A Directed Genetic Algorithms for Treating Linear Programming Problems

Dr. Bayoumi M.A.H. & El-Feky E.Z.

Department of OR&DSS, Faculty of Computers and Information, Cairo University, Egypt.

E-mail: bayoumi2000@operamail.com
E-mail: ehab_zaky@hotmail.com

Abstract

Most modifications of Genetic Algorithms (GAs) for solving optimization problems are to find some mathematical support to the GAs. This paper introduces a new GAs called Directed Genetic Algorithms (DGAs) that will be moved towards the optimal solution if it exists, for the given problem. These GAs at each generation know how far it is from the optimal solution.

Keywords: Linear Programming, Duality, Genetic Algorithms.

1. Introduction

There are several techniques to treat mathematical optimization problems. Most of them have several conditions and assumptions, which often make the operation of applying them difficult, but it has an advantage, that is, once these conditions and assumptions are satisfied, the optimal solution will be in hand.

Genetic Algorithms (GAs) is one of these techniques.

2. Genetic Algorithms

GAs are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival.

The idea behind all GAs is to do what nature does, that best one has high

probability to survive, and most badly has low probability to survive.

The idea behind genetic algorithms is to do what nature does. Let us take rabbits as an example; at any given time there is population of rabbits. Some of them are faster and smart than other rabbits. These faster, smarter rabbits are less likely to be eaten by foxes, and therefore more of them survive to do what rabbits do best.

Of course, some of the slower, dumber rabbits survive just because they are lucky. This surviving population of rabbits starts breeding. The breeding results in a good mixture of rabbit genetic material: some slow rabbits breed with fast rabbits, some fast with fast, some smart with dumb rabbits, and so on. And on the top of that, nature throws in a ‘wild hare’ every once in a while by mutating some of the rabbit genetic material. The resulting rabbits will (on average) be faster and smarter than those in the original population, this because more faster, smarter parents survived. (It is good thing that the foxes are undergoing similar process, otherwise the rabbits might become too fast and smart for the foxes to catch any of them) [7].

See Flow Chart of the basic GAs in Appendix A.

3. **Problem Formulation.**

For the following Linear Programming problem P.

\[ \text{P: Minimize } z(X) = CX, \]
\[ \text{Subject to:} \]
\[ AX \geq b, \]
\[ X \geq 0. \]

It has \( X \) as an optimal solution that gives the lower bound for the objective function \( z \).

On the other side, the dual of this problem has the following form D:

\[ \text{D: Maximize } y(W) = Wb, \]
\[ \text{Subject to:} \]
\[ WA \leq C, \]
\[ W \geq 0. \]

Where:
\[ A_{mxn}, b_{nx1}, X_{nx1}, W_{mx1}, C_{nx1}, \]
\( m \) number of constraints & \( n \) number of decision variables.

This problem \( D \) has \( W^* \) as an optimal solution that gives the upper bound for the objective function \( y \).

*Note:* there is exactly one dual variable for each primal constraint, and exactly one dual constraint for each primal variable.

The definition we have selected for the dual problem leads to many important relationships between the primal and dual linear programs. Let \( X_0 \) and \( Y_0 \) be any feasible solutions to the primal and dual programs respectively. Then:

\[ CX_0 \geq Y_0 b \]

Which is known as the “**weak duality theorem**” [5]. i.e the objective function value for any feasible solution to the minimization problem is always greater than or equal to the objective function value for any feasible solution to the maximization problem.

In particular, the objective value for any feasible solution to the minimization problem gives an upper bound on the optimal objective value of the maximization problem. Similarly, the objective value of any feasible solution of the maximization problem is a lower bound on the optimal objective value of the minimization problem.
Corollary

If $X_0$ and $Y_0$ are feasible solutions to the primal and dual problems such that

$$CX_0 = Y_0 b,$$

Then $X_0$ and $Y_0$ are optimal solutions to their respective problem [5].

Theorem “Strong Duality Theorem”

If one problem possesses an optimal solution, then both problems possess optimal solutions and the two optimal objective values are equal [5].

4. Computational Implementation

In this new treating, the main aim is to direct the GAs how far from the optimal solution. This can be done by formulating the LP problem called $T_{BH}$ which has the features of both problems $P$ and $D$:

$$\begin{align*}
T_{BE}: \text{Minimize } & N_{BE}(X,W), \\
\text{Subject to: } & \quad N_{BE} = CX - Wb, \\
& \quad AX \geq b, \\
& \quad WA \leq C, \\
& \quad X, W \geq 0.
\end{align*}$$

By this formulation and the theorem mentioned in the previous section we found the optimal solution for both problem $P$ and $D$ at one shot, when $N_{BH}$ vanishes. And as this value increases as the gap between the solutions of problems $P$ & $D$, so we can use this property in GAs when it gets solution to know how this solution is good? And how it is far from the optimal solution?

Theorem

$(X^*, W^*)$ is an optimal solution of the triality problem $T_{BH}$ if and only if $X^*$ and $W^*$ are the optimal solutions for the Primal and the Dual problems respectively.

Proof

We regard the triality problem $T_{BH}$ as a two group of variables $X$ and $W$, for both the objective and system of constraints so, we can solve the 1st group of variable $X$ independent of the 2nd group of variable $W$ and then the solution $X^*$ is an optimal for the 1st problem $P$ and the solution $W^*$ is an optimal for its dual problem $D$ and vice versa.

Here we introduce a simple example to clarify the given idea.

5. Illustrative Example

For the following Linear Programming problem $P$:

$$\begin{align*}
P: \text{Minimize } & z(X) = 6x_1 + 6x_2, \\
\text{Subject to: } & \quad 2x_1 + x_2 \geq 1, \\
& \quad x_1 + 2x_2 \geq 1, \\
& \quad x_1, x_2 \geq 0.
\end{align*}$$

It has $X^* = \left( \frac{1}{3}, \frac{1}{3} \right)$ as an optimal solution that gives the lower bound of the objective function $z = 4$.

On the other side, the dual of this problem has the following form $D$:

$$\begin{align*}
D: \text{Maximize } & y(W) = w_1 + w_2, \\
\text{Subject to: } & \quad 2w_1 + w_2 \leq 6, \\
& \quad w_1 + 2w_2 \leq 6, \\
& \quad w_1, w_2 \geq 0.
\end{align*}$$
Which has $W^* = (2, 2)$ as an optimal solution that gives the upper bound of the objective function $y = 4$.

As the suggested program the new problem will be as following:

**T\textsubscript{BE}:** Minimize $N_{\text{BE}}(X,W)$,

Subject to:

$N_{\text{BE}} = 6x_1 + 6x_2 - w_1 - w_2$,

$2x_1 + x_2 \geq 1$,

$x_1 + 2x_2 \geq 1$,

$2w_1 + w_2 \leq 6$,

$w_1 + 2w_2 \leq 6$,

$x, W \geq 0$.

Which has $(X^*, W^*) = (1/3, 1/3, 2, 2)$ as an optimal solution that gives the objective function $N_{\text{BE}} = 0$.

These problems are solved using WinQSB software.

When these problems solved using GenoCopIII (a software uses GAs) [8], the results are as shown below:

For problem P: $X^* = (0.331772178411, 0.336455613375)$, and $z = 4.009366512299$. See appendix B

For problem D: $W^* = (2.000329256058, 1.999341607094)$, and $y = 3.999670982361$. See appendix C

And Finally problem $T_{\text{BE}}$: $(X^*, W^*) = (0.333626151085, 0.333186924458, 2.028869152069, 1.942261934280)$, and $N_{\text{BE}} = 0.029747247696$. See appendix D

6. Conclusions

Under assumption that the two problems P & D has an optimal solution, we can conclude that the GAs have the ability to know the goodness of solution obtained in the processing generation, also it can gives a strict decision about the optimality of this solution.

In that area, it is remain to cover the general idea between both problems in different aspects when one of the two problems is infeasible, unbounded, … etc.

The DGAs parameters design has a great effect on the value $N_{\text{BH}}$ to stop, in other words, these parameters can be adjusted to get more accurate solution.

Also in the area of Nonlinear Programming, this topic can be generalized easy using what is meant by the “the Sequential Linearization Technique” [1,2], and it can be generalized to cover the area of Integer, Multi-objective and Large-Scale Programming which we hope to do in the future work.
Appendix A
Please insert Flow chart of the Basic Genetic Algorithm
Appendix B
GenoCopIII Solution of the P problem

-- GENERAL DATA ------------------------------------------

No. of variables ..............: 2
No. of linear equalities ..........: 0
No. of non-linear inequalities ......: 0
No. of linear inequalities ..........: 2
No. of domain constraints ............: 2
Reference population size ............: 5
Search population size .............: 20
No. of operators .................: 9
Total no. of evaluations ..........: 10000
Reference population evolution period : 100
No. of offspring for ref. population : 10
Reference point selection method ......: 0
Search point repair method ..........: 0
Search point replacement ratio ......: 0.250000
Reference point initialization method : 0
Search point initialization method ..: 0
Objective function type ............: 1
Test case number .................: 490634662
Precision factor ..................: 0.000100
Random number generator Seed #1 ......: 15000
Random number generator Seed #1 ......: 15000
Frequency control mode.............: 0

-- DOMAIN CONSTRAINTS -------------------------------------

Variable # 1; Lower Limit 0.000000; Upper Limit 2.000000.
Variable # 2; Lower Limit 0.000000; Upper Limit 2.000000.

-- LINEAR CONSTRAINTS -------------------------------------

# 1 : 2.000000*X[1] + 1.000000*X[2] + -1.000000 >= 0.0
# 2 : 1.000000*X[1] + 2.000000*X[2] + -1.000000 >= 0.0

-- OPERATOR DATA ------------------------------------------

Operator # 1 : 8.000000
Operator # 2 : 4.000000
Operator # 3 : 6.000000
Operator # 4 : 4.000000
Operator # 5 : 8.000000
Operator # 6 : 7.000000
Operator # 7 : 4.000000
Operator # 8 : 7.000000
Operator # 9 : 4.000000

-- NORMALIZED -----------------------------------------------
-- OPERATOR DATA ------------------------------------------

Operator # 1 : 0.153846
Operator # 2 : 0.076923
Operator # 3 : 0.115385
Operator # 4 : 0.076923
Operator # 5 : 0.153846
Operator # 6 : 0.134615
Operator # 7 : 0.076923
Operator # 8 : 0.134615
Operator # 9 : 0.076923

-- OPTIMIZATION -------------------------------------------
-- Initial Evaluation ---------------------------------------

Evaluation Count: 25; Best Ref. Val = 8.547095298767
Evaluation Count: 28; Best Ref. Val = 8.54709053096
Evaluation Count: 31; Best Ref. Val = 8.547089567621
Evaluation Count: 81; Best Ref. Val = 7.434729099274
Evaluation Count: 86; Best Ref. Val = 7.027225017548
Evaluation Count: 92; Best Ref. Val = 6.973612308502
Evaluation Count: 98; Best Ref. Val = 6.967389106750
Evaluation Count: 100; Best Ref. Val = 4.014125823975
Evaluation Count: 646; Best Ref. Val = 4.011924743652
Evaluation Count: 649; Best Ref. Val = 4.01077566101
Evaluation Count: 656; Best Ref. Val = 4.009944438934
Evaluation Count: 675; Best Ref. Val = 4.009549140930
Evaluation Count: 678; Best Ref. Val = 4.009538650513
Evaluation Count: 681; Best Ref. Val = 4.009522438049
Evaluation Count: 700; Best Ref. Val = 4.009490966797
Evaluation Count: 717; Best Ref. Val = 4.009471893311
Evaluation Count: 732; Best Ref. Val = 4.009469509125
Evaluation Count: 755; Best Ref. Val = 4.009468078613
Evaluation Count: 756; Best Ref. Val = 4.009466171265
Evaluation Count: 760; Best Ref. Val = 4.009464263916
Evaluation Count: 785; Best Ref. Val = 4.009463787079
Evaluation Count: 820; Best Ref. Val = 4.009463310242
Evaluation Count: 838; Best Ref. Val = 4.009462833405
Evaluation Count: 6349; Best Ref. Val = 4.009374618530
Evaluation Count: 6425; Best Ref. Val = 4.009372711182
Evaluation Count: 6426; Best Ref. Val = 4.009368419647
Evaluation Count: 6500; Best Ref. Val = 4.009366512299
--BEST VECTOR FOUND-------------------------------------

0.331772178411 0.336455613375

--ACTUAL OPERATOR FREQUENCY--------------------------

Operator #1 was used 1776 times
Operator #2 was used 863 times
Operator #3 was used 1214 times
Operator #4 was used 817 times
Operator #5 was used 1632 times
Operator #6 was used 1515 times
Operator #7 was used 836 times
Operator #8 was used 1452 times
Operator #9 was used 871 times
Appendix C
GenoCopIII Solution of the D problem

-- GENERAL DATA ---------------------------------------------

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of variables</td>
<td>2</td>
</tr>
<tr>
<td>No. of linear equalities</td>
<td>0</td>
</tr>
<tr>
<td>No. of non-linear inequalities</td>
<td>0</td>
</tr>
<tr>
<td>No. of linear inequalities</td>
<td>2</td>
</tr>
<tr>
<td>No. of domain constraints</td>
<td>2</td>
</tr>
<tr>
<td>Reference population size</td>
<td>5</td>
</tr>
<tr>
<td>Search population size</td>
<td>20</td>
</tr>
<tr>
<td>No. of operators</td>
<td>9</td>
</tr>
<tr>
<td>Total no. of evaluations</td>
<td>10000</td>
</tr>
<tr>
<td>Reference population evolution period</td>
<td>100</td>
</tr>
<tr>
<td>No. of offspring for ref. population</td>
<td>10</td>
</tr>
<tr>
<td>Reference point selection method</td>
<td>0</td>
</tr>
<tr>
<td>Search point repair method</td>
<td>0</td>
</tr>
<tr>
<td>Search point replacement ratio</td>
<td>0.25000</td>
</tr>
<tr>
<td>Reference point initialization method</td>
<td>0</td>
</tr>
<tr>
<td>Search point initialization method</td>
<td>0</td>
</tr>
<tr>
<td>Objective function type</td>
<td>0</td>
</tr>
<tr>
<td>Test case number</td>
<td>273</td>
</tr>
<tr>
<td>Precision factor</td>
<td>0.000100</td>
</tr>
<tr>
<td>Random number generator Seed #1</td>
<td>15000</td>
</tr>
<tr>
<td>Random number generator Seed #1</td>
<td>15000</td>
</tr>
<tr>
<td>Frequency control mode</td>
<td>0</td>
</tr>
</tbody>
</table>

-- DOMAIN CONSTRAINTS ---------------------------------------

Variable # 1; Lower Limit 0.000000; Upper Limit 10.000000.
Variable # 2; Lower Limit 0.000000; Upper Limit 10.000000.

-- LINEAR CONSTRAINTS ----------------------------------------


-- OPERATOR DATA ---------------------------------------------

Operator # 1 : 8.000000
Operator # 2 : 23.000000
Operator # 3 : 24.000000
Operator # 4 : 8.000000
Operator # 5 : 24.000000
Operator # 6 : 8.000000
Operator # 7 : 8.000000
<table>
<thead>
<tr>
<th>Operator #</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.067227</td>
</tr>
<tr>
<td>2</td>
<td>0.193277</td>
</tr>
<tr>
<td>3</td>
<td>0.201681</td>
</tr>
<tr>
<td>4</td>
<td>0.067227</td>
</tr>
<tr>
<td>5</td>
<td>0.201681</td>
</tr>
<tr>
<td>6</td>
<td>0.067227</td>
</tr>
<tr>
<td>7</td>
<td>0.067227</td>
</tr>
<tr>
<td>8</td>
<td>0.067227</td>
</tr>
<tr>
<td>9</td>
<td>0.067227</td>
</tr>
</tbody>
</table>

--- NORMALIZED -----------------------------------------------

--- OPERATOR DATA ------------------------------------------

<table>
<thead>
<tr>
<th>Operator #</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.067227</td>
</tr>
<tr>
<td>2</td>
<td>0.193277</td>
</tr>
<tr>
<td>3</td>
<td>0.201681</td>
</tr>
<tr>
<td>4</td>
<td>0.067227</td>
</tr>
<tr>
<td>5</td>
<td>0.201681</td>
</tr>
<tr>
<td>6</td>
<td>0.067227</td>
</tr>
<tr>
<td>7</td>
<td>0.067227</td>
</tr>
<tr>
<td>8</td>
<td>0.067227</td>
</tr>
<tr>
<td>9</td>
<td>0.067227</td>
</tr>
</tbody>
</table>

--- OPTIMIZATION -------------------------------------------

--- Initial Evaluation ----------------------------------------

<table>
<thead>
<tr>
<th>Evaluation Count</th>
<th>Best Ref. Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>2.705785036087</td>
</tr>
<tr>
<td>41</td>
<td>2.724233389011</td>
</tr>
<tr>
<td>42</td>
<td>2.896569252014</td>
</tr>
<tr>
<td>44</td>
<td>3.123461723288</td>
</tr>
<tr>
<td>67</td>
<td>3.163487911224</td>
</tr>
<tr>
<td>80</td>
<td>3.502447128296</td>
</tr>
<tr>
<td>90</td>
<td>3.520674705505</td>
</tr>
<tr>
<td>124</td>
<td>3.530769585653</td>
</tr>
<tr>
<td>139</td>
<td>3.753813266754</td>
</tr>
<tr>
<td>162</td>
<td>3.885721206665</td>
</tr>
<tr>
<td>180</td>
<td>3.904642105103</td>
</tr>
<tr>
<td>200</td>
<td>3.929771423340</td>
</tr>
<tr>
<td>302</td>
<td>3.929925441742</td>
</tr>
<tr>
<td>329</td>
<td>3.931427478970</td>
</tr>
<tr>
<td>7489</td>
<td>3.946381092072</td>
</tr>
<tr>
<td>7492</td>
<td>3.949172973633</td>
</tr>
<tr>
<td>7500</td>
<td>3.960334777832</td>
</tr>
<tr>
<td>7504</td>
<td>3.975806204531</td>
</tr>
<tr>
<td>7512</td>
<td>3.979189157486</td>
</tr>
<tr>
<td>7522</td>
<td>3.981322765350</td>
</tr>
<tr>
<td>7530</td>
<td>3.985266208649</td>
</tr>
<tr>
<td>7533</td>
<td>3.988231658936</td>
</tr>
<tr>
<td>7560</td>
<td>3.993949990137</td>
</tr>
<tr>
<td>7565</td>
<td>3.994106292725</td>
</tr>
<tr>
<td>7581</td>
<td>3.994702100754</td>
</tr>
<tr>
<td>7600</td>
<td>3.998849153519</td>
</tr>
<tr>
<td>7849</td>
<td>3.998871564865</td>
</tr>
<tr>
<td>7896</td>
<td>3.998879432678</td>
</tr>
</tbody>
</table>
Evaluation Count: 7900 ; Best Ref. Val = 3.999114990234
Evaluation Count: 7921 ; Best Ref. Val = 3.999491930008
Evaluation Count: 7946 ; Best Ref. Val = 3.999511003494
Evaluation Count: 7977 ; Best Ref. Val = 3.999559640884
Evaluation Count: 8019 ; Best Ref. Val = 3.999572753906
Evaluation Count: 8052 ; Best Ref. Val = 3.999582290649
Evaluation Count: 8089 ; Best Ref. Val = 3.999582767487
Evaluation Count: 8101 ; Best Ref. Val = 3.999583721161
Evaluation Count: 8105 ; Best Ref. Val = 3.999584674835
Evaluation Count: 8117 ; Best Ref. Val = 3.99958628510
Evaluation Count: 8119 ; Best Ref. Val = 3.999593734741
Evaluation Count: 8126 ; Best Ref. Val = 3.999633312225
Evaluation Count: 8179 ; Best Ref. Val = 3.999636650085
Evaluation Count: 8185 ; Best Ref. Val = 3.99966782919
Evaluation Count: 8250 ; Best Ref. Val = 3.99966859756
Evaluation Count: 8275 ; Best Ref. Val = 3.999668598175
Evaluation Count: 8285 ; Best Ref. Val = 3.999668836594
Evaluation Count: 8324 ; Best Ref. Val = 3.999669075012
Evaluation Count: 8369 ; Best Ref. Val = 3.999670028687
Evaluation Count: 8496 ; Best Ref. Val = 3.999670267105
Evaluation Count: 8548 ; Best Ref. Val = 3.999670505524
Evaluation Count: 9730 ; Best Ref. Val = 3.999670743942
Evaluation Count: 9757 ; Best Ref. Val = 3.999670982361

-- BEST VECTOR FOUND -----------------------------------------------

2.000329256058 1.999341607094

-- ACTUAL OPERATOR FREQUENCY ---------------------------------------

Operator #1 was used 735 times
Operator #2 was used 2133 times
Operator #3 was used 2251 times
Operator #4 was used 747 times
Operator #5 was used 2134 times
Operator #6 was used 735 times
Operator #7 was used 756 times
Operator #8 was used 735 times
Operator #9 was used 750 times
Appendix D
GenoCopIII Solution of the TBE problem

-- GENERAL DATA -------------------------------------------

No. of variables .....................: 4
No. of linear equalities .............: 0
No. of non-linear inequalities ......: 0
No. of linear inequalities ..........: 4
No. of domain constraints ..........: 4
Reference population size ..........: 10
Search population size ..........: 50
No. of operators .................: 9
Total no. of evaluations ..........: 10000
Reference population evolution period : 100
No. of offspring for ref. population : 10
Reference point selection method ....: 0
Search point repair method .......: 1
Search point replacement ratio ..: 0.250000
Reference point initialization method : 1
Search point initialization method : 1
Objective function type ..........: 1
Test case number .................: 0
Precision factor .................: 0.000000
Random number generator Seed #1.......: 1000
Random number generator Seed #1.......: 5000
Frequency control mode...........: 1

-- DOMAIN CONSTRAINTS -------------------------------------

Variable # 1; Lower Limit 0.000000; Upper Limit 2.000000.
Variable # 2; Lower Limit 0.000000; Upper Limit 2.000000.
Variable # 3; Lower Limit 0.000000; Upper Limit 10.000000.
Variable # 4; Lower Limit 0.000000; Upper Limit 10.000000.

-- LINEAR CONSTRAINTS -------------------------------------

0.000000*X[4] + -1.000000 >= 0.0
0.000000*X[4] + -1.000000 >= 0.0
1.000000*X[4] + 6.000000 >= 0.0
# 4 : 0.000000*X[1] + 0.000000*X[2] + -1.000000*X[3] + -
2.000000*X[4] + 6.000000 >= 0.0
--OPERATOR DATA------------------------------------------

Operator # 1 : 1.000000
Operator # 2 : 1.000000
Operator # 3 : 1.000000
Operator # 4 : 1.000000
Operator # 5 : 1.000000
Operator # 6 : 1.000000
Operator # 7 : 1.000000
Operator # 8 : 1.000000
Operator # 9 : 1.000000

--NORMALIZED---------------------------------------------

Operator # 1 : 0.111111
Operator # 2 : 0.111111
Operator # 3 : 0.111111
Operator # 4 : 0.111111
Operator # 5 : 0.111111
Operator # 6 : 0.111111
Operator # 7 : 0.111111
Operator # 8 : 0.111111
Operator # 9 : 0.111111

--OPTIMIZATION-------------------------------------------

--Initial Evaluation-----------------------------------

Evaluation Count:  60; Best Ref. Val = 1.821146130562
Evaluation Count:  133; Best Ref. Val = 1.821145176888
Evaluation Count:  154; Best Ref. Val = 1.821128964424
Evaluation Count:  158; Best Ref. Val = 1.821128487587
Evaluation Count:  164; Best Ref. Val = 1.818080663681
Evaluation Count:  193; Best Ref. Val = 0.75082648444
Evaluation Count:  198; Best Ref. Val = 0.710123181343
Evaluation Count:  200; Best Ref. Val = 0.99754176788
Evaluation Count:  263; Best Ref. Val = 0.080043315887
Evaluation Count:  281; Best Ref. Val = 0.029747247696

--BEST VECTOR FOUND--------------------------------------

0.33626151085 0.33186924458 2.028869152069 1.942261934280

--ACTUAL OPERATOR FREQUENCY-------------------------------

Operator #1 was used 1252 times
Operator #2 was used 1228 times
Operator #3 was used 1182 times
Operator #4 was used 1231 times
Operator #5 was used 1222 times
Operator #6 was used 1206 times
Operator #7 was used 1233 times
Operator #8 was used 1162 times
Operator #9 was used 1225 times

**Solution with N = 0.0297**
A Directed Genetic Algorithms

References


