A Population-Based, Parent Centric Procedure for Constrained Real-Parameter Optimization

Ankur Sinha, Aravind Srinivasan and Kalyanmoy Deb
Kanpur Genetic Algorithms Laboratory (KanGAL)
Indian Institute of Technology Kanpur
Kanpur, PIN 208 016, India
Email: {ankurs,aravinds,deb}@iitk.ac.in

Abstract—Despite the existence of a number of procedures for constrained real-parameter optimization using evolutionary algorithms, there is still the need for a systematic and unbiased comparison of different approaches on a carefully chosen set of test problems. In this paper, we suggest a parent centric procedure for constrained real-parameter optimization. The algorithm so developed is applied to a set of 24 test problems and the results are presented. The proposed procedure is able to find the exact optimum within the specified number of function evaluations for 22 of the 24 test problems. In the remaining two problems, the proposed algorithm shows steady progress towards the respective optima, but it was unable to solve within the specified number of evaluations. It is also noteworthy that the algorithm was able to find solutions, better than the ones specified in the original problem description (http://www.ntu.edu.sg/home/EPNSugan/) for a number of test problems.

I. INTRODUCTION

This paper is written for the special session devoted to comparing different constrained real-parameter optimization methods on a set of 24 test problems. In this paper, we employ a population-based, steady-state optimization algorithm for the purpose. The algorithm is developed based on adaptation of a population-based algorithm-generator [1]. The generator requires specification of four plans of the optimization process: (i) selection plan, (ii) generation plan, (iii) replacement plan and (iv) update plan. These plans are designed based on essential aspects needed in solving various optimization problems, such as importance of diversity preservation and need for creation of offspring solutions based on diversity in parent solutions (also used in other successful real parameter optimization schemes, such as evolution strategy [2], [4] and differential evolution [5]).

The test problems involve equality and inequality constraints, linear and non-linear constraints, low to high dimensionality etc. It is our intuition that to solve such wide variety of test problems to a reasonable level of satisfaction, the algorithm has to be simple and not specifically designed to solve a particular problem. In the following section, we describe the proposed procedure. In Section 5, we present the simulation results in tabular form and in Section 8, we discuss the performance of our algorithm on different test problems.

II. DESCRIPTION OF THE ALGORITHM

The optimization algorithm used here is derived from the population-based algorithm generator suggested elsewhere [1]. The algorithm-generator requires four plans to be specified and generates a steady-state optimization procedure:

1) Selection Plan (SP): Strategy used to select a fixed number of parents for recombination from the current population.

2) Generation Plan (GP): Methodology used to create offspring solutions from parents chosen in the selection plan.

3) Replacement Plan (RP): Strategy used to select a fixed number of members from population that will compete with the newly generated offspring solutions for inclusion in the population.

4) Update Plan (UP): Strategy used to decide the winners from a set consisting of offspring solutions and members obtained from replacement plan that will eventually get included in the current population.

The above division of an algorithm into different plans allows one to design each essential feature of an optimization task independently. With a set of population members, the first task (SP) is to choose a set of good solutions (parents) so that they can be utilized to create new solutions in GP. The use of a suitable probability distribution around parent solutions to create new offspring solutions would be one way to implement a GP. Once the offspring solutions are created they can be accepted in the fixed-size population by first choosing a set of possible population members for deletion using a RP and then designing a scheme for updating the population in UP. This is where an elite preserving strategy can be implemented.

The equality constraints \( h(\vec{x}) = 0 \), as suggested, have been transformed into inequality constraints of the form \(|h(\vec{x})| - \epsilon \leq 0\) and a solution is regarded as feasible if the following conditions are satisfied:

\[
g_j(\vec{x}) \leq 0 \quad \text{for} \quad j = 1, 2, ..., q \quad (1)
\]

\[
|h(\vec{x})| - \epsilon \leq 0 \quad \text{for} \quad j = q + 1, q + 2, ..., m \quad (2)
\]

It is well known that the equality constraints are tough to satisfy as they provide a narrow window to the algorithm to operate. To avoid this, equality constraints are handled...
here in such a way that initially the size of the window is larger and slowly it narrows down to the required accuracy as the number of function evaluations increases. The following function has been used to define the \( \epsilon \) value during intermediate function evaluations. This value is referred as \( \epsilon_t \) where \( t \) denotes the number of function evaluations.

\[
\epsilon_t = \epsilon (1 + 1000e^{-200(t/t_{\text{max}})})
\]  

(3)

It is clear that when \( t = t_{\text{max}} \) then \( \epsilon_t = \epsilon \), so after maximum function evaluations, only those solutions would be feasible which satisfy the equality constraint with an accuracy of \( \epsilon \). To begin with, the window would be larger and initially the algorithm would not have any problem in exploring the search space. The values in the above equation have been intuitively adjusted just to meet the requirements but this can be replaced with even efficient parameters. By introducing this equation in the algorithm we are enhancing the exploration of the search space for problems having equality constraints and it makes the algorithm less susceptible to quickly loose its diversity or get caught in a local minima, hence increasing its reliability.

The constraint violation is the summation of the violations of all the equality and inequality constraints. A solution \( x^i \) is said to 'constraint dominate' \( 9 \) a solution \( x^j \) if any of the following conditions are true:

1) Solution \( x^i \) is feasible and solution \( x^j \) is not.
2) Solution \( x^i \) and \( x^j \) are both infeasible but solution \( x^i \) has a smaller constraint violation.
3) Solution \( x^i \) and \( x^j \) are both feasible but the objective value of \( x^i \) is less than that of \( x^j \).

The algorithm starts with an initial population (generated randomly) of size \( N \). In the Selection Plan, stochastic remainder selection \( 10 \) is used to form a pool from which index parents are selected for crossover. To create an offspring, a parent is chosen from the pool which becomes the index parent and the other two parents are chosen from the population randomly. We shall describe this crossover operator (PCX) a little later.

In stochastic remainder selection, the fitness assignment scheme is done in the following manner by keeping in mind the above three constraint handling principles:

1) If all the members of the population are infeasible then the fitness to a member \( x^i \) is assigned as

\[
\text{fitness}_{x^i} = 1 - \frac{v_{x^i} - v_{\text{min}}}{v_{\text{max}} - v_{\text{min}}}
\]  

(4)

\( v_{x^i} \) = summation of the violations of member \( x^i \)
\( v_{\text{min}} \) = minimum violation among all the members
\( v_{\text{max}} \) = maximum violation among all the members

2) Otherwise

For an infeasible member

\[
\text{fitness}_{x^i} = 1 - \frac{(t + v_{x^i}) - o_{\text{min}}}{o_{\text{max}} - o_{\text{min}}}
\]  

(5)

For a feasible member

\[
\text{fitness}_{x^i} = 1 - \frac{o_{x^i} - o_{\text{min}}}{o_{\text{max}} - o_{\text{min}}}
\]  

(6)

\( t = \) worst objective value among feasible members
\( o_{\text{min}} = \) best (minimum) objective value among feasible members.
\( o_{\text{max}} = \) worst (maximum) objective value among feasible members.
\( o_{x^i} = \) objective value of the member \( x^i \).

The above fitness evaluation scheme assigns a higher value to a better solution.

After these fitness values are computed, they are degraded by a niching procedure which does not use any niching parameter. The fitness degradation scheme is identical to that of the sharing function method \( 12 \), except that the niche radius \( \sigma_{\text{share}} \) is not a problem parameter, rather is set by an adaptive procedure. The extent of the search space dictated by the population-minimum and maximum values is divided into exactly \( N \) small hypercubes and the size of the hypercube is defined as \( \sigma_{\text{share}} \).

In the Generation Plan, we create an offspring solution from the chosen parent solutions. We use the parent-centric recombination (PCX) operator \( 6 \) with modification for the purpose of recombination and produce an offspring. Here a planar version of PCX operator has been used, where \( 3 \) parents are used to create a child on the plane formed by the three parents. The following is the recombination operator which is used to create a child from the parents.

\[
\vec{C} = \vec{X}_p + \omega_\xi \vec{D} + \omega_\eta \frac{\vec{P}_2 - \vec{P}_1}{2}
\]  

(7)

The terms used in the above equation are defined as follows:

- \( \vec{X}_p \in \mathcal{B} \) is index parent
- \( \vec{G} \) is the mean of \( \mu \) parents
- \( \vec{D} = \vec{X}_p - \vec{G} \)
- \( \vec{P}_1 \) and \( \vec{P}_2 \) are parents chosen randomly from the population
- \( \omega_\xi \) and \( \omega_\eta \) are the two parameters which are updated by the one-fifth success rule \( 11 \). It starts with unity and its value is decreased by a factor \( c_d = 0.817 \) if the success rate in the generation is less than \( 1/5 \) otherwise it is increased by a factor \( c_d^{-1} = 1/0.817 \). If the value of the parameters become less than 0.5 then it is again assigned a value of 1.0.

In each generation \( \lambda \) children are created. After the generation plan, next we use the Replacement Plan to choose \( r \) solutions from the population. In the present scheme, we choose these solutions randomly from the entire population. We then form a pool (of size \( \lambda + r \)) consisting of \( r \) solutions chosen from the population by the replacement plan and \( \lambda \) newly created offspring solutions by the generation plan. The current population is then updated using the Update Plan, in
which we replace the $r$ solutions chosen in the replacement plan by the best $r$ solutions of the pool. This operation ensures an elite-preservation strategy. The best ones are chosen by the constraint domination principle described above.

In performing a single generation of above mentioned procedure, we have exhausted $\lambda$ function evaluations (same as the number of offspring solutions produced). The iteration continues until a prescribed number of function evaluations is achieved or a pre-defined termination criterion is met. We use the polynomial mutation [8] as the mutation operator with a mutation probability $p_m = 1/n$, where $n$ is the number of real variables.

III. PC CONFIGURATION

- System: RedHat Linux 9.0 (i386 GNU/Linux)
- CPU: P-III 1.0 GHz
- RAM: 256 MB
- Language: ANSI-C
- Compiler Used: GCC version-3.2.2

IV. PARAMETER SETTING

1) Population Size: $N = 100$. For problem 2, 100 population members does not provide a high success rate, although a steady progress towards the optimum is observed. Here, for problem 2, we use $N = 200$.

2) Number of offsprings created in each generation: $\lambda = 10$

3) Number of parent solutions selected randomly for replacement: $r = 2$

V. RESULTS OBTAINED

The results achieved have been presented in Tables I, II, III, IV, and V. The first four tables show the function error values, $(f(x) - f(x^*))$, for all the functions at $5 \times 10^5$, $5 \times 10^4$, and $5 \times 10^3$ function evaluations. 25 runs were done for each test problem and the best, median and worst results have been presented along with the mean value and the standard deviation. In the table, $c$ represents the number of violated constraints at the median solution. The sequence of three numbers indicate the number of constraint violations greater than 1.0, 0.01 and 0.0001 respectively. $\bar{c}$ is the mean value of the violations of all the constraints at the median solution. The numbers in the parenthesis after the error value of the best, median and worst solution indicates the number of constraints which are not satisfied at the corresponding location.

Tables I, II, III, and IV show some negative error values which means that the objective value was better than the best value because of the violation of the constraints. Moreover in some of the problems the best value obtained is better than the values suggested in the special session. This would make the final error values negative. But to avoid confusion the final errors have been reported as 0.0000 in those cases.

Table V shows the number of function evaluations needed to achieve the fixed accuracy level $(|f(x) - f(x^*)| < 0.0001)$, success rate, feasible rate and success performance. The mean and the standard deviation for the number of function evaluations has also been presented.

VI. CONVERGENCE MAP

The convergence plots for the median run of the total runs and with a termination criterion of maximum function evaluations (500,000) for each test problem are shown in Figures 1, 2, 3 and 4. The semi-log graphs show $\log_{10}(|f(x) - f(x^*)|)$ vs FES and $\log_{10}(\tau)$ vs FES for each problem. Points, with error values very close to zero or negative, could not be shown in the graph as the logarithm of the function is not defined for values less than or equal to zero. Problem 19, where a feasible solution was obtained while generating the initial population does not appear in the $\log_{10}(\tau)$ vs FES graph as $\tau$ is always zero in such a case and its logarithm could not be produced. For other problems also the graph ends as soon as the error value becomes zero.

VII. DISCUSSION OF RESULTS

Objective values better than the ones provided have been obtained in problems 3, 4, 5, 6, 8, 10, 11, 13, 14, 15, 16, 17, 18, 19, 21, 23, and 24. In case of the problems having equality constraints this can be attributed to the $\epsilon$ window which has been provided to the algorithm to operate. Out of the above mentioned 17 problems the problems having equality constraints are 3, 5, 11, 13, 14, 15, 21, and 23. On all other problems the algorithm has been able to produce results better than that available.

Relatively high values of $\omega_k$ and $\omega_m$ have been taken in order to span the entire search space. Smaller values would ensure a faster convergence. The parameters used in the algorithm are very simple to understand and can be effectively put to use. However, default parameters have been used for the entire results presented here, and the algorithm is able to successfully solve the test problems with a good success rate.

VIII. ALGORITHM COMPLEXITY

Table VI shows the complexity of the algorithm. Here $T_0$ is the time (in milli-seconds) taken to execute the provided sample program. $T_1$ is the average computing time (in milliseconds) per problem for 10,000 evaluations of the given 24 test problems. $T_2$ is the average computing time (in milliseconds) per problem for the algorithm with 10,000 evaluations for the given 24 test problems.

IX. CONCLUSIONS

In this paper we have presented a parent centric [1], population based procedure for constrained optimization. 24 different benchmark test problems have been tried and the algorithm has been successful in solving most of the problems. The algorithm has been developed with a simple approach which makes it robust to handle a wide variety of problems. Extensive simulation results have demonstrated that the algorithm has been successful in attaining the desired
### Table I

**Error Values Achieved When FES = 5 × 10³, FES = 5 × 10⁴, FES = 5 × 10⁵ for Problems 1-6.**

<table>
<thead>
<tr>
<th>FES</th>
<th>g01</th>
<th>g02</th>
<th>g03</th>
<th>g04</th>
<th>g05</th>
<th>g06</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 × 10³</td>
<td>5.7195(0)</td>
<td>3.24(0)</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td>5.03(0)</td>
</tr>
<tr>
<td>Best</td>
<td>5.72(0)</td>
<td>3.24(0)</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td>5.03(0)</td>
</tr>
<tr>
<td>Median</td>
<td>3.24(0)</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td>5.03(0)</td>
<td></td>
</tr>
<tr>
<td>Worst</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td>5.03(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>3.24(0)</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Std</td>
<td>3.24(0)</td>
<td>5.05(0)</td>
<td>5.09(0)</td>
<td>5.04(0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FES</th>
<th>g10</th>
<th>g11</th>
<th>g12</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 × 10³</td>
<td>5.0(0)</td>
<td>0.0(0)</td>
<td>5.0(0)</td>
</tr>
<tr>
<td>Best</td>
<td>5.0(0)</td>
<td>0.0(0)</td>
<td>5.0(0)</td>
</tr>
<tr>
<td>Median</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
</tr>
<tr>
<td>Worst</td>
<td>5.0(0)</td>
<td>0.0(0)</td>
<td>5.0(0)</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
</tr>
<tr>
<td>Std</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
<td>0.0(0)</td>
</tr>
</tbody>
</table>

### Table II

**Error Values Achieved When FES = 5 × 10³, FES = 5 × 10⁴, FES = 5 × 10⁵ for Problems 7-12.**

### Figure 1

Convergence Graph for Problems 1-6
### Table III
#### Error Values Achieved When FES = $5 \times 10^3$, FES = $5 \times 10^4$, FES = $5 \times 10^5$ for Problems 13-18.

<table>
<thead>
<tr>
<th>FES</th>
<th>g13</th>
<th>g14</th>
<th>g15</th>
<th>g16</th>
<th>g17</th>
<th>g18</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 10^3$</td>
<td>Best 0.0382(3)</td>
<td>-272.4364(3)</td>
<td>-0.0066(2)</td>
<td>0.0356(0)</td>
<td>-14.2458(4)</td>
<td>0.2437(0)</td>
</tr>
<tr>
<td>Median 0.9258(3)</td>
<td>-179.37(13)</td>
<td>1.0686(2)</td>
<td>0.0911(0)</td>
<td>0.0590(4)</td>
<td>0.4592(0)</td>
<td></td>
</tr>
<tr>
<td>Worst 2.7296(3)</td>
<td>-134.7809(3)</td>
<td>0.0079(2)</td>
<td>0.1870(0)</td>
<td>335.9880(4)</td>
<td>0.844260</td>
<td></td>
</tr>
<tr>
<td>$\sigma$ 0.33</td>
<td>3.33</td>
<td>0.3</td>
<td>3.4</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\bar{\sigma}$ 0.1069</td>
<td>5.9201</td>
<td>-0.0080</td>
<td>0.0000</td>
<td>0.6642</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean 1.0217</td>
<td>-188.0528</td>
<td>2.5392</td>
<td>0.0421</td>
<td>108.3202</td>
<td>0.1471</td>
<td></td>
</tr>
<tr>
<td>Std 0.0173</td>
<td>36.3390</td>
<td>2.3592</td>
<td>0.0421</td>
<td>108.3202</td>
<td>0.1471</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FES</th>
<th>g20</th>
<th>g21</th>
<th>g22</th>
<th>g23</th>
<th>g24</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 10^3$</td>
<td>Best 493.9014(0)</td>
<td>4.9451(20)</td>
<td>228.1415(5)</td>
<td>554.1268(19)</td>
<td>81.8483(4)</td>
</tr>
<tr>
<td>Median 848.0931(0)</td>
<td>8.3784(20)</td>
<td>475.8560(5)</td>
<td>808.0194(19)</td>
<td>148.1150(4)</td>
<td></td>
</tr>
<tr>
<td>Worst 1494.5490(0)</td>
<td>11.6830(20)</td>
<td>790.2094(5)</td>
<td>1824.1313(19)</td>
<td>390.7865(5)</td>
<td></td>
</tr>
<tr>
<td>$\sigma$ 0.0024</td>
<td>0.1220</td>
<td>0.44</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>$\bar{\sigma}$ 0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean 1.7205</td>
<td>0.7935</td>
<td>0.5002</td>
<td>0.0000</td>
<td>80.8104</td>
<td>0.0490</td>
</tr>
<tr>
<td>Std 0.1730</td>
<td>0.335</td>
<td>0.0421</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FES</th>
<th>g25</th>
<th>g26</th>
<th>g27</th>
<th>g28</th>
<th>g29</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 10^4$</td>
<td>Best 15.9795(1)</td>
<td>1.0580(19)</td>
<td>0.000000</td>
<td>266.6223(19)</td>
<td>36.5234(24)</td>
</tr>
<tr>
<td>Median 32.3321(0)</td>
<td>2.1397(20)</td>
<td>0.000000</td>
<td>10416.2777(19)</td>
<td>145.5704(41)</td>
<td></td>
</tr>
<tr>
<td>Worst 99.2407(0)</td>
<td>4.8240(20)</td>
<td>0.000000</td>
<td>19852.9288(19)</td>
<td>386.9147(41)</td>
<td></td>
</tr>
<tr>
<td>$\sigma$ 0.0000</td>
<td>0.55</td>
<td>0.3</td>
<td>0.4</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>$\bar{\sigma}$ 0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean 31.0120</td>
<td>2.995</td>
<td>0.0000</td>
<td>10560.5451</td>
<td>191.4602</td>
<td></td>
</tr>
<tr>
<td>Std 10.7445</td>
<td>1.806</td>
<td>0.0000</td>
<td>4392.3507</td>
<td>150.0816</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FES</th>
<th>g30</th>
<th>g31</th>
<th>g32</th>
<th>g33</th>
<th>g34</th>
</tr>
</thead>
<tbody>
<tr>
<td>$5 \times 10^5$</td>
<td>Best 0.0000(0)</td>
<td>0.0131(9)</td>
<td>0.000000</td>
<td>2191.7665(13)</td>
<td>0.000000</td>
</tr>
<tr>
<td>Median 0.000001</td>
<td>0.0604(12)</td>
<td>0.000000</td>
<td>1980.5809(15)</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>Worst 0.000001</td>
<td>0.1223(10)</td>
<td>0.000000</td>
<td>1847.3220(11)</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>$\sigma$ 0.0000</td>
<td>0.27</td>
<td>0.0000</td>
<td>5.66</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>$\bar{\sigma}$ 0.0000</td>
<td>0.289</td>
<td>0.0000</td>
<td>11.1027</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Mean 0.0000</td>
<td>0.0991</td>
<td>0.0000</td>
<td>1305.7552</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>Std 0.0000</td>
<td>0.219</td>
<td>0.0000</td>
<td>4447.3325</td>
<td>0.0000</td>
<td></td>
</tr>
</tbody>
</table>

### Table IV
#### Error Values Achieved When FES = $5 \times 10^3$, FES = $5 \times 10^4$, FES = $5 \times 10^5$ for Problems 19-24.

---

![Fig. 2. Convergence Graph for Problems 7-12](#)

---

243
### TABLE V
NUMBER OF FES REQUIRED TO ACHIEVE A GIVEN ACCURACY LEVEL FOR PROBLEMS 1 – 24

<table>
<thead>
<tr>
<th>Prob</th>
<th>1&lt;sup&gt;st&lt;/sup&gt;</th>
<th>13&lt;sup&gt;th&lt;/sup&gt;</th>
<th>25&lt;sup&gt;th&lt;/sup&gt;</th>
<th>Mean</th>
<th>Std</th>
<th>Feas. Rate</th>
<th>Succ. Rate</th>
<th>Succ. Perf</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>32420</td>
<td>41340</td>
<td>62026</td>
<td>55204.44</td>
<td>5420.82</td>
<td>100</td>
<td>100</td>
<td>55204.44</td>
</tr>
<tr>
<td>g2</td>
<td>52900</td>
<td>88463</td>
<td>500000</td>
<td>87900.00</td>
<td>12310.24</td>
<td>100</td>
<td>64</td>
<td>127900.00</td>
</tr>
<tr>
<td>g3</td>
<td>28330</td>
<td>33590</td>
<td>47300</td>
<td>34937.00</td>
<td>2202.33</td>
<td>100</td>
<td>100</td>
<td>34937.00</td>
</tr>
<tr>
<td>g4</td>
<td>25310</td>
<td>31270</td>
<td>40140</td>
<td>30989.00</td>
<td>1891.00</td>
<td>100</td>
<td>100</td>
<td>30989.00</td>
</tr>
<tr>
<td>g5</td>
<td>29740</td>
<td>54740</td>
<td>125000</td>
<td>94785.00</td>
<td>7568.73</td>
<td>100</td>
<td>100</td>
<td>94785.00</td>
</tr>
<tr>
<td>g6</td>
<td>22560</td>
<td>35700</td>
<td>36180</td>
<td>33221.19</td>
<td>771.00</td>
<td>100</td>
<td>100</td>
<td>33221.19</td>
</tr>
<tr>
<td>g7</td>
<td>79200</td>
<td>91550</td>
<td>258840</td>
<td>117121.00</td>
<td>7733.66</td>
<td>100</td>
<td>100</td>
<td>117121.00</td>
</tr>
<tr>
<td>g8</td>
<td>1710</td>
<td>3030</td>
<td>3510</td>
<td>2826.00</td>
<td>228.00</td>
<td>100</td>
<td>100</td>
<td>2826.00</td>
</tr>
<tr>
<td>g9</td>
<td>32100</td>
<td>44400</td>
<td>58700</td>
<td>46527.00</td>
<td>4057.66</td>
<td>100</td>
<td>100</td>
<td>46527.00</td>
</tr>
<tr>
<td>g10</td>
<td>71700</td>
<td>92110</td>
<td>109970</td>
<td>89028.00</td>
<td>5487.33</td>
<td>100</td>
<td>100</td>
<td>89028.00</td>
</tr>
<tr>
<td>g11</td>
<td>15840</td>
<td>31420</td>
<td>46620</td>
<td>38688.00</td>
<td>3727.33</td>
<td>100</td>
<td>100</td>
<td>38688.00</td>
</tr>
<tr>
<td>g12</td>
<td>3710</td>
<td>8630</td>
<td>11940</td>
<td>8960.00</td>
<td>640.00</td>
<td>100</td>
<td>100</td>
<td>8960.00</td>
</tr>
<tr>
<td>g13</td>
<td>34310</td>
<td>45948</td>
<td>77946</td>
<td>53735.00</td>
<td>5425.00</td>
<td>100</td>
<td>100</td>
<td>53735.00</td>
</tr>
<tr>
<td>g14</td>
<td>37900</td>
<td>58230</td>
<td>84340</td>
<td>59237.00</td>
<td>925.00</td>
<td>100</td>
<td>100</td>
<td>59237.00</td>
</tr>
<tr>
<td>g15</td>
<td>25550</td>
<td>35940</td>
<td>91400</td>
<td>46936.00</td>
<td>6462.00</td>
<td>100</td>
<td>100</td>
<td>46936.00</td>
</tr>
<tr>
<td>g16</td>
<td>19130</td>
<td>30680</td>
<td>36790</td>
<td>30395.00</td>
<td>2451.66</td>
<td>100</td>
<td>100</td>
<td>30395.00</td>
</tr>
<tr>
<td>g17</td>
<td>88410</td>
<td>129483</td>
<td>188072</td>
<td>136110.00</td>
<td>3670.00</td>
<td>100</td>
<td>100</td>
<td>136110.00</td>
</tr>
<tr>
<td>g18</td>
<td>48350</td>
<td>62350</td>
<td>96180</td>
<td>70027.00</td>
<td>4559.00</td>
<td>100</td>
<td>100</td>
<td>70027.00</td>
</tr>
<tr>
<td>g19</td>
<td>75210</td>
<td>145762</td>
<td>187734</td>
<td>129676.00</td>
<td>5908.00</td>
<td>100</td>
<td>100</td>
<td>129676.00</td>
</tr>
<tr>
<td>g20</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>g21</td>
<td>31150</td>
<td>40850</td>
<td>46868</td>
<td>38216.66</td>
<td>426.34</td>
<td>100</td>
<td>100</td>
<td>38216.66</td>
</tr>
<tr>
<td>g22</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>500000</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>g23</td>
<td>66300</td>
<td>157420</td>
<td>211464</td>
<td>167118.88</td>
<td>6533.27</td>
<td>100</td>
<td>100</td>
<td>167118.88</td>
</tr>
<tr>
<td>g24</td>
<td>4800</td>
<td>9800</td>
<td>13090</td>
<td>11646.00</td>
<td>681.33</td>
<td>100</td>
<td>100</td>
<td>11646.00</td>
</tr>
</tbody>
</table>

![Fig. 3. Convergence Graph for Problems 13-18](image1)

![Fig. 4. Convergence Graph for Problems 19-24](image2)
accuracy in most of the problems. In the problems where desired accuracy was not met the algorithm showed a steady improvement in the objective function value. The algorithm also turns out to be reliable in most of the cases with a 100% success rate.

REFERENCES


<table>
<thead>
<tr>
<th>$T_0$</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$(T_2 - T_1)/T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>23</td>
<td>1216</td>
<td>2.386</td>
</tr>
</tbody>
</table>

TABLE VI

Computational Complexity (Time in milliseconds)