Scientific Research and Essays Vol. 5(12), pp. 1506-1518, 18 June, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

*Full Length Research Paper*

# Particle swarm optimization performance on special linear programming problems

**Pakize Erdoğmuş**

Department of Electronic and Computer Education, Faculty of Technical Education, Düzce University, Duzce, Türkey. E-mail: pakizeerdogmus@duzce.edu.tr. Fax: +90 380 5421134.

Accepted 19 May, 2010

**Linear programming (LP) is one of the best known optimization problems solved generally with Simplex Method. Most of the real life problems have been modeled as LP. The solutions of some special LP problems exhibiting cycling have not been studied except for the classical methods. This study, aimed to solve some special LP problems, exhibit cycling with Particle Swarm Optimization (PSO) and to show PSO performance for these problems. So, some special problems taken from literature have been solved with PSO. Results taken from Genetic Algorithm (GA) and PSO have been compared with the reference. The results have shown that PSO performance is generally better than GA in view of optimality and solution time. And it is also proposed that cycling problems are used for testing the performance of new developed algorithms like Numerical Benchmark Functions.**

**Key words:** LP, cycling, PSO.

## INTRODUCTION

Optimization problems arise in a wide variety of scientific and engineering applications including signal processing, system identification, filter design, function approximation, regression analysis, and so on. In many engineering and scientific applications, the real-time solution of optimization problems is widely required. However, traditional algorithms for digital computers may not be efficient since the computing time required for a solution is greatly dependent on the dimension and structure of the problems (Effati and Nazemi, 2006).

Linear programming (LP) is perhaps the most widely applicable technique in Operational Research (Troutt et al., 2005). Most of the real-life problem such as hydro-power reservoir optimal operation (Cheng at al., 2008) water distribution system design problems (Milan, 2010), the calculation of chemical equilibrium in complex thermo-dynamic systems (Belov, 2010), Power Allocation for Coded orthogonal frequency-division multiplexing (OFDM) (Kenarsari and Lampe, 2009) have been modeled and solved using LP approximation. LP model aims to optimize a linear object function, subject to linear equality or inequality constraints. Different solution methods have been proposed for LP.

Simplex Method is the best-known method developed by Dantzig. Revised Simplex methods have been developed by G.B. Dantzig for computer based solutions (Dantzig, 1998). There are a lot of different approaches for solving linear programming problems except for the classical methods. Simplified neural net-work (Oskoei and Mahdavi-Amiri, 2006), recurrent neural network (Malek and Alipour, 2007) and PSO (Kuo, 2009) have been used for solving linear programming problems.

Particle swarm optimization (PSO) is a population based optimization technique developed by Kennedy and Eberhart (1995), inspired by social behavior of bird flocking or fish schooling (Lazinca, 2009). Since PSO is also population based method, convergence to optimal solution is quite rapid.

The solution of LP problems exhibit cycling is difficult with classical Simplex Method. So some extra methods like Perturbation method are applied to problem, even if a cycling is a rare situation for LP models. Comparing with iterative methods, metaheuristics are thought to be successful for the cycling LP problems solutions due to their algorithms. In this study, some special LP problems exhibit cycling are selected for solving with PSO.

## LINEAR PROGRAMMING MODELS

A general LP model can be formulated as a mathematical optimization problem as Equation (1):

fopt = $\sum_{i=1}^{n} a_i x_i$

$$\sum_{i=1}^{n} b_i x_i < c_i$$

$$\sum_{i=1}^{n} d_i x_i = e_i \, ,$$

$$\sum_{i=1}^{n} f_i x_i > g_i \, , \quad X_i > 0 \quad i=1, 2,\ldots n \qquad (1)$$

Solution in a General LP problem is found with movement from the basic solution point to the more optimum basic solution point, until it reaches the best optimum point. LP problems converge to an optimal solution, according to non-degeneracy assumption (NDA) (Gass and Vinjamuri, 2004).

Finding the leaving variable, the presence of more than one candidate for leaving the basis causes degeneration. These candidates, namely basic solutions with one or more basic variables at zero are called degenerate. Simplex iterations that do not change the basic solution are also called degenerate. In some cases, after some degenerate iterations, simplex method reaches non-degenerate solution.

But sometimes Simplex method goes through an endless sequence of iterations without ever finding an optimal solution. Simplex Method repeats some iterations in a loop. The first example that was shown to cycle have been constructed by Hoffman in 1953 (Chvatal, 1983; Garc´ia and Palomo, 2003). And some techniques preventing cycling have been developed by Dantzig. Even if cycling is a rare situation in practical applications, cycling is overcome in most computer implementations of simplex method (Gass and Vinjamuri, 2004).

## PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is introduced by James Kennedy and Russell Eberhart in 1995. PSO is an evolutionary computation technique like genetic algori-thms. Since PSO have many advantages such as compa-rative simplicity, rapid con-vergence and little parameters to be adjusted, it has been used in many fields such as function optimization, neural network training, fuzzy system control and pattern identification ( Li and Xiao, 2008).

The particle swarm algorithm is an optimization technique inspired by the metaphor of social interaction observed among insects or animals. The kind of social interaction modeled within a PSO is used to guide a population of individuals (particles) moving toward the most promising area of the search space. In a PSO algorithm, each particle is a candidate solution and each particle "flies" through the search space, depending on two important fac-tors; the best position the current particle have found so far and the global best position identified from the entire population. The rate of position change of particle is given by its velocity (Clerc, 1999). k is the iterations number. Particles velocity and positions are updated according to (2), (3) and (4) equations related to the pbest and gbest values:

$$v_i^{k+1} = K(v_i^k + \varphi_1 rand()(p_{besti}{}^k - x_i^k) + \varphi_2 rand()(g_{best} - x_i^k)) \qquad (2)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \qquad (3)$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \qquad \varphi = \varphi_1 + \varphi_2 \qquad \varphi > 4 \qquad (4)$$

The velocity and position formulas with constriction factor(K) have been introduced by Maurice Clerc (Clerc, 1999) given (2),(3),(4). The constriction factor(K) pro-duces a damping effect on the amplitude of an individual particle's oscillations, and as a result, the particle will converge over time. $\varphi_1$ and $\varphi_2$ represent the cognitive and social parameters, respectively, rand is the random number uniformly distributed (Parrot and Li, 2006).

The PSO algorithm shares many similarities with evolu-tionary computation techniques such as GAs. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, the PSO algorithm have no evolutionary operators, such as crossover and mutation. In the PSO algorithm, the potential solutions, called particles, move through the problems space by following the current optimal particles (Kuo and Huang, 2009).

PSO has been used for solving discrete and continuous problems. It has been applied to a wide range of applications such as function optimization (Seo at al., 2006), Electrical Power System applications (Valle at al., 2008), neural network training (Zhang at al., 2007), task assignment and scheduling problems in Operation Research (Yoshida at al., 1999; Sevkli and Guner, 2006).
PSO is started with initial solutions belonging to each particle.

The global best solutions are selected to fitness function among initial solutions. Local best solutions for each particle is saved. Velocity and positions are updated according to the formulas. Until the stopping criteria, global best solutions and local best solutions are updated for the iterations.
The pseudo code of PSO is given as:

```
Generated initial P particle swarm

Do
    For i=1:P
        If fitness(Pi) <fitness(Gbest) then Gbest=Pi
        If  fitness(Pi) <fitness(Pbest(i)) then Pbest(i)=Pi
         Compute velocity with Formula (4)
Update  particle P with Formula (5)
    end
Until stopping criteria is not true
```

## SPECIAL LINEAR PROGRAMMING PROBLEM SOLUTION WITH PSO

Most of the real life optimization problems like LP model have constraints. The main difficulty for the optimization problems is to find a solutions in a feasible region speci-fied by constraints. Since variables are continuous, the increment of the number of constraints and variables cause feasible solutions' space to be more complex. Meta-heuristics methods has been considered to be acceptably good solvers of unconstrained continuous problems (Hedar and Fukushima, 2001).

PSO also runs according to unconstrained optimization procedure. So, the constrained continuous optimization problems has been transformed into unconstrained continuous optimization problem by penalizing the objective function value with the quadratic penalty function. In case of any violation of a constraint boundary, the fitness of corresponding solutions is penalized, and thus kept within feasible regions of the solution space by increasing the value of the objective function, when constraint violations are encountered.

The penalty coefficients($R_i$) for each constraint have to be judiciously selected. So the rea-sonable solutions importantly depend on these values of penalty coefficients (Saruhan, 2006):

$$P = \sum_{i=1}^{k} R_i \left( \max \left[ 0,\ g_i \right) \right] )^2 \qquad (5)$$

$$\text{Fmin} = \text{Fobjective} + \text{P} \qquad (6)$$

For the equality;

$$\text{gi} = \text{abs}\left(\sum_{i=1}^{n} d_i x_i - e_i\right) - \text{tol} \qquad (7)$$

In minimization problem, penalty function is added to fitness function. k is the number of constraints, $g_i$ is the constraints' result for the current variables. Tol is the tolerance value, for the equality constraints. Namely if the sum is approached to $e_i$ with an accepted tolerance, equality constraints are thought to be satisfied. If the problem variables satisfy the constraints, $g_i$ will be negative and P will be zero. Namely, if constraints are in feasible region, then P is equal to zero and if not the fitness function is penalized by P.

In this study, LP problems solved with PSO are the best known LP problems exhibiting cycling presented, respect-tively (Hoffman1953; Beale 1955; Yudin and Gol'shtein, 1965; Marshall and Suurballe, 1969; Chvatal, 1983). The paper by Gass and Vinjamuri (2004) gives explicit statements of 11 problems that are known to exhibit cycling (Gass and Vinjumari, 2004). Problems used in this study are given in Table 1.

In LP models the aim is to find best solutions satisfying the linear constraints. As observed in the Table 1 the objects of the LP models may be minimization or maximization. PSO aims to find global minimum without getting trapped by local minimums. So if the object function is a maximization problem, object function is converted to minimization problem multiplying by minus 1. Fitness function is the sum of object function and penalty function. Parameters used for solving these problems have been given in Table 2.

According to the literature, the problems given in Table 1 have optimal results except for 9 and 11. Tables 9 and 11 have unbounded solutions.

PSO algorithms for the solutions of these problems have been written in Matlab©. And for the comparison, problems have also been solved with GA. After specifying the proper parameter values, for each problem, 30 runs were simulated.

All simulations have been implemented on a personal computer with Intel Pentium Duo CPU 2.8 GHz and 1.87 GB RAM using MATLAB. The average solutions time for each problem has been calculated tic/toc commands in MATLAB. The mean and standard deviation of these simulations and best solutions for each problem have been given respectively in Table 3 - 22 comparing GA and PSO results with Reference Article. Relative errors have been calculated for the optimal values for GA and PSO with the following equation:

$$\varepsilon = \left| \frac{f_{\min^*} - f_{\min}}{f_{\min^*}} \right| \qquad (8)$$

$F_{min^*}$: Optimal solutions value given by Reference
$F_{\mathbf{min}}$: Average optimum solutions value found GA or PSO method

## Conclusions

This study aimed to show PSO performance on LP problems exhibiting cycling. So some special LP problems exhibiting cycling have been solved in both PSO and GA for the constant parameters set. As observed in Table 3 - 23, PSO is successfully applied to such problems. The performance of PSO is generally better than GA in view of optimality.

As observed in the tables, PSO showed better results for the problems, both in view of average optimal solution and standard deviation. PSO has shown quite better results in view of relative errors for the problems having optimal values different from zero. Comparing average optimal values, the results show that PSO solutions are more stable, since the standard deviation of PSO solutions for the solved problems is generally smaller than GA.

Although both PSO and GA are population based optimization methods, the solution time for the problems solved with PSO is better than GA, because of the fact that PSO uses less parameter (Table 23). This is quite important for real time implementations.

It is observed that the two of the problems' results are different given in the reference (Gass and Vinjumari, 2004) (Table 19 - 21). These results have also been found with Matlab Linprog for controlling.

As it is seen from the results that the solutions for these kind of problems are strictly dependent to the parameters. So it is suggested to use such cycling problems for testing new developed algorithms' performances as benchmark problems. Also, sensitivity of the parameters for these problems are out of scope. It is suggested to study parameter sensitivity of such special problems solutions.

**Table 1.** Special LP problems by Gass and Vinjamuri (2004).

| Problem no | Problem object function and constrained |
|---|---|
| 1 | Minimize<br>$-2.2361x_4 + 2x_5 + 4x_7 + 3.6180x_8 + 3.236x_9 + 3.6180x_{10} + 0.764x_{11}$<br>subject to<br>$x_1 = 1$<br>$x_2 + 0.3090x_4 - 0.6180x_5 - 0.8090x_6 - 0.3820x_7 + 0.8090x_8 + 0.3820x_9 + 0.3090x_{10} + 0.6180x_{11} = 0$<br>$x_3 + 1.4635x_4 + 0.3090x_5 + 1.4635x_6 - 0.8090x_7 - 0.9045x_8 - 0.8090x_9 + 0.4635x_{10} + 0.3090x_{11} = 0$ |
| 2 | Minimize $- 3/4x_1 + 150x_2 - 1/50x_3 + 6x_4$<br>subject to<br>$(1/4)x_1 - 60x_2 - (1/25)x_3 + 9x_4 + x_5 = 0$<br>$(1/2)x_1 - 90x_2 -( 1/50)x_3 + 3x_4 + x_6 = 0$<br>$x_3 + x_7 = 1$ |
| 3 | Maximize $x_3 - x_4 + x_5 - x_6$<br>subject to<br>$x_1 + 2x_3 - 3x_4 - 5x_5 + 6x_6 = 0$<br>$x_2 + 6x_3 - 5x_4 - 3x_5 + 2x_6 = 0$<br>$3x_3 + x_4 + 2x_5 + 4x_6 + x_7 = 1$ |
| 4 | Maximize $x_3 - x_4 + x_5 - x_6$<br>subject to<br>$x_1 + x_3 - 2x_4 - 3x_5 + 4x_6 = 0$<br>$x_2 + 4x_3 - 3x_4 - 2x_5 + x_6 = 0$<br>$x_3 + x_4 + x_5 + x_6 + x_7 = 1$ |
| 5 | Minimize $- 2x_4 - 3x_5 + x_6 + 12x_7$<br>subject to<br>$x_1 - 2x_4 - 9x_5 + x_6 + 9x_7 = 0$<br>$x_2 + (1/3)x_4 + x_5 -(1/3x_6 )- 2x_7 = 0$<br>$x_3 + 2x_4 + 3x_5 - x_6 - 12x_7 = 2$ |
| 6 | Minimize $2x_1 + 4x_4 + 4x_6$<br>subject to<br>$x_1 - 3x_2 - x_3 - x_4 - x_5 + 6x_6 = 0$<br>$2x_2 + x_3 - 3x_4 - x_5 + 2x_6 = 0$ |
| 7 | Minimize $- 0.4x_5 - 0.4x_6 + 1.8x_7$<br>subject to<br>$x_1 + 0.6x_5 - 6.4x_6 + 4.8x_7 = 0$<br>$x_2 + 0.2x_5 - 1.8x_6 + 0.6x_7 = 0$<br>$x_3 + 0.4x_5 - 1.6x_6 + 0.2x_7 = 0$<br>$x_4 + x_6 = 1$ |
| 8 | Minimize $- 2x_3 - 2x_4 + 8x_5 + 2x_6$<br>subject to<br>$x_1 - 7x_3 - 3x_4 + 7x_5 + 2x_6 = 0$<br>$x_2 + 2x_3 + x_4 - 3x_5 - x_6 = 0$ |

**Table 1.** Contd.

| | |
|---|---|
| 9 | Maximize 3x1 – 80x2 + 2x3 – 24x4<br>subject to<br>x1 – 32x2 – 4x3 + 36x4 + x5 = 0<br>x1 – 24x2 – x3 + 6x4 + x6 = 0 |
| 10 | Minimize 10x1 – 57x2 – 9x3 – 24x4<br>subject to<br>0.5x1 – 5.5x2 – 2.5x3 + 9x4 + x5 = 0<br>0.5x1 – 1.5x2 – 0.5x3 + x4 + x6 = 0<br>x1 + x7 = 1 |
| 11 | Maximize $-3x_2 + x_3 - 6x_4 - 4x_6$<br>subject to<br>$x_1 + x_2 + (1/3)x_5 + (1/3)x_6 = 2$<br>$9x_2 + x_3 - 9x_4 - 2x_5 - (1/3)x_6 + x_7 = 0$<br>$x_2 + (1/3)X_3 - 2x_4 - (1/3)x_5 - (1/3)x_6 + x_8 = 2$ |

**Table 2.** GA and PSO parameters used for the solution of test problems.

| Optimization method | Parameter | Value |
|---|---|---|
| GA | Population | 80 |
| | Generations | 800 |
| | Crossover | 0.8 |
| | Mutation | 0.1 |
| PSO | Particles | 80 |
| | Iteration number | 800 |
| | $\varphi_1$ | 2 |
| | $\varphi_2$ | 2.1 |
| | K | 0.7298 |

**Table 3.** Results for the Hofman problem.

| Prob No | Values | Best solution Ref (Gass and Vinjamuri, 2004) | Best solution with GA | Best solution with PSO |
|---|---|---|---|---|
| 1 | Fmin | 0 | -3E-06 | 1.88E-04 |
| | X1 | 1 | 1 | 1 |
| | X2 | 0 | 0 | 0.000484 |
| | X3 | 0 | 0 | 3.81E-05 |
| | X4 | 0 | 1.34E-06 | 0.001406 |
| | X5 | 0 | 0 | 2.94E-05 |
| | X6 | 0 | 0 | 0.001009 |
| | X7 | 0 | 0 | 2.79E-05 |
| | X8 | 0 | 0 | 1.26E-05 |
| | X9 | 0 | 0 | 9.4E-05 |
| | X10 | 0 | 0 | 7.7E-05 |
| | X11 | 0 | 0 | 1.48E-05 |
| | Iteration | 10 | - | - |

**Table 4.** Mean and standard deviation for the Hofman problem.

| Prob No | Values | Best solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 1 | Fmin | 0 | -0.00075 | 0.010944 | 0.00038 | 0.040025 |
| | X1 | 1 | 1.000036 | 1 | 0.00034 | 0 |
| | X2 | 0 | 1.04E-05 | 0.002476 | 1.4E-05 | 0.008003 |
| | X3 | 0 | 3.22E-05 | 0.00019 | 0 | 0.000337 |
| | X4 | 0 | 0.000335 | 0.004196 | 0.00017 | 0.006937 |
| | X5 | 0 | 0 | 9.52E-05 | 0 | 0.000215 |
| | X6 | 0 | 1.24E-05 | 0.000208 | 1.34E-05 | 0.000805 |
| | X7 | 0 | 0 | 0.002843 | 0 | 0.012208 |
| | X8 | 0 | 0 | 0.000243 | 0 | 0.000534 |
| | X9 | 0 | 0 | 0.001572 | 0 | 0.005884 |
| | X10 | 0 | 0 | 8.93E-05 | 0 | 0.000183 |
| | X11 | 0 | 0 | 0.000169 | 0 | 0.00036 |
| | ε | - | - | - | - | - |
| | Iteration | 10 | | | | |

**Table 5.** Results for the Beale problem.

| Prob No | Values | Best solution Ref (Gass and Vinjamuri, 2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 2 | Fmin | -0.05 | 0.0046 | -0.05246 |
| | X1 | 0.04 | 0.0026 | 0.044171 |
| | X2 | 0 | 0.0000 | -1.9E-06* |
| | X3 | 1 | 1.0006 | 1 |
| | X4 | 0 | 0.0044 | -1.3E-06* |
| | X5 | 0.03 | 0.0000 | 0.02969 |
| | X6 | 0 | 0.0063 | -1.9E-07* |
| | X7 | 0 | 0.0000 | 0 |
| | Iteration | 6 | - | - |

**Table 6.** Mean and standard deviation for the Beale problem.

| Prob No | Values | Best solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 2 | Fmin | -0.05 | 0.001808 | -0.04529 | 0.002829 | 0.109015 |
| | X1 | 0.04 | 0.004206 | 0.126893 | 0.001338 | 1.304608 |
| | X2 | 0 | 1.32E-08 | 0.000459 | 3.36E-08 | 0.007248 |
| | X3 | 1 | 0.99251 | 1 | 0.037313 | 0 |
| | X4 | 0 | 0.004135 | 1.6E-06 | 0.000455 | 6.17E-05 |
| | X5 | 0.03 | 0.001469 | 0.035843 | 0.002386 | 0.108656 |
| | X6 | 0 | 0.005828 | 2.24E-05 | 0.000585 | 0.000195 |
| | X7 | 0 | 0.008037 | 0 | 0.037386 | 0 |
| | ε | - | >1 | 0.094138 | - | - |
| | Iteration | 6 | | | | |

**Table 7.** Results for the Yudin and Gol'shtein.

| Prob No | Values | Best solution Ref (Gass and Vinjamuri,2004) | Best solution with GA | Best solution with PSO |
|---|---|---|---|---|
| 3 | Fmax | 0.5 | 0.40283 | 0.48213 |
| | X1 | 2.5 | 1.411504 | 2.076877 |
| | X2 | 1.5 | 0.380302 | 1.032365 |
| | X3 | 0 | 0.113277 | 0.046606 |
| | X4 | 0 | 0.023864 | 6.34E-06 |
| | X5 | 0.5 | 0.313417 | 0.436278 |
| | X6 | 0 | 0 | 6.9E-06 |
| | X7 | 0 | 0.010291 | 5.13E-07 |
| | iteration | 6 | - | - |

**Table 8.** Mean and Standart Deviation for the Yudin and Gol'shtein.

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 3 | Fmax | 0.5 | 0.34545 | 0.4488 | 0.069596 | 0.013882 |
| | X1 | 2.5 | 1.191161 | 1.452802 | 0.279237 | 0.254993 |
| | X2 | 1.5 | 0.292941 | 0.330382 | 0.1087 | 0.282348 |
| | X3 | 0 | 0.11502 | 0.112457 | 0.019995 | 0.026786 |
| | X4 | 0 | 0.035043 | 1.99E-05 | 0.023188 | 5.72E-05 |
| | X5 | 0.5 | 0.276161 | 0.336828 | 0.035617 | 0.040601 |
| | X6 | 0 | 0.010687 | 4.52E-05 | 0.026594 | 0.000126 |
| | X7 | 0 | 0.025672 | 0.000565 | 0.021501 | 0.001996 |
| | ε | - | 0.65455 | 0.1024 | | |
| | Iteration | 6 | | | | |

**Table 9.** Results for the Yudin and Gol'shtein(2).

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 4 | Fmax | 1 | 0.9959 | 1.0125 |
| | X1 | 3 | 1.698622 | 2.425748 |
| | X2 | 2 | 0.062865 | 1.12498 |
| | X3 | 0 | 0.321778 | 0.152485 |
| | X4 | 0 | 1.82E-06 | -2.5E-07 |
| | X5 | 1 | 0.675247 | 0.862516 |
| | X6 | 0 | 0.00112 | -2.5E-07 |
| | X7 | 0 | 0.00259 | -1.2E-07 |
| | iteration | 6 | - | - |

**Table 10.** Mean and standard deviation for the Yudin and Gol'shtein (2).

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 4 | Fmax | 1 | 0.95329 | 1.01244 | 0.105593 | 0.000275 |
| | X1 | 3 | 1.72661 | 1.824238 | 0.143779 | 0.201671 |
| | X2 | 2 | 0.175376 | 0.210569 | 0.284471 | 0.305418 |
| | X3 | 0 | 0.304624 | 0.303967 | 0.05029 | 0,050873 |
| | X4 | 0 | 0,018184 | 1,66E-05 | 0,031774 | 8,94E-05 |
| | X5 | 1 | 0,671951 | 0,710903 | 0,01708 | 0,050874 |
| | X6 | 0 | 0,005105 | -2,8E-07 | 0,027028 | 5,24E-07 |
| | X7 | 0 | 0,000828 | 2,74E-06 | 0,001827 | 1,33E-05 |
| | ε | | 0.046715 | 0.01244 | - | - |
| | Iteration | 6 | | | | |

**Table 11.** Results for the Balinski and Tucker.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 5 | Fmin | -2 | -2.001 | -2.00872 |
| | X1 | 2 | 0.5001 | 2.006841 |
| | X2 | 0 | 0.0002 | 0.002876 |
| | X3 | 0 | 1.5001 | 0.000622 |
| | X4 | 2 | 0.5001 | 1.975657 |
| | X5 | 0 | 0.0001 | 6.15E-05 |
| | X6 | 2 | 0.4997 | 1.938235 |
| | X7 | 0 | 0.0000 | -5.1E-06 |
| | iteration | 6 | - | - |

**Table 12.** Mean and standard deviation for the Balinski and Tucker.

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 5 | Fmin | -2 | -0.71389 | -1.96514 | 0.459497 | 0.110148 |
| | X1 | 2 | 1.298878 | 2.249378 | 1.854423 | 0.472153 |
| | X2 | 0 | 0.033816 | 0.012419 | 0.113663 | 0.036487 |
| | X3 | 0 | 1.286864 | 0.042577 | 0.459391 | 0.102879 |
| | X4 | 2 | 1.136697 | 2.04438 | 1.2516 | 0.311655 |
| | X5 | 0 | 0.070701 | 0.040981 | 0.208571 | 0.06966 |
| | X6 | 2 | 1.125361 | 2.102472 | 1.46674 | 0.366267 |
| | X7 | 0 | 0.053854 | 0.011585 | 0.119924 | 0.041531 |
| | ε | | 0.643055 | 0.017428 | - | - |
| | Iteration | 6 | | | | |

**Table 13.** Results for the Marshall and Suurballe.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri,2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 6 | Fmin | 0.0000 | 0.0000 | -2.2E-06 |
| | X1 | 0.0000 | 0.0000 | -2.5E-07 |
| | X2 | 0.0000 | 0.0000 | 0.00012 |
| | X3 | 0.0000 | 0.0000 | 0.001185 |
| | X4 | 0.0000 | 0.0000 | -5E-07 |
| | X5 | 0.0000 | 0.0000 | 0.004208 |
| | X6 | 0.0000 | 0.0000 | -5.1E-07 |
| | X7 | 0.0000 | 0.0000 | -2.5E-07 |
| | iteration | 6 | - | - |

**Table 14.** Mean and standard deviation for the Marshall and Suurballe.

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 6 | Fmin | 0.0000 | 0.366 | -2.2E-06 | 1.380533 | 3.49E-07 |
| | X1 | 0.0000 | 0.084538 | -2.2E-07 | 0.356788 | 1.73E-07 |
| | X2 | 0.0000 | 0.046342 | 0.000789 | 0.176358 | 0.000719 |
| | X3 | 0.0000 | 0.002212 | 0.002894 | 0.011891 | 0.002379 |
| | X4 | 0.0000 | 0.022703 | -4.9E-07 | 0.107176 | 4.07E-08 |
| | X5 | 0.0000 | 0.079811 | 0.002813 | 0.429783 | 0.002346 |
| | X6 | 0.0000 | 0.026528 | -5E-07 | 0.103081 | 2.83E-08 |
| | X7 | 0.0000 | 0.084538 | -2.2E-07 | 0.356788 | 1.73E-07 |
| | ε | - | - | - | - | - |
| | Iteration | 6 | | | | |

**Table 15.** Results for the Marshall and Suurballe (2).

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 7 | Fmin | -2.0000 | -1.26253 | -2.04151 |
| | X1 | 4.0000 | 2.761524 | 4.040216 |
| | X2 | 1.0000 | 0.892748 | 1.004056 |
| | X3 | 0.0000 | 0.11161 | 0.000398 |
| | X4 | 0.0000 | 0.012631 | 4.63E-07 |
| | X5 | 4.0000 | 3.519165 | 4.107072 |
| | X6 | 1.0000 | 0.986436 | 1.017612 |
| | X7 | 0.0000 | 0.299839 | 8.84E-06 |
| | iteration | 6 | - | - |

**Table 16.** Mean and standard deviation for the Marshall and Suurballe (2).

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 7 | Fmin | -2.0000 | -0.00072 | -1.81723 | 0.2017 | 0.212396 |
| | X1 | 4.0000 | 0.00031 | 3.594129 | 1.639638 | 0.761029 |
| | X2 | 1.0000 | 0.000108 | 0.894073 | 0.37794 | 0.200981 |
| | X3 | 0.0000 | 7.69E-05 | 0.001073 | 0.440847 | 0.221419 |
| | X4 | 0.0000 | 1.00015 | 0.101782 | 0.221794 | 0.176971 |
| | X5 | 4.0000 | 0.001546 | 3.65473 | 1.160903 | 0.662888 |
| | X6 | 1.0000 | 0.000249 | 0.905404 | 0.221631 | 0.177301 |
| | X7 | 0.0000 | 0 | 0.000755 | 0.236954 | 0.186508 |
| | ε | - | 0.99964 | 0.091386 | - | - |
| | Iteration | 6 | | | | |

**Table 17.** Results for the Solow.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri,2004) | Best Solution with GA | Best Solution with PSO |
|---|---|---|---|---|
| 8 | Fmin | 0.0000 | -0.00033 | -0.012 |
| | X1 | 0.0000 | 0.000209 | 1.031692 |
| | X2 | 0.0000 | 0.000244 | -2.5E-07 |
| | X3 | 0.0000 | 0.000126 | -2.5E-07 |
| | X4 | 0.0000 | 3.94E-05 | 1.009681 |
| | X5 | 0.0000 | 0 | -2.5E-07 |
| | X6 | 0.0000 | 0 | 0.998681 |
| | Iteration | 6 | - | - |

**Table 18.** Mean and standard deviation for the Solow

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| 8 | Fmin | 0.0000 | -0.00092 | -0.012 | 0.000401 | 3.47E-12 |
| | X1 | 0.0000 | 0.000769 | 0.787441 | 0.000763 | 0.321427 |
| | X2 | 0.0000 | 4.34E-05 | -2.5E-07 | 9.77E-05 | 2.21E-10 |
| | X3 | 0.0000 | 7.37E-05 | -2.5E-07 | 9.23E-05 | 8.65E-11 |
| | X4 | 0.0000 | 0.000396 | 0.765464 | 0.000256 | 0.321547 |
| | X5 | 0.0000 | 5.87E-10 | -2.5E-07 | 3.16E-09 | 1.14E-10 |
| | X6 | 0.0000 | 8.56E-06 | 0.754464 | 3.2E-05 | 0.321547 |
| | ε | - | - | - | - | - |
| | Iteration | 6 | | | | |

**Table 19.** Results for the Chavatal.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with MATLAB Linprog | Best Solution with MATLAB GA | Best Solution with PSO |
|---|---|---|---|---|---|
| | Fmin | 1.0000 | -245540913249* | -458.116 | -6821.39 |
| | X1 | 1.0000 | $0.0000x10^{10}$ | 0.628162 | 0.535329 |
| | X2 | 0.0000 | $3.0192\ x10^{10}$ | 6.131148 | 95.07208 |
| | X3 | 1.0000 | $0.5366\ x10^{10}$ | 1.756172 | 3.236557 |
| 10 | X4 | 0.0000 | $1.9846\ x10^{10}$ | 4.129844 | 58.58254 |
| | X5 | 2.0000 | $0.0852\ x10^{10}$ | 0.629458 | 3.666556 |
| | X6 | 0.0000 | $2.8124\ x10^{10}$ | 5.629954 | 85.34194 |
| | X7 | 0.0000 | $0.0000\ x10^{10}$ | 0.372307 | 0.966603 |
| | Iteration | 6 | *Unbounded | | |

**Table 20.** Mean and standard deviation for the Chavatal.

| Prob No | Values | Best Solution given Ref | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|
| | Fmin | 1.0000 | -223.897 | -3664.96 | 96.22793 | 1078.986 |
| | X1 | 1.0000 | 0.380679 | 0.520619 | 0.397145 | 0.269873 |
| | X2 | 0.0000 | 2.905999 | 51.46951 | 1.279992 | 15.50198 |
| | X3 | 1.0000 | 1.376416 | 2.944872 | 1.113509 | 3.796213 |
| 10 | X4 | 0.0000 | 2.069748 | 31.58738 | 0.901027 | 9.281742 |
| | X5 | 2.0000 | 0.606591 | 5.940383 | 0.493378 | 8.711723 |
| | X6 | 0.0000 | 2.786373 | 46.52708 | 1.191863 | 13.5573 |
| | X7 | 0.0000 | 0.619903 | 0.517884 | 0.397234 | 0.307233 |
| | ε | | - | - | - | - |
| | Iteration | 6 | | | | |

**Table 21.** Results for the Nering and Tucker.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with MATLAB Linprog | Best Solution with MATLAB GA | Best Solution with PSO |
|---|---|---|---|---|---|
| | Fmax | Unbounded | 12* | 17.8232 | 10.3368 |
| | X1 | | 0 | 0.115641 | 0.277586 |
| | X2 | | 0 | 0.00521 | 0.000592 |
| | X3 | | 193.9551 | 5.924058 | 10.39124 |
| 11 | X4 | | 30.3258 | 0.434357 | 0.002472 |
| | X5 | | 6 | 0.351101 | 5.194964 |
| | X6 | | 0 | 5.289198 | -4,9E-05 |
| | X7 | | 90.9775 | 0,402528 | 0,000155 |
| | X8 | | 0 | 2,767951 | 0,298196 |
| | Iteration | 6 | *Optimal | *Optimal | *Optimal |

**Table 22.** Mean and Standart Deviation for the Nering and Tucker.

| Prob No | Values | Best Solution Ref (Gass and Vinjamuri, 2004) | Best Solution with MATLAB Linprog | Mean of GA | Mean of PSO | Std. Dev. Of GA | Std. Dev. Of PSO |
|---|---|---|---|---|---|---|---|
| 11 | Fmax | Unbounded | 12* | 5.46214 | 8.13893 | 6.073661 | 0.930759 |
| | X1 | | 0 | 1.159933 | 0.649337 | 0.812075 | 0.192641 |
| | X2 | | 0 | 0.234493 | 0.001054 | 0.429035 | 0.001421 |
| | X3 | | 193.9551 | 1.919658 | 8.655093 | 2.374153 | 1.341794 |
| | X4 | | 30.3258 | 0.419526 | 0.077417 | 0.661039 | 0.143141 |
| | X5 | | 6 | 0.427237 | 4.086683 | 0.58091 | 0.575386 |
| | X6 | | 0 | 1.392029 | 0.003354 | 1.66011 | 0.005795 |
| | X7 | | 90.9775 | 1.063227 | 0.208742 | 3.014762 | 0.57001 |
| | X8 | | 0 | 2.57027 | 0.647727 | 0.824895 | 0.212848 |
| | ε | | - | - | - | - | - |
| | Iteration | 6 | | | | | |

**Table 23.** The solution time for each problem.

| Problem no | Mean of the solution time for GA (seconds) | Mean of the solution time for PSO (seconds) |
|---|---|---|
| 1 | 3.0885 | 0.0285 |
| 2 | 3.7562 | 1.4011 |
| 3 | 2.8744 | 0.0271 |
| 4 | 3.1823 | 1.4357 |
| 5 | 3.8144 | 1.4560 |
| 6 | 2.6133 | 1.8302 |
| 7 | 3.1675 | 1.2169 |
| 8 | 2.9108 | 1.8240 |
| 10 | 2.9956 | 1.8467 |
| 11 | 3.7887 | 1.4968 |

## REFERENCES


Belov G (2010).On linear programming approach for the calculation of chemical equilibrium in complex thermodynamic systems. J. Math. Chem., 47: 446-456.

Cheng CT, Wang WC, Xu DM, Chau KW (2008). Optimizing Hydropower Reservoir Operation Using Hybrid Genetic Algorithm and Chaos. Water Resour. Manage., 22: 895-909.

Chvatal V (1983). Linear Programming. New York: W. H. Freeman and Company.

Clerc M (1999). The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. Proc. Congress on Evolutionary Computation. Washington DC, pp. 1951-1957.

Dantzig G (1998). Linear Programming and Extensions. Available: http: http://books.google.com. 03-07-2009.

Effati S, Nazemi AR (2006). Neural network models and its application for solving linear and quadratic programming problems. Applied Math. Comput., 172: 305-331.

Garc´ia PG, Palomo AS (2003). On Hoffman's celebrated cycling LP example. Technical Report. Dept. Applied Mathematics. Univ. M´alaga.

Gass SI, Vinjamuri S (2004). Cycling in Linear Programming problems. Comput. Operations Res., 31: 303-311.

Hedar AR, Fukushima M (2001). Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization. Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, Japan, pp. 606-8501.

Kenarsari A, Lampe L (2009). Power Allocation for Coded OFDM via Linear Programming. IEEE Communi. Lett., 13 (12): 887-889.

Kennedy J,Eberhart RC (1995). Particle swarm optimization. Proceeding IEEE International Conference of Neural Network IV, pp. 1942-1948.

Kuo RJ, Huang CC (2009). Application of particle swarm optimization algorithm for solving bi-level linear programming problem. Comput. Math. Appl., 58: 678-685.

Lazinca A (2009). Particle swarm optimization. Available: http://intechweb.org, 03-07-2009.

Li J, Xiao X (2008). Multi-Swarm and multi-best particle swarm optimization algorithm. Proceedings of the 7th World Congress on Intelligent Control and Automation, China. pp. 6281-6286.

Malek A, Alipour M (2007). Neural network models and its application for solving linear and quadratic programming problems. Appl. Math. Comput., 192: 27-39.

Milan C (2010). Hybrid Genetic Algorithm and Linear Programming Method for Least-Cost Design of Water Distribution Systems. Water Resour. Manage., 24: 1-24.

Oskoei-Ghasabi H, Mahdavi-Amiri N (2006). An efficient simplified neural network for solving linear and quadratic programming problems. Appl. Math. Comput. 175: 452-464.

Parrott D, Li X (2006). Locating and tracking multiple dynamic optima

by a particle swarm model using speciation. IEEE Trans. Evolutionary Comput., 10(4): 440-458.
Saruhan H (2006). Optimum design of rotor-bearing system stability performance comparing an evolutionary algorithm versus a conventional method. Int. J. Mech. Sci., 48(12): 1341-1351.
Seo JH, Im CH, Heo CG,Kim JK, Jung HK, Lee CG(2006). Multimodal Function Optimization Based on Particle Swarm Optimization. IEEE Transactions On Magnetics, 42(4): 1095-1098.
Sevkli M, Guner AR (2006). A continuous particle swarm optimization algorithm for uncapacitated facility location problem. Lecture Notes in Computer Sciences. Springer Berlin, 4150: 316-323.
Troutt MD, Tadisina SK, Sohn C, Brandyberry AA (2005). Linear programming system identification, Eur. J. Oper. Res., 161: 663-672.
Valle Y, Digman M, Gray A, Perkel J, Venayagamoorthy GK, Harley RG (2008). Enhanced Particle Swarm Optimizer for Power System Applications. IEEE 2008 IEEE Swarm Intelligence Symposium St. Louis MO USA.
Yoshida H, Kawata K, Fukuyama Y, Nakanishi Y (1999). A particle swarm optimization for reactive power and voltage control considering voltage stability. IEEE International Conference on Intelligent System Applications to Power Systems. Rio de Janeiro.
Zhang JR, Zhang J, Lok TM, Lyu MR (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Appl. Math. Comput., 185: 1026-1037.