m + \rho\tau_o \quad \ldots \quad (7)$$

Where:

τ_m is quantity of pheromone deposit at edges m

τ_o is value of pheromones deposited at initialization stage

ρ is the rate at which pheromones concentration vaporises on edge (m) for the

ρ Values that lies between $0 \leq \rho \leq 1$

The reasons for the local updates is to enhance diversities in the solutions constructed by the ants and ensure that edges already visited becomes less attractive for the follow-up ants (k), thereby enhancing the exploration of the ants to other edges (n) that has never been explored.

The steps above is repeated using Equations (6) and (7) respectively. After the successful completion of these transition tours, a global pheromones update is initiated to enhance and update those tours that represents the best solutions using the global transition update Equation (8):

$$\tau_m = (1-\rho)\tau_m + \rho\Delta\tau_m \qquad \forall(m,n) \in T^{bs} \quad \ldots \quad (8)$$

Where:

τ_m is the quantity of pheromone deposit at edges m;

$\Delta\tau_m$ is the minimum value of pheromones deposited at initialization stage;

ρ is the rate at which pheromones concentration vaporises on edge (m):

$for\ values\ that\ lies\ in\ between$: $0 \leq \rho \leq 1$.

5.0 Evaluation of constructed/update of solution matrix

As ants transverse through the edges and nodes, each ant constructs a solution that should be instantly updated locally These local update changes is very important for the search paradigm of the ant colony algorithm as they serve multi-purposes. The local change updates the row values of the matrix and also serves the purpose of demotivating more and more ants from following the already visited truss, as the pheromone level decreases with more visits. Consequently, these action ensures that unvisited truss edges are becoming more attractive for the up-coming ants, which in-turn increases the exploration rate of this algorithm (exploration versus exploitation). The exploitation of this algorithm is archived by employing a local hill-climber algorithm.

When all the ants have constructed their respective solutions, i.e. an assumption that ants have ended their tours; only those solutions identified as best candidates solutions are globally updated, using the global update Equation (8).

After the global update of the best solutions, the kinematic stability of the solutions are evaluated alongside the infinite analysis method. A solution can be accepted or rejected through the application of the Grubler's criteria or internal stability Equation (9)

$$F(m,n) = 3(N-1) - 2J_1 - 2J_2) \qquad (9)$$

Where: N is number of nodes;

J_1 is lower level reaction forces;
J_2 is higher hinges/reaction forces;
m is current energy state;
n is the next energy state;
F is Degree of freedom.

Applying the Gruebler's equation, if the degree of freedom (D.O.F) is zero, it implies that there exists a perfect balancing of the forces, that means that the solution is a structure, however if the obtained value (D.O.F) is greater than zero, it means that we have a mechanism, an indication that our solution may not be able to withstand the forces that will be at play. The final solution will be those solutions in the matrixes whose value is zero, from applying the Gruebler's equation.

6.0 Numerical results and analysis

In this section, the numerical results are presented for the comparative analysis. The Standard Genetic Algorithm is selected for comparative analysis with the proposed new ACO method. The 7-bar truss in Figure 2 is used for the simulation. Table 1 shows the constraints of the search simulations.

Table 1 Search constraints

Item name	**Symbol**	**Value range**	**Purpose**
Alpha	α	1.0	Enhance exploitation
Beta	β	0 - 0.5	Heuristics information
Diameters (D)	D1……D7	20mm < D < 55mm	Prevent over dimensioning
E-Module Steel	E-module	200,000 N / mm^2	Standard value
Material		Carbon steel 40-60	
Allowable Stress	σ_{all}	$\sigma_{all} \leq 89\ N/mm^2$	Prevent collapse
Deflection	δ	1.0 mm $\leq \delta \leq$ 1.25 mm	
Strength of pheromones		1.8d	D = diameter
Reaction forces	P_1, P_2, P_3	$P_1 = P_2 = 0$, $P_3 = 250$ KN	
Initial ant size		10000	
Density		7830 kg / m^3	
GA standard deviation		0.01595	

6.1 Reasons for Standard Genetic Algorithm (GA) for comparison

The Standard Genetic algorithm (GA) has been applied successfully in solving many complex problems, including some design problems. The genetic algorithm (GA) which is inspired by Darwin's theory of evolution is rooted in the principles of survival of the fittest in nature (environment).

Genetic algorithms are based on the evolutionary processes that occurs in nature, involving different species and their ability to survive over time (generations) in their environment (nature). The GA uses a population of individuals to represent the species, that will reproduces offspring's over periods of generations and whose ability to survive into the future (over many generations) will largely depends on their fitness and ability to adapt to the changing nature of their environment.

In GA's the population of individuals (the off-spring) which are produced over each generation may or may not look like their predecessors (parents), depending on some factors. These offspring's are produced as a result of undergoing through the process of evolution, known as genetic operators, consisting of crossover, selection and mutation respectively.

When the genetic operators are applied to a population of genes, represented by bit strings in the simplest form, each gene receives a numerical evaluation that connotes her fitness to the environment (the search space). Each point in the search space represents one possible solution and each possible solution can be identified by its fitness value. The solutions with the highest fitness value are preserved into the next generation. These characteristics (population of solutions) makes GA's one of the most attractive algorithms for solving problems of larger dimensional variables. AG has been applied in optimisation design.[12-15] So in this paper GA is selected for comparative analysis with the proposed new ACO method.

GA was run, over 100 generations, producing varying areas for the 7-bar trusses. Figure 4 shows the curve of fitness against the generations. The diameters of the 7 bars generated using GA method are listed Table 2.

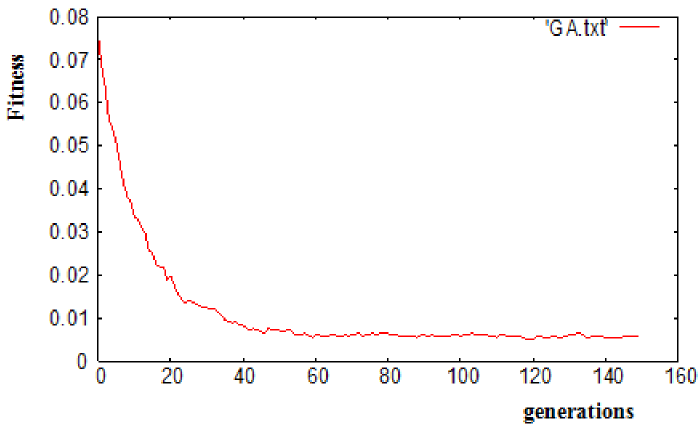

Figure 4 The relation between fitness and generations

Table 2 The simulation results using GA (Unit = mm)

	D1	D2	D3	D4	D5	D6	D7
1	45	41	36	51	41	37	33
2	50	36	32	41	32	31	45
3	53	44	43	36	38	40	43
4	43	42	36	44	42	43	44
5	40	44	41	42	43	39	37
6	34	41	44	43	42	43	43
7	43	50	41	43	45	31	41
8	48	43	43	43	41	43	44
9	46	47	41	36	43	44	42
10	40	42	44	43	43	43	41

From the result above, an average deflection of 1.25mm was obtained; the impact of this value will affect the durability and the life span of the structure and consequently the factor of safety. However, with 1.25 mm deflection on the structure, this represent an infinitesimal influence on the factor of safety of the structure in the short-term, as this could be within the limit of constraint. However, the overall influence can be used to estimate the life span of the structure (product), in the form of a guarantee.

In Table 2, the highlighted rows 10 and 8 represents the minimum and maximum row values for the diameters generated by the GA. The aim of selecting these two extreme values is to be able to calculate the corresponding areas and stress on the construction. These quantities will forms the basis for comparative analysis.

The minimum area for minimum GA solution is 9839 mm^2 with stress of $25.4\ N/mm^2$ and the maximum area for GA solution $= 10982\ mm^2$ with a stress of $22.7\ N/mm^2$.

From the simulation results, it can be seen that the better and more refined results begins to emerge as the GA approaches over 100 generations. Slow convergence rate was observed, which raises the suspicion whether, more exploration/exploitation were taking place. The rate of convergence can be subject to questioning, but since this research is not about GA, such debate would not be appropriate in this forum. This is a standard GA.

From the result above, an average deflection of 1.25mm was obtained with a maximum stress on the material or construction of $25.4N/mm^2$. The impact of this value will affect the durability and the life span of the structure and consequently the factor of safety. However, with 1.25 mm deflection on the structure, this represent an infinitesimal influence on the factor of safety of the structure in the short-term, as this could be within the limit of constraint. However, the overall influence can be used to estimate the life span of the structure (product), in the form of a guarantee.

6.2 Numerical results using the proposed ACO-based discrete and continuous optimisation algorithm

The proposed ACO-based discrete and continuous optimization algorithm has been applied for optimising the same 7-bar truss as used in GA method.

Table 3 lists the numerical results of the diameters of all seven bars using the proposed ACO-based discrete and continuous optimisation algorithm.
In Table 3, the highlighted rows 7 and 12 represents the minimum and maximum row values for the diameters generated by the GA. The minimum area for ACO solution = $4593mm^2$ with a minimum stress of $36.7\ N/mm^2$ and the maximum area for ACO solution = $6811\ mm^2$ with maximum stress of $54.4\ N/mm^2$.

Comparing the simulation results of Table 2 (GA method) and Table 3 (the proposed ACO based algorithm), it can be noticed that the bar diameters obtained using the proposed ACO based method is much less than the those obtained from the GA. That means more materials can be saved. This can only be best described as the over-dimensioning of the structure. The over-dimensioning is usually caused by computational inefficiencies, which gives rise to material waste and hence financial lost.

These values could be attributed to the elimination criteria used by Grubler for decisions regarding the kinematic stability of the structure with respects to the forces at play, especially on the trusses that are directly affected.

Table 3 Simulation results using the proposed ACO (unit = mm)

	D1	D2	D3	D4	D5	D6	D7
1	34	39	33	34	23	40	23
2	41	28	26	31	28	40	28
3	38	41	30	37	25	33	38
4	23	40	35	28	38	41	39
5	32	29	33	28	33	26	28
6	31	40	37	29	41	31	40
7	26	33	25	28	24	27	37
8	37	25	33	24	29	32	41
9	38	36	32	25	38	30	40
10	38	29	26	23	23	39	32
11	38	40	24	29	41	30	24
12	41	35	38	35	39	30	26
13	29	36	41	29	38	30	39
14	40	32	30	32	24	26	29
15	37	27	30	34	33	40	40
16	33	34	30	32	33	32	36
17	36	24	35	35	41	40	30
18	41	41	25	40	26	29	30

In Table 3, the highlighted rows 7 and 12 represents the minimum and maximum row values for the diameters generated by the GA. The minimum area for ACO solution = $4593mm^2$ with a minimum stress of 36.7 N/mm^2 and the maximum area for ACO solution = 6811 mm^2 with maximum stress of 54.4 N/mm^2.

Comparing the simulation results of Table 2 (GA method) and Table 3 (the proposed ACO based algorithm), it can be noticed that the bar diameters obtained using the proposed ACO based method is about 20% to 30% less than the those obtained from the GA. That means more materials can be saved. This can only be best described as the over-dimensioning of the structure. The over-dimensioning is usually caused by computational inefficiencies, which gives rise to material waste and hence financial lost.

These values could be attributed to the elimination criteria used by Grubler for decisions regarding the kinematic stability of the structure with respects to the forces at play, especially on the trusses that are directly affected.

7 Conclusion

In this paper, a variant of ant-colony metaheuristic algorithm (ACO) has been proposed. The proposed algorithm includes a local search metaheuristic. In this algorithm, the ACO is used exploitatively to identify high performance area of the search field, through the continuous interactions between the search agents (ants) and then using the local search algorithms, such as hill-climber to localise the search solutions.

A 7-bar truss was selected for the comparative analysis of the optimised designs between the proposed ACO based method and GA. From the computer simulation results, it has been shown that the proposed algorithm can be used simultaneously, in searching across multi-level hierarchies for solving trust topological design problems with both the discrete and continuous design parameters within a given set of constraints. The bar diameters obtained using the proposed ACO based method is about 20% to 30% less than the those obtained from the GA.

It should be pointed out that in this paper only a 7-bar truss is used for analysis. Further work should be carried out to determine whether the effectiveness of solving multi-hierarchy problem, consisting of both then discrete and continuous design variables for large industrial appliances, where the discrete value is used to determine which of the continuous search spaces are active or non-active and the effect or contribution of the non-active genes in enhancing population diversities.

References

1. Bezdek, J.C, (1994) What is computational intelligence? in, *Computational Intelligence: Imitating Life*, (ed. Zurada, J.M. et al.) 1-12 (IEEE Press, New York, 1994).
2. Dorigo, M., Maniezzo, V. and Colorni, A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics*. 26, 1-13(1996).
3. Dorigo, M and Gambardella, L. Ant colonies for the travelling sales man problem. *IEEE Transactions on Evolutionary Computing*, 1, 53-66 (1997).
4. Kim, Y. and Weck, O. Variable chromosome length, an genetic algorithm for structural topology design optimization, *The Proceedings of 45th AIAA/ASME*, Florida, USA, 1491-1497 (2004).
5. Rao, S. S. and Xiong, Y. A hybrid genetic algorithm for mixed-discrete design optimization. *Journal of Mechanical Design, Transactions of the ASME*. 6, 1100-1112(2005).
6. Kicinger, R., Arciszewski, T. and De Jong, K.A. Evolutionary design of steel structures in tall buildings. *Journal of Computing in Civil Engineering*. 19, 223-238 (2005).
7. Raich, A.M. and Ghaboussi, J. Implicit representation in Genetic Algorithm. *Evolutionary Computation*. 5, 277-302(1997).
8. Dorigo, M., Bonabeau, E. and Theraulaz, G. Ant algorithm and stigmergy IRIDIA, Universite Libre de Bruxelles (2000).
9. Bilchev, G. and Parmee, I. The ant colony metaphor for searching continuous design spaces. *Evolutionary Computing of Lecture Notes in Computer Science*.993, 25-39 (2005)
10. Dorigo, M. Optimization, learning and natural algorithms. Ph.D. Thesis, Politecnico di Milano, Italy, (1992).
11. Deb, K., Pratap, A., Agarwal, S. and T. Meyarivan T. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation.* VOL. 6, 182-197 (2002).
12. Croce, E. S., Ferreira, E. G. and Lemonge, A. C, A genetic algorithm for structural optimisation of steel truss roof. Available at: http://www.lncc.br/~hcbm/artigo(567).pdf. (Accessed 20th March 2016).
13. Aminifar, F., Aminifar, F. and Nazarpour, D. Optimal design of truss structures via an augmented genetic algorithm. *Turkish Journal of Engineering & Environmental*

Sciences. 37, 56 – 68 (2013).
14. Wu, S. and Chow, P. Steady-State Genetic Algorithms for Discrete Optimization of Trusses. *Computers and Structures*. 56, 979–991 (1995).
15. Yang, J. and Soh, C. K. Structural Optimization by Genetic Algorithms with Tournament Selection, *Journal of Computing in Civil Engineering*. 11, 195–200 (1997).