Eng.& Tech. Journal. Vol.27,No.16,2009

Approximate Solution of The Linear Programming Problems By Ant System Optimization

Faiz Faig Showkat* , Saad Mohsen Salman & Haithem Saleem Dawood**

Received on:30/3/2009
Accepted on:2/7/2009

Abstract

In this paper we use the ant system optimization metaheuristic to find approximate solution to the linear programming problems, we also use the duality theory to estimate how good is this approximate solution and check if it is optimal, the advantageous and disadvantageous of the suggested method also discussed focusing on the parallel computation and real time optimization, it's worth to mention here that the suggested method doesn't require any artificial variables the slack and surplus variables are enough, a test example is given at the end to show how the method works.

Keywords: Linear Programming Solution ;Ant System Optimization ;Linear Programming Problems

الحل التقريبي لمسائل البرمجة الخطية باستخدام نظام النمل للامثليه

الخلاصة

استخدمنا في هذا البحث نظام النمل للامثليه لإيجاد حلول تقريبية لمسائل البرمجة الخطيــة كما استخدمنا نظريه الثنائية لمعرفه ما إذا كان الحل التقريبي المقترح هو الحل المثالي أيضا وان لم يكن فكم هو قريب من الحل المثالي ، تم مناقشه الفوائد المتوقعة لهذا الأسلوب في حل مســائل البرمجة الخطية وتحديد نقاط الضعف كذلك وتم التركيز على الفوائد المتوقعة في جانب إجــراء الحسابات المتوازية والامثليه الانيه ، من الجدير بالذكر هنا أيضا إن الأسلوب المقترح لايحتــاج إلى أضافه متغيرات مصطنعه فقد كانت المتغيرات الفائضة كافيه لإيجاد الحــل التقريبــي ممــا يعطي الأسلوب المقترح ميزه اضافيه بالنسبة لعدد المتغيرات التي يجب التعامل معهــا لإيجــاد الحل.

Introduction

The linear programming (called LP for brief) is an optimization method which can be used to find solutions to problems where the objective function and constraints are linear functions of the decision variables, the intersection of constraints result in a polyhedron which represents the region that contain all feasible solutions, the constraints equation in the LP problems may be in the form of equalities or inequalities, and the inequalities can be changed to equalities by using slack or surplus variables, one referred to Rao [1], Philips[2], for good start and introduction to the linear programming problems.

The LP type of optimization problems were first recognized in the

* College of science, Al-Nahrain University/Baghdad
** Applied Science Department, University of Technology/ Baghdad

https://doi.org/10.30684/etj.27.16.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

30's by economists while developing methods for the optimal allocations of resources, the main progress came from George B. Dantizag when he introduced the simplex method for solving the LP problems i.e. finding the optimum values of the decision variables and hence the optimum value of the objective function . In real world problem simplex method was the first practical useful approach for solving LP problems and after it is introduced the number of applications of LP becomes large ranging from transportation, assignment, production planning, transshipment…etc

Although the simplex method is the most popular and practical method for solving LP problems but there are other methods introduced by researchers like logarithmic barrier method, affine scaling, and interior methods, one referred to Vanderberi[7] for more information.

There are many forms for representing the LP problems we will use the following standard (scalar) form

$$\textbf{Min } f(x_1,x_2,\dots,x_n)=\sum_{i=1}^{n} C_i x_i \quad ..(1)$$

$$\textbf{Subject to } \sum_{i=1}^{n} a_{ij} x_i = b_j \textbf{ for } j=1,2,\dots,m\dots\dots \quad (2)$$

$$x_i \geq 0 \textbf{ for } i=1,2,\dots,n\dots\dots..(3)$$

Where (n) represents number of decision variables and (m) represents number of constraints, $\mathbf{C_i}$, $\mathbf{a_{ij}}$, and $\mathbf{b_j}$ are known constants, now if (n=m) then there exits one solution (if any) and can be found by solving the system of linear equations (2) simultaneously by any known linear algebra method like gauss elimination, gauss – siedel, … etc , see Bernard [9] for solving system of linear equations , this case (i.e. n=m) is of no interest in optimization theory . The second case raise when m>n it means there are (m-n) redundant constraints which can be removed out and go back again to the case of (m=n). The last and most important case is when n>m, this case is tackled by LP to find the solution in which the objective function is optimum.

In LP problem any solution which satisfies equation (2) and (3) above is called a feasible solution, but the one in which (n-m) variables are set to zero called basic feasible solution, the set of variables not equal to zero (i.e. which obtain the basic feasible solution) are called basic variables, while those set of variables equal to zero called non basic variables, when it happen that the basic feasible solution make objective function optimum it is called optimum basic solution. The basic feasible solution of the LP problem has some characteristics, from these characteristics it is always found at an extreme point of the polyhedron generated by the intersection of constraints, and also each extreme point is a basic solution, these characteristics and their proof can be found in Rao[1], Gauss[3].

One way to find the optimum solution of a given LP problem is to generate all the basic solutions and pick the one which is feasible and correspond to the optimal value of the objective function, this is done because the optimal solution (if one exist) always occurs at an extreme point of the polyhedron (i.e. the vertex) Rao[1], Gauss[3], so if there are (m) equality constraints in (n) variables with n>m , a basic solution can be obtained by setting any of the (n-m) variables equals to zero and solve for the remaining variables, the total number of basic feasible solution obtained by this way can be found using (n! /(n-m)! m!) , this procedure also known as algebraic method, its main disadvantageous is the huge number of basic solutions that we must inspect in order to find the optimal solution which must satisfy the nonnegative constraints and the other ($\mathbf{a_{ij}x_i=b_j}$) constraints .The simplex method provides us with a procedure to move from one vertex to another vertex with a better objective function value.

In this paper we will present another method to move from one vertex to another by using the ant system optimization metaheuristic , the search for the optimum basic feasible solution by this method will be random and probabilistic in nature, so no guarantee will be given that the optimum solution will be found and this leads us to say the solution given by ant system will be approximate solution but with the help of duality theory the suggested (approximate) solution can be evaluated and checked if it is optimum, more about advantageous and disadvantageous of the suggested method will be discussed later. The literature survey shows that some attempts were carried out to find optimum feasible solution using random search, most highlighted one is the shadow vertex method Kelner[6]and genetic algorithm approach Bayoumi [10].

Metaheuristic:

Algorithms which search for optimal solution using procedures that are probabilistic in nature listed under the topic of approximate solution Dorgio [4], Merkle [5], these algorithms generally classified into two main types, *local search* algorithms and *constructive search algorithms*, the local search type repeatedly try to improve the current solution by making movement to neighborhood solutions and if the neighborhood solution is better than the current solution the algorithm replaces the current one by the new one, but if no better neighborhood solution found the search stops. The second type (constructive algorithm) generates solution from scratch by adding solution components step by step.

One of the main disadvantages of above iterative improvement algorithms is that they may become stuck in poor quality local optima , so efforts directed to build more general purpose techniques for guiding the constructive or local search (heuristic) algorithms, these technique are often called metaheuristic and they consist of concepts that can be used to define heuristic methods, there are other definitions available to metaheuristic, like " general algorithmic frame work which can be applied to different (combinatorial) optimization problems with relatively few modifications", in recent researches metaheuristic are widely used and organized as the most promising approach for attacking hard combinational optimization problems example of metaheuristic algorithms are simulated annealing, tabu search, iterated local search, variable neighborhood search algorithms, greedy randomized adaptive search procedures, evolutionary algorithms, genetic algorithms and the recent metaheuristic found by Diorgio [4] called the ant system optimization metaheuristic (and it's variant models) abbreviated as ASO

Ants in Real world:

Ant system optimization was inspired from the real ant's behavior, so let us simplify the understanding of ant system optimization by taking a fast and brief look on the behavior of real ants and how they search for food and communicate between each other.

A very interesting aspect of the behavior of several ants is their ability to find shortest paths between the ant's nest and the food sources , this is done by the help of deposit of some ants to a chemical material called pheromone, so if there is no pheromone trails, ants move essentially at random, but in the presence of pheromone they have a tendency to follow the trail and experiments show that ants probabistically prefer paths that are marked by high pheromone concentration, the stronger the pheromone trail in a path then this path will have the higher desirability and because ants follow that path they, will in turn, deposit more pheromone on the path and they will reinforce the paths , this mechanism allows the ants to discover the shortest path, this shortest path get another enforcement by noting that the pheromone evaporates after some time, in this way the less promising paths progressively loss pheromone because less and less ants will use these paths, for more information for the real ants behavior and the experiments done about the ants one refer to Diorgio [4] .

Artificial Ants:

Researchers try to simulate the behavior of real ants by introducing the artificial ants, which are a simple computational agent that tries to build feasible solution to the problem being tackled by exploiting the available pheromone trails and heuristic information, the main characteristics of artificial ants are Diorgio [4] :

A. It search for minimum cost feasible solution for the problem being solved (i.e. shortest path)

B. It has a memory storing information about the path followed until the end, this stored information can be used to

I. Build feasible solution.
II. Evaluate the generated solution.
III. Retrace back the path the ant followed.

C. It has initial state that usually corresponds to a unitary sequence, and one or more termination condition.

D. It starts with the initial state and moves towards feasible states, building its associated solution incrementally.

E. The movement of the artificial ant is made by applying a transition rule, which is a function of locally available pheromone trail, heuristic value, the ant private memory, and the problem constraints, the transition rules are of probabilistic nature , the most general formula is shown below in (4), this formula gives the probability an artificial ant found at point (i) will go to point (j) in the next move, i.e. selecting path (ij), at the n^{th} iteration

$$Pr_{ij}^{(n)} = \frac{[ph_{ij}^{(n-1)}]^{a}[y_{ij}]^{b}}{\sum_{j}[ph_{ij}^{(n-1)}]^{a}[y_{ij}]^{b}} \quad \text{....(4)}$$

where,

Pr_{ij}^{n} Is the probability that artificial ant move from point (i) to point (j) at the n^{th} iteration;

Ph_{ij}^{n-1} Is the net pheromone value along the path (ij) at the end of $(n-1)^{th}$ iteration;

$\mathbf{y_{ij}}$ Is the heuristic value (desirability) of the path (ij) ;

a, b Are Control variables which determine the relative influence of pheromone trail (**Ph_{ij}**) and heuristic value (Y_{ij}) so when a=0 we depend on heuristic value only in calculating the transition rule (4) and when b=0 then we depend on pheromone trail only.

Usually after calculating all the probabilities of movement to each permissible point, a random number is generated (0,1) and the Monte Carlo wheel used to find the corresponding point the ant will move to.

F. The construction procedure ends when any termination condition is satisfied, usually when an objective state is reached, or after certain predetermined number of iterations is carried out

G. After the artificial ant reaches the objective state, the objective function or cost becomes obvious , the ants use their memory and trace back the path they follow and update the pheromone trail, there are two methods for updating the pheromone trail which are:

I. During the construction procedure, when an ant move from one point (or node or state or…etc) say point (i) to another point say (j) it update the pheromone trail immediately; this method of updating known as *online step – by- step pheromone trail update*

II. The ant allows to update the pheromone trail only after it finishes the path, so the ant trace back the traveled path and updates the points it passes through, this method known as *online delayed pheromone trail update* and it is most popular than the first one.

H. The mechanism of pheromone evaporation which deposited by artificial ants is different from the evaporation of real ants pheromone, they usually designed to enable ants to forget their history and encourage them to search new places of the solution space, the most popular formula for updating pheromone trail is shown bellow

$$Ph_{ij}^{n} = \begin{cases} Ph_{ij}^{n-1}(1-v) & \text{if the route (ij) dose not used by ants in } (n-1)^{th} \text{ iteration} \\ Ph_{ij}^{n-1}(1-v)+T^{n-1} & \text{if the route (ij) used by ants at } (n-1)^{th} \text{ iteration.} \end{cases} \quad \ldots\ldots\ldots (5)$$

Where **0<v<1** represent the evaporation rate which is constant during the iterations, and $T^{(n-1)}$ represents how good the food found at the end of iteration (n-1), we can see from equation (5) above that all routes will be evaporated first then only the routes that ants chose in the iteration will get extra pheromone by quantity $T^{(n-1)}$, the new iteration (n) will use the new pheromone distribution to guide the ants in the search of a new solution through out the solution space.

Ant System Optimization:

There are mainly (5) basic algorithms of the ant optimization metaheuristic, Dorgio [4], Ant system, Ant colony system, Max-Min ant system, Ranked –Based ant system, and the Best-Worst ant system. This paper will use the ant system to introduce a probabilistic method to find approximate solution of the linear programming problems, other methods can also be applied but with additional steps to fulfill their requirements.

Ant system (AS) method which was developed by Dorgio[4] ,assumed the first ant optimization algorithm and it have three variants, they are :

I.AS- density, pheromone updated using online step-by-step method and a given constant amount of pheromone added each time

II.AS- quantity, pheromone updated using online step-by-step method but the amount added depend on (Y_{ij}).

III. As-cycle, the pheromone updated at the end of the cycle using the online delayed pheromone update and the quantity added depend on the value of the solution. Experiment shows that AS-cycle is the best performance among the other.

Ant system for linear programming:

As we saw before, the Lp problem always has (n>m) where (n) represent the number of decision variables and (m) represents the number of constraints. To find a solution we have to set (n-m) variable equal to zero (i.e. non basic variables) then solve for the value of the rest of variables (i.e. basic variables) which should satisfy all the constraints, then pick the one that makes the objective function optimum, to apply the ant system we have to shift the search idea and concentrate on non basic variable instead of the basic variable so we have to search the space solution that consist of n!/m!(n-m)! vertices and compare the value of the objective function at each vertex to pinpointed the optimum one that satisfies all the constraints. We will use ant system to make such a search. To do so let us calculate first (D) which represents the number of non basic variables and can be found easily as shown below:

$$\mathbf{D=n-m} \qquad \ldots\ldots \quad (6)$$

As shown in Fig (1), we will release (D) ants from the nest at the beginning of each iteration, and each ant will be forced to choose one of the (remaining) decision variables according to the transition rule explained earlier but with some modifications, the transition rule, which we will apply is shown in equation (7), the variables that the ants chose will be assigned as non basic variables and set to zero.

In Fig (1) D- ants released at each iteration and each ant will in turn choose one variable to be non basic variable, ant 1 chose from n variables, ant 2 chose from n-1 remaining variables,….., ant D chose from n- D+1 remaining variable

$$Pr_{ij}^{(n)} = \frac{[ph_{ij}^{(n-1)}]^{a}\,[Y_j]^{b}\,H_j^{\,n}}{\sum_j [ph_{ij}^{(n-1)}]^{a}\,[Y_j]^{b}\,H_j^{\,n}} \quad \ldots\ldots(7)$$

Where

i is the index of the (D) ant released at each iteration (not like the original rule where it was an index of the point to be moved from).

J is the index of the decision (n) variables (basic and non basic).

$Ph_{ij}^{(n-1)}$ is the pheromone that ant (i) will see along the path from the nest to variable (j) at n^th iteration .

$\mathbf{Pr}_{ij}^{n}$ is the probability that ant (i) will chose variable x_j as non basic variable when it released from the nest.

$\mathbf{Y_j}$ is the heuristic value represents how much the variable (x_j) is attractive to ants and is calculated as shown bellow, the index (i) removed since the heuristic value is the same for all ants

For minimization problems

$$Y_j = \begin{cases} \left|\dfrac{1}{C_j}\right| & \text{if } C_j < 0 \\ C_j & \text{if } C_j > 0 \\ 1 & \text{if } C_j = 0 \end{cases} \quad \ldots(8)$$

For maximization problems

$$Y_j = \begin{cases} |C_j| & \text{if } C_j < 0 \\ \dfrac{1}{C_j} & \text{if } C_j > 0 \\ 1 & \text{if } C_j = 0 \end{cases} \quad \ldots(9)$$

where $\mathbf{C_j}$ is the coefficient of the $\mathbf{x_j}$ in the objective function, to understand how this heuristic work, one should first, remember that the heuristic value represents the desirability of the decision variable for the ant, secondly, referring to equation (8) above and suppose that we handle a minimization problem then we want variables with negative coefficients to be appear in the basic variable set, while one with a positive coefficient to be appear with non basic variables set in order to minimize he objective function, and this how the function of ($\mathbf{Y_j}$) works, it gives higher value to the positive coefficients and lower values to the negative coefficients, of course this is true since both coefficients are >1, otherwise one should first multiply the objective function with suitable constant to ensure that none of the $\mathbf{C}_j$ is < 1, at last, the coefficients of slack variables are zero in the objective function and we use $\mathbf{Y_j}=1$ for it.

$$\mathbf{H^n_j} = \begin{cases} 1 & \text{at the beginning of each iteration for all (j)} \\ 0 & \text{when any ant chose the } j^{th} \text{ variable to be non-basic variable at } n^{th} \text{ iteration} \end{cases}$$

$\mathbf{H^n_j}$ is used for two reasons:

I. To ensure that no variable will be chosen twice during a single iteration to be a non basic variable, i.e. no two ants will chose the same variable

II. To adjust the effect of removing the chosen variables by previous ants in the same iteration, on the transition rule calculations.

Pheromone update:

After each iteration one should update the pheromone trials used by ants in the iteration so that ants in the next iteration make use of the result of the previous iteration results. The method that we used in this paper is different from the usual methods of pheromone update, this because we face the fact that the solution generated by solving the (m) equations may be either feasible or infeasible, it is clear that if the solution violate any constraints including non negativity constraints it will be considered infeasible; so we must test the solution first and decide whether it is feasible or infeasible and according to the test results we update the pheromone trails by one of the two methods shown bellow :

A. *feasible solution (reward),* when we find a feasible solution, equation (10) shown bellow will be used to update the pheromone trails of the paths chosen by the ants

$$ph_{ij}^{(n)} = (1 - n)\, ph_{ij}^{(n-1)} + T^{(n-1)} S_{ij} \quad \ldots(10)$$

where

$$S_{ij} = \begin{cases} 1 & \text{if ant (i) chose variable (j) to be non basic variable} \\ 0 & \text{else where} \end{cases}$$

$\mathbf{T^{(n-1)}}$ represents the reward that the ants will get when they find feasible solution, this rewards go to the ants as an extra pheromone added to the residual pheromone trails after evaporation, and this extra pheromone added only to the paths the ants choose in the $(n-1)^{th}$ iteration, this in turn will make the assigned non basic variables have a bigger chance to appear again as non basic variables in next iterations, we suggest two methods for evaluating the reward, so if we let the value of the objective function at the end of $(n-1)^{th}$ iteration which results in a feasible solution is $Z^{(n-1)}$, then the two methods are:

I . First method (AS-cycle)

$$T^{(n-1)} = \begin{cases} \text{or maximization} & Z^{(n-1)} \\ \text{For minimization} & \left|\dfrac{1}{z^{(n-1)}}\right| \end{cases} \quad ...(11)$$

This method is better than the second method (will be explained next) since it finds feasible solution faster, but we must ensure that $Z^{(n-1)}$ is not equal to zero, and there is no sign change in the value of $Z^{(n-1)}$ during the iterations

II. Second method (AS-density)

$$T^{(n-1)} = k \quad \ldots (12)$$

where k is a constant, i.e. the reward the ants get dose not depend on the value of the objective function, the ants get the reward since they found a feasible solution and for both minimization or maximization problems.

B. *Infeasible solution (punishment)*, in case the ants find a solution which is infeasible (violate one or more constraints) then instead of rewarding ants by assigning extra pheromone to the paths they chose from nest to the (D) non basic variables we punish them by decreasing the pheromone in the paths they have chosen in addition to the usual decrease of pheromone due to evaporation and the following equation will be used to update the pheromone.

.

$$ph_{ij}^{(n)} = (1 - v - wS_{ij})ph_{ij}^{(n-1)} \quad ....(13)$$

where $0 < w < 1$ is the punishment factor and represents the percentage of pheromone erased as a punishment to the ants because they have chosen paths that lead to an infeasible solution. At last one should be careful in choosing the value of both evaporation rate and punishment factor to avoid negative values of pheromone, their sum should be less than one (i.e. $v + w < 1$).

Evaluating the approximate solution:

Since we expected an approximate (near optimum) solution in general not an exact (optimum) one by using the ant system optimization, then it is very useful to have a method to evaluate this approximation to see how much it is near to the exact solution (which is unknown) the evaluation method we suggest is to solve the dual problem with primal one at the same time, so instead of repeating the calculation (n) times of the primal problem we carry (n/2) times the primal and (n/2) times the dual this will provide us with the upper bound and the lower bound of the exact solution, Nocedal[8], and if the difference between them is acceptable (say 5% or 10%) we accept the approximate solution and stop the calculations otherwise we should repeat the iterations until an acceptable difference between the bound is reached of course any time the two bound intersect, we can say that we reach an exact solution not an approximate one, fig (2) shows the expected behavior of the bound (upper and lower) for a minimization problem.

By solving the primal and dual problem at the same time one can also check if the problem in hand is unbounded, this can be done by using the weak duality principle, Rao[1], so whenever the e lower bound goes above the upper bound (or vise versa) then this is an indicator that we are handling unbounded linear programming problem.

Advantages and disadvantages

The main advantages and disadvantages of the suggested method compared with the simplex algorithm are shown bellow, we use to compare with simplex method since it is the most popular and practical method used to solve the linear programming problem.

A. *parallel computation*. There are two main types of parallel computation, Grotschel[12], the fine grained and the coarse grained models. In fine grained method the problem split into individuals and each individual assigned to a computer (or processor) and there is an exchange of information between these computers but each computer has its own assigned part of calculation , this is usually done with simplex method where the simplex tableau divided between many computers each carry calculation for some fixed number of columns (pivoting) and by collecting the information from the various computers the simplex tableau is assembled again, the main disadvantages here is that no guarantee is given that all the computer will share the same load and if any computer finish the calculation first it must wait (i.e. idle) until the last computer finish the calculation so we can proceed to the next pivoting iteration. In coarse grained method the problem is loaded fully to each computer and calculation rarely depend on the other computer calculation the coarse grained parallelization is most promising and load balance can be achieved to acceptable percentages specially when there is a requirement for online optimization or real time online optimization (as will be explained latter), it is clear the simplex method cannot be carried out through coarse grained parallelization since there is no benefit can be gained by solving the same tableau on different computers at the same time, but for ant colony optimization the coarse grain parallelization is useful

B.*Online optimization*, Generally speaking if the model of optimization have to give answers (computed solution) each time a piece of data become available we call it online optimization model, Grotschel[12], if this answer should be given in a small time interval we call it real time-online models, following the literature online model divided into two main categories the sequence model where the optimization model have to give solutions according to the sequence of availability of data and time stamp model where the model have to give solution in predetermined points of time according to the data available to it, in both models the parallel computation is an important tool to achieve these goals specially with time stamp models and as shown earlier the suggested method is more suitable for parallel computation than the simplex algorithm

C. The suggested method doesn't need artificial variables because the slack and surplus variable are enough to carry the calculation and achieve the solution, so in general the suggested method deals with less variables

D. No degeneracy and cycling.

E. The suggested method opens the door widely to make use of any improvement take place in linear algebra especially in the methods of solving system of linear equations to link it to the linear programming problem.

F. Although the suggested method is a random search but it is guided not like other random search method (like Monte Carlo, greedy search… etc)

G. The suggested method doesn't involve exponential worst-case as in simplex method and depend very much on the method used to solve the system of linear equations

H. Very suitable for problems where the value of the objective function is immature like constraint satisfaction problems, graph colony problem, time tables and weapon target assignment problem

I. The main disadvantages is that no guarantee that the optimum will found and this problem over come by parallel computation as explained earlier.

Example:

Suppose we have to find the solution of the following linear programming

Max $Z= 20x_1 + 18x_2 + 9x_3 - 4x_4 + 6x_5$

s.t. $90x_1+65x_2+23x_3 + 14x_4 + 9x_5 \leq 530$

$81x_1 + 57x_2 + 11x_3 + 8x_4 + 6x_5 \geq 390$

$73x_1 + 41x_2 + 9x_3 + 3x_4 = 289$

This problem has a solution as $Z=194.9772$ $x_1=3.9589$, $x_5=19.2998$, $x_2=x_3=x_4=0$, this solution can be found by using ordinary simplex tableau after (7) iterations.

To apply our method we have first to change inequalities to equalities i.e. the problem become

Max $Z= 20x_1 +18x_2 +9x_3 - 4x_4 +6x_5$

S.T.

$90x_1+65x_2+23x_3+14x_4+9x_5+x_6 = 530$

$81x_1+57x_2 + 11x_3 + 8x_4 + 6x_5 - x_7 =390$

$73x_1 + 41x_2 + 9x_3 + 3x_4 = 289$

So our linear programming problem has number of variables (n=7), and number of equations (m=3), then

D=7-3=4

We will use the following data to solve the above linear programming problem:

Evaporation rate v = 0.1 .
Control variables a= b = 1 .
Initial pheromone for all paths = 100 .
Number of ants D = 4 .
Number of iterations = 100, split as 50 for primal and 50 for dual .

For pheromone update, If ants find feasible solution, we use AS-density method with k equal to 20, but if ants find infeasible we use punishment factor equals to 0.25, the Stop criteria is when total number of iterations reached or when the upper bound equals the lower bound (i.e. optimum solution found), or when lower bound cross the upper band (or vise versa) at any time because this means that the problem is unbounded.

The heuristic value can be found using equation (9) and the result shown in table (1).

As it is shown in table (1) the heuristic value which represents the desirability of a variable to an ant to choose it as a non basic variable is high for negative coefficients of the objective function while for positive coefficients is much less and depend on the value of the coefficient this will lead to increasing the chance of the negative coefficient to be a non basic variable and when talking about the positive coefficients the less positive coefficient has greater chance to be non basic variable than the most positive one the slack and surplus variable has the value equal to (1) since their coefficient is zero in the objective function.

Now applying equation (7) four times since we have to release 4 ants in each iteration give us the first set of suggested non basic variable then solving the linear programming model for the rest of the variable and repeat this process for a predetermined number (in our example it was 100, 50 for primal and 50 for dual) we have the result shown in table (2) After each iteration we test for feasibility, in case we have feasible solution we use equation (12) to update the pheromone trails (as rewards) but in case we have infeasible solution we use equation (13) to update the pheromone (as punishment).

Using the same procedure that has been explained above we solve also the dual of the example and we get the results in table (3). To check how the method works for on-line optimization we change the right hand side of the first constraint (i.e. b1) as shown in table (4), the optimum values of the objective for each value of (b1) also shown in table (4), these values of objective function found by using ordinary simplex method , table (4) also shows the iteration number at which the new value of (b1) take place, in other words we begin the iterations with b1 equals to (530), then when iterations counter equals to (25) we change the value of (b1) to (100), and so on.

The solid line of fig. (4) represents the optimum value of the objective function for a given value of (b1), we didn't trace the value of the objective function at each iteration of the simplex method, but we simply represents them as horizontal lines during the iterations where the value of (b1) is constant . The doted line in fig. (4) shows the value of the objective function at each iteration (best known objective function value up to that iteration) using ant system optimization metaheuristic, the detailed numerical values listed in table (5), tic , as we see it always converges to the optimum, it worth to note that when changing the value of b1 in the ant system optimization metaheuristic we did not change any thing else i.e. we don't set the value of the pheromone trials to their initial value instead we use the available pheromone trial distribution and start the search for the new optimum objective function value. This is more close to practical applications of on line-optimization and of course we can't do this in the simplex method , instead we can use sensitivity analysis which is available only after doing all the required calculations to find the optimum value of the objective function and if it fails then we have to restart the calculation from the beginning .

Conclusion and further work:

We use the ant system optimization metaheuristic to solve the linear programming problems, i.e. finding the optimum values of the decision variables and the objective function, to do so we shift the search to find the optimum non basic variables, we made some modifications on the equations used in ant system optimization so we can apply it to linear programming problems, the modifications include the transition rule and pheromone trails update, we also show how we can find the value of the heuristic and its relation to the coefficient of decision variables in the objective function equation. The solution we expect from the ant system is an approximate solution in general so we show how can we estimate how good is the solution and test if it is optimum one by using duality theory, we also show how can we use the duality theory to detect the unbounded models , the main advantageous of our method is in the sense of parallel computation, on line optimization , and less variables handled during the calculations . For further work we suggest the following main directions

I. Applying the ant colony optimization metaheuristic or any other ant variants models

II. Studying the computational complexity of the suggested method and compare it with other methods used to solve the linear programming problems

III. More studies required about the control variables like a,b ,evaporation rate, …etc

References

[1] Rao S.S.; "optimization theory and applications"; 2nd edition, Wiley eastern limited, New Delhi, 1984

[2] Philips D.T.; Ravindran AA. ; And Solberg I. J., "Operation research: principles and practice

", John wiley and sons; Inc., Canada, 1976
[3] Gauss S. I., "Linear programming: methods and applications ", 4th edition, 1975
[4] Dorgio M. and Stuzle T., "Ant colony Optimization", A bradford book, MIT press, England, 2004
[5] Merkle D. and Middendorf M., "ant colony optimization with the relative pheromone evaluation method ", 3rd European workshop on evolutionary methods for AT planning, Ireland, LNCS 2279, 2002, PP325-333
[6] Kelner J. A., and Spielman D. A., "A randomized polynomial-time simplex algorithm for linear programming (preliminary version), Electronic colloquium on computational complexity, Report no-156, (2005)
[7] Vanderben R. j.,"Linear programming: fundamentals and extensions"; 2nd edition, Princeton, UAS, 2001
[8] Nocedal J. and Wright S.J.,"Numerical optimization ", 2nd edition, Springer science business media,L.L.C, USA, 2006
[9] Kolman B., "introductory linear algebra with applications", Macmillan publishing company. 1984
[10] Bayoumi M. A. and El-feky E. Z., "A directed genetic algorithm for treating linear programming problems", International journal of computer, The internet and management, Vol. 11, No. 2 , 2003.
[12]grotschel M.,Krumke S. O., Rambau j.,Winter T.,and Zimmerman U. T.; "Combinatorial online optimization in real time"; ZIB – report 01-16; july 2001 .

.

Table (1) The heuristic value using equation (9).

j	C_j	y_j
1	20	0.05
2	18	0.06
3	9	0.111
4	-4	4
5	6	0.167
6	0	1
7	0	1

Table (2) result of solving linear programming model by ASO, iterations 1,3,5,6 result infeasible solution

iteration	Z	X1	X2	X3	X4	X5
2	168.4801	0	5.26	8.14	0	0.08
4	185.5667	3.13	0	6.71	0	10.42
7	194.9772	3.959	0	0	0	19.89985

Table (3) result of solving the dual problem

iteration	Z	Y1	Y2	Y3	Y4
4	285.3656	1.90	1.85	0	0
7	266.156	1.68	1.531	0	0.107
15	194.9772	0.67	0	0	0.547

Table (4) objective function value by simplex method after changing the (b1) value

Iteration number where the new value of b1 introduced	The new value of b1	Objective function value after changing b1 to the new value using ordinary simplex method
25	1000	508.3105
50	2000	$1.17x10^3$
75	3000	$1.84x10^3$
90	1500	841.6438

.

Table (5) objective unction value after changing b1 using ant system optimization

iteration	objective function
2	168.4801
11	185.5666
19	194.9772
28	488.0976
48	508.3105
51	1154.467
54	1174.977
76	1841.644
91	821.431
95	841.6438

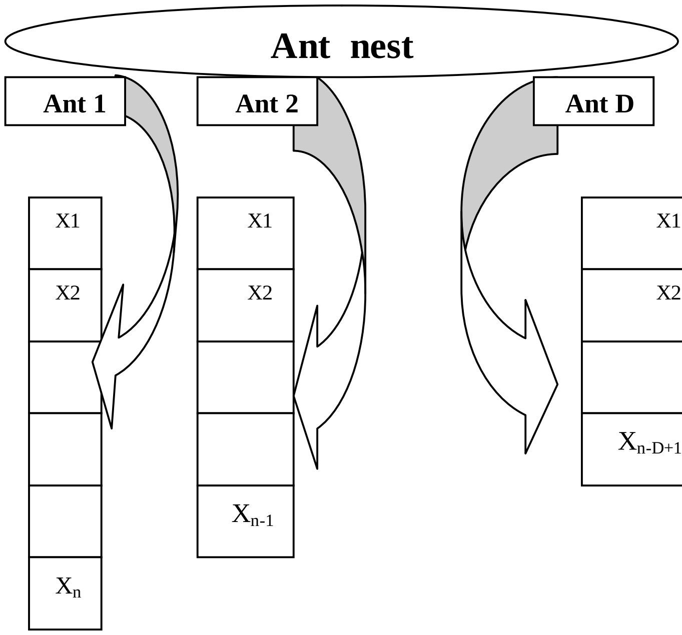

Figure (1) D- ants released at each iteration and each ant will in turn choose one variable to be non basic variable ,ant 1 chose from n variables, ant 2 chose from n-1 remaining variables,….., ant D chose from n- D+1 remaining variable

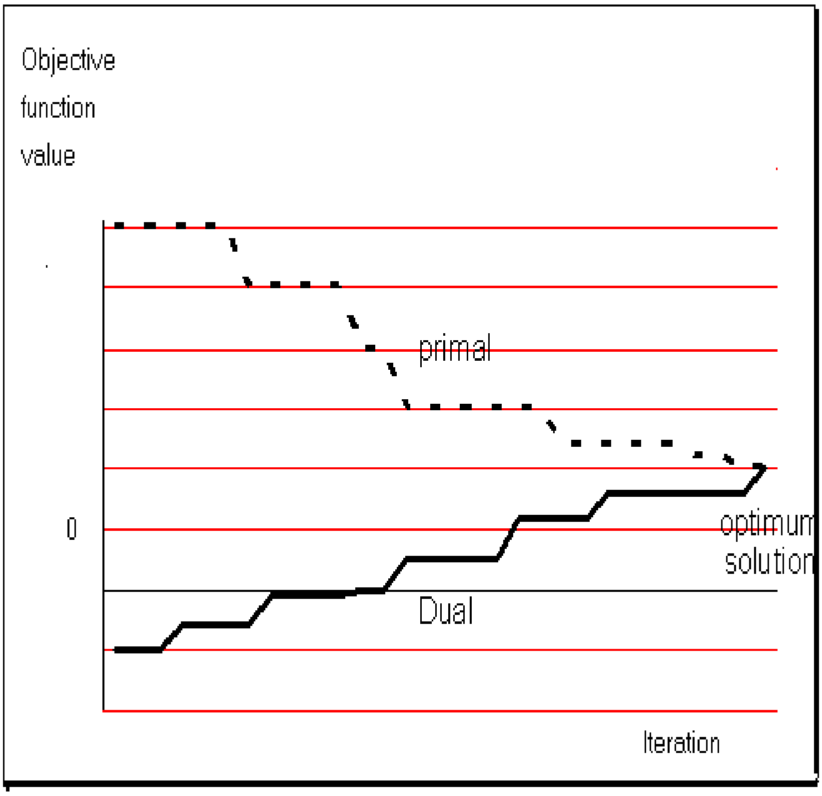

Figure (2) the upper and lower bands of the objective function for minimization Problem

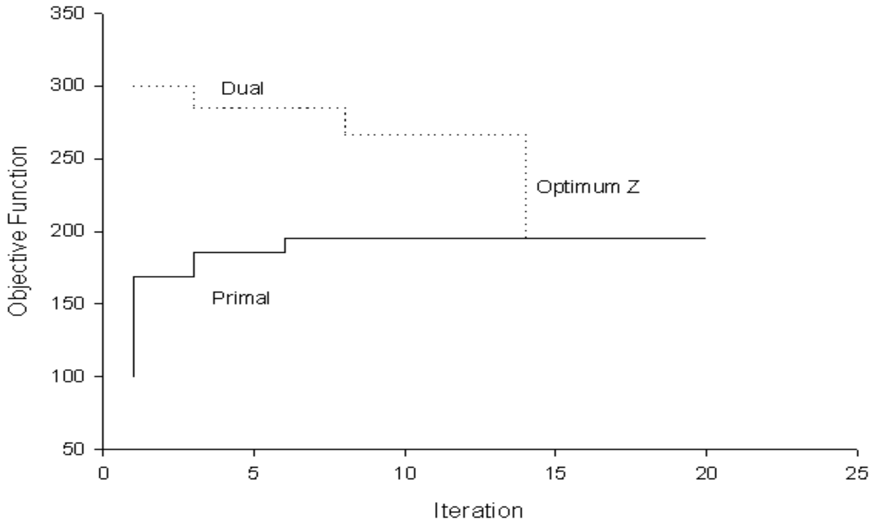

Figure (3) Coupling Dual and Primal to check for optimality of Z

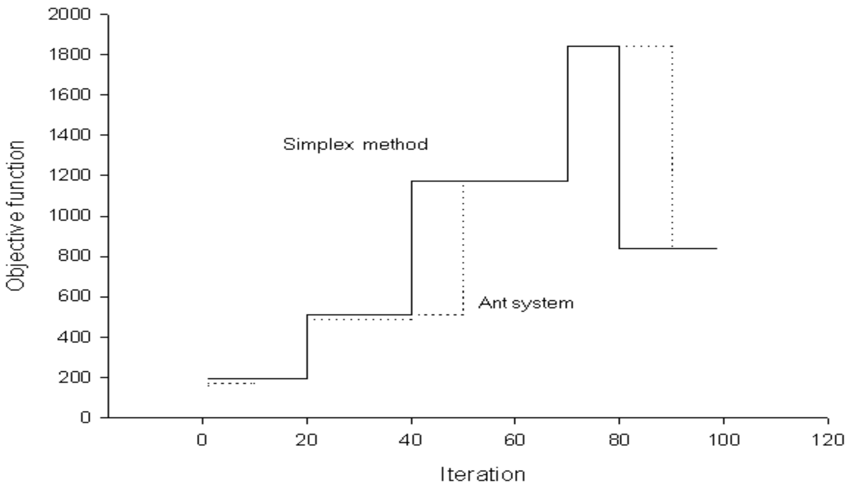

Figure (4) On line optimization using simplex method and ant system optimization metaheuristic