Using Ant System Optimization Technique for Approximate Solution to Multi-objective Programming Problems

M. El-sayed Wahed and Amal A. Abou-Elkayer
Department of Computer Science, Faculty of Computers and Information

Suez Canal University (EGYPT)

E-mail: mewahed@yahoo.com,

Abstract

In this paper we use the ant system optimization metaheuristic to find approximate solution to the Multi-objective Linear Programming Problems (MLPP), 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: Multi-objective Linear Programming problems, Ant System Optimization Problems

Introduction

The Multi-objective Linear Programming problems is an optimization method which can be used to find solutions to problems where the Multi-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 MLPP 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 MLPP. The LP type of optimization problems were first recognized in the 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 MLPP i.e. finding the optimum values of the decision variables and hence the optimum value of the set of objective functions and given minimum division about the optimal solution . In real world problem simplex method was the first practical useful approach for solving MLPP 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 MLPP 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 MLPP we will use the following standard (scalar) form

$$\text{Max} \quad Z^{(k)}(x) = \sum_{j=1}^{n} c_j^k x_j, \quad k = 1,2,\ldots,K \tag{1}$$

$$\text{Subject to:} \quad \sum_{j=1}^{n} a_{ij} x_j \le b_j, \qquad i = 1,\ldots,m, \tag{2}$$

$$x_j \ge 0, \; j = 1,\ldots,n. \tag{3}$$

Where n represents number of decision variables and m represents number of constraints, k the number of objectives, c_j, a_{ij}, and b_i 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 Multi -objective function (MOF) is optimum.

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, the 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 sometime, 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 for the Multiple objective Linear Programming:

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] :

Multi-objective decision making refers to making decision in the presence of multiple, usually conflicting, objectives. For example, in the Sales Mix problem is a very popular multiple-objective linear programming model, the decision maker (DM) wants to minimize the cost, to maximize the profit and to maximize the quality service at the same time.

A multi-objective programming is a particular case of multi-criteria problem. Mathematically, can be stated as (1-3)

In the literature this problem is often referred to as a vector maximum problem (VMP). Traditionally there are two approaches for solving (VMP). Here we give a short description for these two approaches.

(i) The first approach is to optimize one of the objectives while appending the other objectives to constraint set, so that the optimal solution would satisfy these objectives at least up to a predetermined level, a_i. the problem in this case is given as:

$$\text{Max} \quad f_j(\underline{x})$$

$$\text{Subject to:} \quad g_i(\underline{x}) \leq 0 \quad ,i=1,2,\ldots,m$$

$$f_i(\underline{x}) \geq a_i \quad , \; i=1,2,\ldots,k \quad \text{and } i \neq j$$

Where: a_i is any acceptable predetermined level for objective i.

(ii) the second approach is to optimize a super-objective function created from the given objectives, namely the weighted sum of these objectives, with previously

International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013
ISSN 2229-5518

determined weights. This approach leads to the solution of the following problem:

$$\text{Max} \ \sum_i w_i f_i(\underline{x})$$

$$\text{Subject to: } g_i(\underline{x}) \le 0 \ , \ i=1,2,\ldots,m$$

The weights are usually normalized so that $\sum_i w_i = 1$.

Any of these approaches leads to a solution which may not be the best or most satisfactory one. In the first approach we are sure which one of the objectives will be considered as our objective function and which of them will be in the constraints. On the other hand, if the decision maker provides us with a ranking of the objectives we are in a situation to choose the main objective as objective function. Another problem which face this approach is the choice of the acceptable levels a_i's in (3.5) that will result in a non empty constraint set in the first attempt for the solution. Also in such approach, may find that some of these objectives, which are transformed to the set of constraints, may decrease the feasible region or may be out of the feasible region which has no meaning.

In second approach, the major problem is in determining the proper weights w_i, i=1,2,…,m. the w_i's sensitive to the levels of a particular objective as well as the levels of all other objectives.

To eliminate the previous difficulties, many methods for multiple objective decision making (MIDM) are developed, most of them have taken place within the last decade. One of the earliest considerations of multiple objectives was given by [40].

International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013
ISSN 2229-5518

3.2.1. the step Method (STEM).

The STEP method (STEM) is, perhaps, one of the first linear multiobjective techniques to be developed. It was first described as the progressive orientation procedure by Benayoun and Tergny. (1984), [52] and later elaborated by Benayoun et al. (1971) [8].

The algorithm: [52] , [44].

This technique is based on the premise that the best-compromise solution has the minimum combined deviation from the idle point f^*. the STEP method is an interactive scheme that progressively elicits information from the DM primarily to modify the constraint set and to slightly modify the weights. This can be done at the first step by constricting a pay-off table displaying values of objective functions at the set x of optimal solution of (3.8) before the first interactive cycle.

Step1: let f_i^* , i=1,2,…,k be the feasible ideal solutions of the following k problems:

Max $f_i(\underline{x}) = \underline{c}\,t\,\underline{x}$

Subject to: $A\ \underline{x} \le b$

$\underline{x} \ge 0$, i= 1,2,…,k

As shown in the table below, row i corresponds to the solution vector x^* which maximizes the objective function f_i^k. A f_i^k is the value taken on by the i^{th} objective f_i when the i^{th} objective f_j reaches its maximum f_i^*.

Table (1): A pay-off table of STEM

	f_1	f_2	…	f_i	f_k
f_1	f_i^*	f_1^2	…	f_1^i	f_1^k
f_2	f_2^1	f_2^*	…	f_{21}^i	f_2^k
⋮	⋮	⋮	⋮	⋮	⋮
f_i	f_i^1	f_i^2	…	f_i^*	f_i^k
⋮	⋮	⋮	⋮	⋮	⋮
f_k	f_k^1	f_k^2	…	f_{k1}^i	f_k^*

Step 2. considered as a calculation phase. At the m^{th} cycle, the feasible solution to LP is considered which is the 'nearest', in the MINIMAX,sense, to the ideal solution f^*j:

Min λ.

$[\underset{-}{x}, \lambda]$

Subject to: $\lambda \geq [f_i^* - f_i(\underset{-}{x})] * w_i$,   i= 1,2,…,k

$\underset{-}{x} \in x^m$

$\lambda \geq 0$

Where x^m is the solution set of $A\underset{-}{x} \leq b, x \geq 0$, plus any constraint added in the previous (m-1) cycles, w_i give the relative importance of the distances to the optimal. But it must be noted that they are only locally effective and are not of overriding importance as weights are in the utility method from the jth column in the pay-off table, f_i^* is the

International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013
ISSN 2229-5518

maximum value of the column and $f_i^{\min}$ be the minimum value, then w_i are chosen such that:

$$w_i = \frac{\alpha_i}{\sum_i \alpha_i}$$

Where $\alpha i = \frac{f_i^{\min} - f_i^*}{f_i^{\min}} \left[\frac{1}{\sqrt{\sum_{i=1}^{n} (cij)2}} \right]; if \ f_i^* \le 0$

$$\alpha i = \frac{f_i^* - f_i^{\min}}{f_i^*} \left[\frac{1}{\sqrt{\sum_{i=1}^{n} (cij)2}} \right]; if \ f_i^* > 0$$

Where the $c_{ji's}$ are the coefficients of the jth objective. The value of αi consists of two terms:

$$\frac{f_i^* - f_i^{\min}}{f_i^*} \ \ _{or} \ \frac{f_i^{\min} - f_i^*}{f_i^{\min}} \ \ _{and} \ \frac{1}{\sqrt{\sum_{i=1}^{n} (cij)2}}$$

Form the first term we can make the following assertion:

If the value of f_i does not vary much form the optimum solution by varying $\underline{x}$, the corresponding objective is not sensitive to the variation in the weighting values. Therefore, a small weight w_i will become correspondingly larger. The second term normalizes the values taken by the objective functions. The αi is used to define in such a way that the sum of weights equals to one.

This means that different solutions obtained from different weighting strategies can be easily compared.

Step 3: Ask the decision maker to compare (f_1 (x_q^*), f_2 (x_q^*),…, fn (x_q^*))^t with ($f_1^*, f_2^*, \cdots, f_n^*$).

(a) If the decision maker is satisfied with the current solution, stop- the best – compromise solution has been found.

(b) If there is no satisfactory objectives, stop- no best compromise solution can be found by this method.

(c) If there are some satisfactory objectives, ask the decision maker to select one such objective f_j^* and the amount Δf_j^* to be sacrificed (increased) in exchange for an improvement of some unsatisfactory objectives.

Step 4: if q= n, stop-no best-compromise solution can be found by this method. Otherwise set q= q+1, compute xq

Where $x^q = x^{-1} \cap x^{-q}$

$$x^{-q} = \{x \in x / f_{\hat{j}}(x) \le f_{\hat{j}}(x^{q-1}) + Df_{\hat{j}} \text{ and } f_j(x) \le f_j(x^{q-1}), \quad \forall j \ne \hat{j}\}$$

The weights should be modified accordingly by setting

$$w_i = \frac{\alpha i}{\sum_i \alpha i}$$

The go to step 1.

Step 5. It search for minimum cost feasible solution for the problem being solved (i.e. shortest path)

Step 6. 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.

Step 7. It has initial state that usually corresponds to a unitary sequence, and one or more termination condition.

Step 8. It starts with the initial state and moves towards feasible states, building its associated solution incrementally.

Step 9. 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 nth iteration

$$\sum_{k=1}^{l} PR_{ij}^{(n)} = \sum_{k=1}^{l} \frac{\left[Ph_{ij}^{(n-1)}\right]^{a}\left[y_{ij}\right]^{b}}{\sum_{j}\left[Ph_{ij}^{(n-1)}\right]^{a}\left[y_{ij}\right]^{b}} \qquad (4)$$

where,

k: Is the number of objective functions;

$PR_{ij}^{(n)}$: 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 $(n\text{-}1)^{th}$ 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 ($\mathbf{Ph_{ij}}$) 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

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

Where $\mathbf{0<v<1}$ represent the evaporation rate which is constant during the iterations, and $\mathbf{T^{(n-1)}}$ 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 $\mathbf{T^{(n-1)}}$**,** 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 ($\mathbf{Y_{ij}}$).

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!(m-n)!) 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:

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

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

$$Pr_{ij}^{(n)} = \frac{\left[ph_{ij}^{(n-1)}\right]^{a}\left[y_{ij}\right]^{b} H_{j}^{n}}{\sum_{j}\left[ph_{ij}^{(n-1)}\right]^{a}\left[y_{ij}\right]^{b} H_{j}^{n}} \qquad (7)$$

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 nth iteration .
ph_{ij}^{rn} is the probability that ant (i) will chose variable xj as non basic variable when it released from the nest.
Yj is the heuristic value represents how much the variable (xj) 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

$$Y_J = \begin{cases} \left|\dfrac{1}{\mathbf{C}_j}\right| \text{ if } Cj < 0 \\ c_j \text{ if } Cj < 0 \\ 1_i \quad \text{if } Cj = 0 \end{cases} \qquad (8)$$

For maximization problems

$$Y_J = \begin{cases} |\mathbf{Cj}| \text{ if } Cj < 0 \\ \dfrac{1}{Cj} \text{ if } Cj < 0 \\ 1_i \quad \text{if } Cj = 0 \end{cases} \qquad (9)$$

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.

$$H_j^n = \begin{cases} 1 & \text{at the beginning of each iteration for all (j)} \\ 0 & \text{when any ant chose the jth variable to be nonbasic variable at nth iteration} \end{cases}$$

$\mathbf{H_j^n}$ 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

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

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

$\mathbf{T^{(n-1)}}$ 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 $\mathbf{Z^{(n-1)}}$, then the two methods are:

I . First method (AS-cycle)

$$\mathbf{T}^{(\mathbf{n-1})} = \begin{cases} Z^{(n-1)} & \max imization \\ \dfrac{1}{Z^{(n-1)}} & \text{minimization} \end{cases} \tag{11}$$

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

II. Second method (AS-density)

$$\boldsymbol{T}^{(\boldsymbol{n}\text{-}\mathbf{1})} = \boldsymbol{k} \tag{12}$$

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.

$$\mathrm{ph}_{\mathrm{ij}}^{\mathrm{n}} = (1 - \mathrm{v} - \mathrm{ws}_{\mathrm{ij}})\,\mathrm{ph}_{\mathrm{ij}}^{(\mathrm{n}-1)} \tag{13}$$

where $\mathbf{0} < \boldsymbol{w} < \boldsymbol{1}$ 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. $\mathbf{v + w < 1}$).

Numerical example

Suppose we have to find the solutions of the following two linear programming

Min Z(1)= -x1 -2x2

Max Z(2)= -3x1 -x2

S.T.

x1 + x2 = 6

2x1 + x2 ≤ 9

2x1 ≤ 4

x2 ≤ 5

x1 , x2 ≥ 0

We obtain the equivalent deterministic programming problem for the above multi-objective programming problem by using Eqs. (8) - (10).

This first problem has a solution as:

Z (1) = -11, x1=1, x2 =5, this solution can be found by using ordinary simplex tableau after (2) iterations. And the second problem has a solution as Z (2)= - 10, x1= 2, x2= 4, this solution can be found also by using ordinary simplex tableau after (2) iterations.

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

Min Z(1)= - x1 - 2x2

S.T.

x1 + x2 = 6

2x1 + x2 $+ x3 = 9$

2x1 $+x4 = 4$

x2 $+x5 = 5$

So our linear programming problems has number of variables

(n=5), and number of equations (m=4), then D=5- 4 = 1

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 = 1.

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 (8) 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.

For each linear programming problem we applying equation (7) one time since we have to release 1 ant 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 and we have the result shown in table (2, 3) After each iteration we test for feasibility, in case we have feasible solution we use equation (10) to update the pheromone trails (as rewards) but in case we have infeasible solution we use equation (13) to update the pheromone (as punishment).

Table (1) the heuristic value using equation (8)

j	C1j	Y1j	C2j	Y2j
1	-1	1	-3	0.333
2	-2	0.5	-1	1
3	0	1	0	1
4	0	1	0	1
5	0	1	0	1

Min Z(1) = - x1 - 2x2

S.T.

x1 + x2 = 6

2x1 + x2 $+ x3 = 9$

2x1 $+x4 = 4$

x2 $+x5 = 5$

Iteration 1

Ph = [100, 100, 100, 100, 100]
Ant 1
$\mathrm{Pr}_{11}^{(1)} = \frac{100*1}{100*1+100*0.5+100*1+100*1+100*1} = 0.222$

$Pr_{12}^{(1)} = 0.111$
$Pr_{13}^{(1)} = 0.222$
$Pr_{14}^{(1)} = 0.222$
$Pr_{15}^{(1)} = 0.222$

We select the large probability of ant1 is x1 this due to this variable in nonbasic and this solution is infeasible solution
We apply the equation (13) in case the ants find a solution which is infeasible then Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\, ph_{ij}^{(n-1)}$$

Ph = [65, 90, 90, 90, 90]

Iteration 2
Ant 1

$Pr_{11}^{(2)} = \frac{65*1}{65*1+90*0.5+90*1+90*1+90*1} = 0.171$
$Pr_{12}^{(2)} = 0.118$
$Pr_{13}^{(2)} = 0.237$
$Pr_{14}^{(2)} = 0.237$
$Pr_{15}^{(2)} = 0.237$

We select the large probability of ant1 is x2 this due to this variable in nonbasic and this solution is infeasible solution
We apply the equation (13) in case the ants find a solution which is infeasible then Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\, ph_{ij}^{(n-1)}$$

Iteration 3
Ph = [58.5, 58.5, 81, 81, 81]
Ant 1

$Pr_{11}^{(3)} = \frac{58.5*1}{58.5*1+58.5*0.5+81*1+81*1+81*1} = 0.177$
$Pr_{12}^{(3)} = 0.088$
$Pr_{13}^{(3)} = 0.245$
$Pr_{14}^{(3)} = 0.245$
$Pr_{15}^{(3)} = 0.245$

We select the large probability of ant1 is x3 this due to this variable in nonbasic and this solution is feasible solution and we have the result shown in table (2).

Table (2) result of solving the first linear programming model

by ant system optimization (ASO) iterations 1,2 result infeasible solution

iteration	Z1	X1	X2
1	-10	0	5
2	- 4	4	0
3	-11	1	5

Min Z(1) = - 3 x1 - x2

S.T.

x1 + x2 = 6

2x1 + x2 + $x3 = 9$

2x1 $+x4 = 4$

x2 $+x5 = 5$

Iteration 1

Ph = [100, 100, 100, 100, 100]

Ant 1

$Pr_{11}^{(1)} = \frac{100*0.333}{100*0.333+100*1+100*1+100*1+100*1} = 0.077$

$Pr_{12}^{(1)} = 0.231$

$Pr_{13}^{(1)} = 0.231$

$Pr_{14}^{(1)} = 0.231$

$Pr_{15}^{(1)} = 0.231$

We select the large probability of ant1 is x2 this due to this variable in nonbasic and this solution is infeasible solution

We apply the equation (13) in case the ants find a solution which is infeasible then Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\,ph_{ij}^{(n-1)}$$

Ph = [90, 65, 90, 90, 90]

Iteration 2

Ant 1

$Pr_{11}^{(2)} = \frac{90*0.333}{90*0.333+65*1++90*1+90*1+90*1} = 0.082$

$Pr_{12}^{(2)} = 0.178$

$Pr_{13}^{(2)} = 0.246$

$Pr_{14}^{(2)} = 0.246$

International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013
ISSN 2229-5518

$\mathrm{Pr}_{15}^{(2)} = 0.246$

We select the large probability of ant1 is x1 this due to this variable in nonbasic and this solution is infeasible solution
We apply the equation (13) in case the ants find a solution which is infeasible then
Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\, ph_{ij}^{(n-1)}$$

Iteration 3

Ph = [81, 42.25, 81, 81, 81]

Ant 1

$\mathrm{Pr}_{11}^{(3)} = \frac{81*0.333}{81*0.333+42.25*1+81*1+81*1+81*1} = 0.086$

$\mathrm{Pr}_{12}^{(3)} = 0.135$

$\mathrm{Pr}_{13}^{(3)} = 0.259$

$\mathrm{Pr}_{14}^{(3)} = 0.259$

$\mathrm{Pr}_{15}^{(3)} = 0.259$

We select the large probability of ant1 is x4 this due to this variable in nonbasic and this solution is feasible solution and we have the result shown in table (3).

Table (3) results of solving the second linear programming model by ant system optimization (ASO), iterations 1,2 result infeasible solution

iteration	Z2	X1	X2
1	-12	4	0
2	- 5	0	5
3	-10	2	4

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.

Min $\propto$

$$\begin{array}{lllllll}
\propto - 0.3614\,x1 & - 0.723x2 & +x3 & & & & = 3.975 \\
\propto - 1.916\,x1 & - 0.6386\,x2 & & + x4 & & & = 6.386 \\
x1 & + x2 & & & & & = 6 \\
2x1 & + x2 & & & + x5 & & = 9 \\
2x1 & & & & & +x6 & = 4 \\
& x2 & & & & +x7 & = 5
\end{array}$$

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

That number of variables (n=8), and number of equations (m=6), then D=8- 6 = 2

Evaporation rate v = 0.1.

Control variables a= b = 1.

Initial pheromone for all paths = 100.

Number of ants D = 2.

Table (4) the heuristic value using equation (8)

j	Cj	Yj
1	1	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1

Iteration 1

Ph = [100, 100, 100, 100, 100, 100, 100,100]

Ant 1

$$\mathrm{Pr}_{11}^{(1)} = \frac{100*1}{100*1+100*1+100*1+100*1+100*1+100*1+100*1+100*1} = 0.125$$

$$\mathrm{Pr}_{12}^{(1)} = 0.125$$

$$\mathrm{Pr}_{13}^{(1)} = 0.125$$

$$\mathrm{Pr}_{14}^{(1)} = 0.125$$

$$\mathrm{Pr}_{15}^{(1)} = 0.125$$

$Pr_{16}^{(1)} = 0.125$
$Pr_{17}^{(1)} = 0.125$
$Pr_{18}^{(1)} = 0.125$

We select the large probability of ant1 is x1 this due to this variable in nonbasic
Ant 2
$Pr_{21}^{(1)} = \frac{100*1}{100*1+0\ +100*1+100*1+100*1+100*1+100*1+100*1} = 0.143$
$Pr_{22}^{(1)} = 0.143$
$Pr_{23}^{(1)} = 0.143$
$Pr_{24}^{(1)} = 0.143$
$Pr_{25}^{(1)} = 0.143$
$Pr_{26}^{(1)} = 0.143$
$Pr_{27}^{(1)} = 0.143$
$Pr_{28}^{(1)} = 0.143$

We select the large probability of ant2 is x3 this due to this variable in nonbasic and this solution is infeasible solution
We apply the equation (13) in case the ants find a solution which is infeasible then Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\, ph_{ij}^{(n-1)}$$

Ph = [90, 65, 90, 65, 90, 90, 90, 90]

Iteration 2

Ant 1
$Pr_{11}^{(2)} = \frac{90*1}{90*1+65*1+90*1+65*1+90*1+90*1+90*1+90*1} = 0.134$
$Pr_{12}^{(2)} = 0.097$
$Pr_{13}^{(2)} = 0.134$
$Pr_{14}^{(2)} = 0.134$
$Pr_{15}^{(2)} = 0.134$
$Pr_{16}^{(2)} = 0.134$
$Pr_{17}^{(2)} = 0.134$
$Pr_{18}^{(2)} = 0.134$

We select the large probability of ant1 is x2 this due to this variable in nonbasic

Ant 2

$Pr_{21}^{(1)} = \frac{90*1}{90*1+65*1+0+65*1+90*1+90*1+90*1+90*1} = 0.155$
$Pr_{22}^{(1)} = 0.112$
$Pr_{23}^{(1)} = 0.155$
$Pr_{24}^{(1)} = 0.155$
$Pr_{25}^{(1)} = 0.155$
$Pr_{26}^{(1)} = 0.155$
$Pr_{27}^{(1)} = 0.155$
$Pr_{28}^{(1)} = 0.155$

We select the large probability of ant2 is x4 this due to this variable in nonbasic and this solution is infeasible solution
We apply the equation (13) in case the ants find a solution which is infeasible then Update the pheromone

$$ph_{ij}^{n} = (1 - v - ws_{ij})\, ph_{ij}^{(n-1)}$$

Iteration 3
Ph = [81, 58.5, 58.5, 58.5, 58.5, 81, 81, 81]
Ant 1
$Pr_{11}^{(3)} = \frac{81*1}{81*1+58.5*1+58.5*1+58.5*1+58.5*1+81*1+81*1+81*1} = 0.145$
$Pr_{12}^{(3)} = 0.105$
$Pr_{13}^{(3)} = 0.105$
$Pr_{14}^{(3)} = 0.105$
$Pr_{15}^{(3)} = 0.105$
$Pr_{16}^{(3)} = 0.145$
$Pr_{17}^{(3)} = 0.145$
$Pr_{18}^{(3)} = 0.145$

We select the large probability of ant1 is x8 this due to this variable in nonbasic

Ant 2
$Pr_{21}^{(1)} = \frac{81*1}{81*1+58.5*1+58.5*1+58.5*1+58.5*1+81*1+81*1+0} = 0.169$
$Pr_{22}^{(1)} = 0.123$
$Pr_{23}^{(1)} = 0.155$
$Pr_{24}^{(1)} = 0.123$
$Pr_{25}^{(1)} = 0.123$
$Pr_{26}^{(1)} = 0.169$
$Pr_{27}^{(1)} = 0.169$
$Pr_{28}^{(1)} = 0.169$

We select the large probability of ant2 is x8 this due to this variable in nonbasic and this solution is feasible solution and we have the result shown in table (5).

Table (5) result of solving the multi-objective programming model by ant system optimization (ASO), iterations 1,2 result infeasible solution

iteration	∝	X1	X2
1	1.22e-15	0	5.5
2	8.8 e-16	4	0
3	7.9514	1	5

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 .

References

[1] Rao S.S.; "optimization theory and applications"; 2ndedition, 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 andmanagement, Vol. 11, No. 2 , 2003.

[11]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 .