Global Journal of Computer Science and Technology
Volume 12 Issue 2 Version 1.0 January 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

# A Cultural Algorithm for the Two Variable Integer Programming Problem

By Senthil Kumar Ramadoss, I.K. Gulam Mohiddin, Ajit Pal Singh
*University Adama*

*Abstract -* A specific implementation of cultural algorithm is presented here for solving the following two variable integer programming problem with $n$ constraints: Maximize or Minimize

are signed integers. A cultural algorithm consists of a population component almost identical to that of the genetic algorithm and, in addition, a knowledge component called the belief space. As the integer programming problem is a constrained optimization problem, the constraints including non-negativity and integer restrictions are availed as the knowledge component and used to build the belief space.

*Keywords : Cultural algorithm, Integer programming problem (IPP), Belief space, Genetic algorithm, Optimization.*

*GJCST Classification: G.1.6*

# A Cultural Algorithm for the Two Variable Integer Programming Problem

Senthil Kumar Ramadoss [α], I.K. Gulam Mohiddin [Ω], Ajit Pal Singh [β]

*Abstract* - A specific implementation of cultural algorithm is presented here for solving the following two variable integer programming problem with $n$ constraints: Maximize or Minimize $Z = p_1x + qx_2$ subject to constraints $\{a_1x_1 + b_1x_1 \le c_i;\ or\ a_1x_1 + b_1x_1 \ge c_i\}$, where $i = 1,2,3,\ldots,n;\ x_1, x_2 \ge 0;\ x_1$ and $x_2$ are integers; $a_i, b_i$ and $c_i$ are positive real numbers; $p$ and $q$ are signed integers. A cultural algorithm consists of a population component almost identical to that of the genetic algorithm and, in addition, a knowledge component called the belief space. As the integer programming problem is a constrained optimization problem, the constraints including non-negativity and integer restrictions are availed as the knowledge component and used to build the belief space.

*Keywords : Cultural algorithm, Integer programming problem (IPP), Belief space, Genetic algorithm, Optimization.*

## I. Introduction

We consider the following integer programming problem (IPP) with two decision variables and $n$ constraints: Maximize or Minimize $Z = p_1x + qx_2$ subject to constraints $\{a_1x_1 + b_1x_1 \le c_i;\ or\ a_1x_1 + b_1x_1 \ge c_i\}$, where $i = 1,2,3,\ldots,n;\ x_1, x_2 \ge 0;\ x_1$ and $x_2$ are integers; $a_i, b_i$ and $c_i$ are positive real numbers; $p$ and $q$ are signed integers.

In 1958, Gomory (1963a) devised a method, known as the "method for integer forms", for solving integer programming problems. In 1960, Gomory (1963b) devised another method for solving all integer linear programming problems. Several computers codes using one or both methods have been written; and have successfully solved many real problems. However, their performance has not been nearly as predictable as that of ordinary linear programming codes- which are themselves rather unpredictable as regards running time (Beale, 1965). Martin et al. (1963) used a variant method of integer form called the "accelerated euclidean algorithms". Dakin (1964) and Driebeck (1964) have developed programs using a "branch and bound" method for mixed integer programming. Later Forest et al. (1974), Tomlin (1971), Driebeck (1966), Dakin (1965), Beale and Small (1965), and others improved or refined the branch and bound approach of solving integer programming problems in a number of ways. There are many survey articles and text books to describe the usage of refined and improved methods of using branch and bound approaches (Hansen, 1979; Gupta and Ravindran, 1983a, 1985b)

But the most important facet of both Gomory's cutting plan approach and branch and bound approach is that they can be applied only after obtaining non-integer solution using traditional optimization techniques like Simplex Algorithms.

But the proposed cultural algorithmic approach directly searches the integer solution in the population space which is a space obtained by narrowing the population space where the population space is obtained by the constraints of the problem.

The implementation of the cultural algorithm is presented here to find $(x_1, x_2)$ which satisfies all the $n$ constraints and yields the optimal value for $Z$. As IPP is a kind of constrained optimization problem, the constraints including non-negativity constraints and integer restrictions are availed as knowledge component to build the belief space. A cultural algorithm, introduced by Reynolds (1994), and seen as extension to genetic algorithm, is a computational model of cultural evolution process in nature where there is a knowledge component in addition to population component. The knowledge component is used to build belief space. The best individuals are selected from belief space using a fitness function. These best individuals are used to update the belief space via a vote acceptance function.

Generally cultural algorithms use five different kinds of knowledge component namely: normative, domain specific, situational, historical and topographical knowledge. The proposed cultural algorithm for solving two variable integer programming problems with $n$ constraints uses a normative kind of knowledge gained from constraints including non-negativity and integer restrictions.

The proposed cultural algorithm for solving two variable integer programming problems uses the frame work for constrained optimization problem introduced by

*Author [α Ω] : Department of Computer Science, Eritrea Institute of Technology, Asmara, The State of Eritrea, North East Africa.*
*E-mail : senanu.kumdev@gmail.com, ikgulam@gmail.com*
*Author [β] : Department of Mechanical & Vehicle Engineering, Adama Science & Technology University, Adama, Ethiopia, Africa.*
*E-mail : singh_ajit_pal@hotmail.com*

Carlos and Licardo (2002) to identify knowledge component and to build belief space. The algorithm is also using a variant of test bed introduced by Chang and Reynolds (1996) as the backbone of computational procedure.

## II. The Cultural Algorithm for Sloving Two Variable Interger Programmin Problems

- *Step 1: Initialization of Population Space*

Lets define sets of integers for $x_1$ and $x_2$ that satisfies each of $n$ constraints and let these sets be $S_1, S_2, ..., S_n$ and $T_1, T_2, ..., T_n$ respectively. $S_i$ and $T_i$ can be defined as follows:

i. *Definition of Population Space for* $x_1$

When the constrain $i$ is as $a_i x_1 + b_i x_1 \le c_i$ with $\le$, then $S_i$ can be defined as $0 \le x_1 \le c_i / a_i$ irrespective of $b_i$ and if $c_i / a_i$ is real, it should be rounded off to the immediate lower positive integer. When the constraint *i* is as $a_i x_1 + b_i x_1 \ge c_i$ with $\ge$, then $S_i$ can be defined as $c_i / a_i \le x_1 \le \infty$ irrespective of $b_i$ and if $c_i / a_i$ is real, it should be rounded off to the immediate higher positive integer. Note that if $a_i = 0$ in the constraint $i$, then this constraint is ignored in defining space for $x_1$.

ii. *Definition of Population Space for* $x_2$

When the constraint $i$ is as $a_i x_1 + b_i x_2 \le c_i$ with $\le$, then $T_i$ can be defined as $0 \le x_1 \le c_i / b_i$ irrespective of $a_i$ and if $c_i / b_i$ is real, it should be rounded off to the immediate lower positive integer. When the constraint $i$ is as $a_i x_1 + b_i x_2 \ge c_i$ with $\ge$, then $T_i$ can be defined as $c_i / b_i \le x_1 \le \infty$ irrespective of $a_i$ and if $c_i / b_i$ is real, it should be rounded off to the immediate higher positive integer. Note that if $b_i = 0$ in the constraint $i$, then this constraint is ignored in defining space for $x_2$.

Now our objective is to find $(x_1, x_2)$ for which $px_1 + qx_2$ is the minimum or maximum according to the objective function, searching $x_1$ from $(S_1, S_2, ..., or\, S_n)$ and $x_2$ from $(T_1, T_2, ..., or\, T_n)$.

Here, all the population of $S_i$ cannot satisfy all $n$ constraints; similarly any member of $T_i$ cannot satisfy all $n$ constraints. Hence we identify the lowest $(l)$ and greatest $(g)$ member of all common components in $S_1, S_2, ..., S_n$ and we build population space for $x_1$, $POP(x_1) = \{l, l+1, l+2, ..., g-2, g-1, g\}$ as the range of values between lowest $(l)$ and greatest $(g)$ for $x_1$ will satisfy all the $n$ constraints. This population space can also be stated as $POP(x_1) = \{S_1 \cap S_2 \cap, ..., \cap S_n\}$. Similarly, we build the population space for $x_2$ as $POP(x_2) = \{T_1 \cap T_2 \cap, ..., \cap T_n\}$.

- *Step 2: Initialization of Belief Space*

When the objective function is to maximize $px_1 + qx_2$ and if $p \rangle 0$ then the largest component of $POP(x_1)$ can only aid in achieving maximum possible value of $px_1 + qx_2$. When the objective function is to maximize and if $p \langle 0$ then the smallest component of $POP(x_1)$ can only aid in achieving maximum possible value of $px_1 + qx_2$.

Similarly, When the objective function is to maximize $px_1 + qx_2$ and if $q \rangle 0$ then the largest component of $POP(x_2)$ can only aid in achieving maximum possible value of $px_1 + qx_2$. When the objective function is to maximize and if $q \langle 0$ then the smallest component of $POP(x_2)$ can only aid in achieving maximum possible value of $px_1 + qx_2$. Using this knowledge, the *fitness function* is built to identify the best fit component (or the optimal contributor $(OC)$ in achieving optimal $px_1 + qx_2$) to build the belief space. The fitness function which identifies and returns the optimal contributor is defined as follows:

*Fitness Function:*

When the objective function is to maximize $px_1 + qx_2$:

The $OC$ for $x_1$ is $\max(POP(x_1))$ if $p \rangle 0$; The $OC$ for $x_1$ is $\min(POP(x_1))$ if $p \langle 0$;

The $OC$ for $x_2$ is $\max(POP(x_2))$ if $q \rangle 0$; The $OC$ for $x_2$ is $\min(POP(x_2))$ if $q \langle 0$;

Similarly, when the objective function is to minimize $px_1 + qx_2$:

The $OC$ for $x_1$ is $\min(POP(x_1))$ if $p \rangle 0$; The $OC$ for $x_1$ is $\max(POP(x_1))$ if $p \langle 0$;

The $OC$ for $x_2$ is $\min(POP(x_2))$ if $q \rangle 0$; The $OC$ for $x_2$ is $\max(POP(x_2))$ if $q \langle 0$;

Global Journal of Computer Science and Technology Volume XII Issue II Version I January 2012

When the objective function is to maximize $px_1 + qx_2$, the optimal contribution that $x_1$ can put in $px_1 + qx_2$ is $\left[OC(x_1))^* p\right]$ and $x_2$ can put is $\left[OC(x_2))^* q\right]$. Here if $\left[OC(x_1))^* p\right] \geq \left[OC(x_2))^* q\right]$ then $x_1$ can be greater contributor than $x_2$, otherwise $x_2$ is greater contributor than $x_1$ in achieving optimal $px_1 + qx_2$.

Similarly, when the objective function is to minimize $px_1 + qx_2$, and if $[OC(x_1))^* p] \geq [OC(x_2))^* q]$ then $x_2$ can be greater contributor than $x_1$, otherwise $x_1$ is greater contributor than $x_2$ in achieving optimal $px_1 + qx_2$.

Thus, firstly, this greater contributor $(GC)$ must be identified. An exceptional instance occurs while identifying this $GC$ is that $POP(x_1)$ and/or $POP(x_2)$ may be a null set or unbounded or infinite. That is there may be no common component in $S_1, S_2, ..., S_n$ and/or in $T_1, T_2, ..., T_n$. This indicates the population space of $x_1$ and/or $x_2$ is unbounded or indefinite. We adapt the following rules in such instances.

1. If the population space of both $x_1$ and $x_2$ are bounded/definite and non-null, we identify the $GC$ as discussed earlier and make use of this in building belief space.
2. If the population space of any one variable is found to be unbounded/indefinite or null, we consider the other variable as $GC$ in building belief space irrespective of the unbounded/indefinite or null population space of former variable.
3. If the population spaces of both variables are found to be unbounded/indefinite or null, then this is indication that there is inability in building the belief space. With such population spaces, we declare that the problem has infeasible or unbounded solution space.

When the $GC$ is identified evidently, we believe that the $\max(POP)(GC))$ is playing the greatest role than all other components in the population space in achieving the maximum $px_1 + qx_2$ and $\min(POP)(GC))$ is playing the greatest role than all other components in population space in achieving the minimum $px_1 + qx_2$. Note that, here, $GC$ is either $x_1$ or $x_2$. Thus we define the belief space $BLF(i)$, ($i$ is initially 0), as follows:

*Building Belief Space:*

*Case I: When $x_1$ is found to be $GC$ and the objective function is to maximize $px_1 + qx_2$:*

$BLF(i) = (x_1, x_2)$ such that:

$x_1 = \max(POP(x_1))$;

$x_2 \, \varepsilon \, B;$

where $B$ is a set of integers which is intersection of $B_1, B_2, ..., B_n$ and $B_i$ can be defined as $x_2 \leq (c_i - a_i x_1)/b_i$ if the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with $\leq$ and as $x_2 \geq (c_i - a_i x_1)/b_i$ if the constraint $i$ as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ where $x_1 = \max(POP(x_1))$.

If the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with of $\leq$, then $(c_i - a_i x_1)/b_i$ should be rounded off to the immediate lower positive integer and it is as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$, then $(c_i - a_i x_1)/b_i$ should be rounded off to the immediate higher positive integer.

*Case II : When $x_1$ is found to be $GC$ and the objective function is to minimize $px_1 + qx_2$:*

$BLF(i) = (x_1, x_2)$ such that:

$x_1 = \min(POP(x_1))$;

$x_2 \, \varepsilon \, B;$

where $B$ is a set of integers which is intersection of $B_1, B_2, ..., B_n$ and $B_i$ can be defined as $x_2 \leq (c_i - a_i x_1)/b_i$ if the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with $\leq$ and as $x_2 \geq (c_i - a_i x_1)/b_i$ if the constraint $i$ as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ where $x_1 = \min(POP(x_1))$.

If the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with of $\leq$, then $(c_i - a_i x_1)/b_i$ should be rounded off to the immediate lower positive integer and it is as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ , then $(c_i - a_i x_1)/b_i$ should be rounded off to the immediate higher positive integer.

*Case III: When $x_2$ is found to be $GC$ and the objective function is to maximize $px_1 + qx_2$:*

$BLF(i) = (x_1, x_2)$ such that:

$x_1 \, \varepsilon \, B;$

$x_2 = \max(POP(x_2))$;

where $B$ is a set of integers which is intersection of $B_1, B_2, ..., B_n$ and $B_i$ can be defined as $x_1 \leq (c_i - b_i x_2)/a_i$ if the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with $\leq$ and as $x_1 \geq (c_i - b_i x_2)/a_i$ if the constraint $i$ as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ where $x_2 = \max(POP(x_1))$.

If the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with of $\leq$, then $(c_i - b_i x_2)/a_i$ should be rounded off to the immediate lower positive integer and it is as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$, then $(c_i - b_i x_2)/a_i$ should be rounded off to the immediate higher positive integer.

*Case IV: When $x_2$ is found to be GC and the objective function is to minimize $px_1 + qx_2$:*

$BLF(i) = (x_1, x_2)$ such that:

$x_1 \varepsilon B$;

$x_2 = \min(POP(x_2))$;

where $B$ is a set of integers which is intersection of $B_1, B_2, ..., B_n$ and $B_i$ can be defined as $x_1 \leq (c_i - b_i x_2)/a_i$ if the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with $\leq$ and as $x_1 \geq (c_i - b_i x_2)/a_i$ if the constraint $i$ as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ where $x_2 = \min(POP(x_1))$.

If the constraint $i$ is as $a_i x_1 + b_i x_2 \leq c_i$ with of $\leq$, then $(c_i - b_i x_2)/a_i$ should be rounded off to the immediate lower positive integer and it is as $a_i x_1 + b_i x_2 \geq c_i$ with $\geq$ , then $(c_i - b_i x_2)/a_i$ should be rounded off to the immediate higher positive integer.

Note that, hereafter, we use the terms greater contributor $(GC)$ and lower contributor $(LC)$, instead of $x_1$ and $x_2$. If $x_1$ is found to be $GC$ in the earlier step then $x_2$ is $LC$, otherwise $x_1$ is $LC$ and $x_2$ is $GC$.

- *Step 3: Evaluation of Space*

The moment we arrive to evaluate the current belief space, $GC$ is fixed with one single best component and $LC$ lies in the set $B$. Now lets search the best component for $LC$ from $B$ which may produce the optimal value of $px_1 + qx_2$ along with the already fixed best component of $GC$. To find the best component for $LC$, We evaluate the set $B$ as follows:

*Case I:* If the objective function is to maximize $px_1 + qx_2$ and the co-efficient of $LC$ in objective function is greater than zero or if the objective function is to minimize $px_1 + qx_2$ and the co-efficient of $LC$ in objective function is less than zero, then the largest component of $B$ is considered as the best component for $LC$.

*Case II:* If the objective function is to maximize $px_1 + qx_2$ and the co-efficient of $LC$ in objective function is less than zero or if the objective function is to minimize $px_1 + qx_2$ and the co-efficient of $LC$ in objective function is greater than zero , the smallest component of $B$ is considered as the best component for $LC$.

Find the value of $Z = px_1 + qx_2$ using the best components of $GC$ and $LC$. And lets call this value as $Z = BLF(i))$ and classify this outcome of evaluation as non-futile.

There may be an uncommon situation here while evaluating current belief space in search of the best component of $LC$ that suits with the best component of $GC$ which is already fixed while defining the current belief space. The unusual situation is that the set $B$ identified for $LC$ while defining the current belief space may be a null set. This indicates that there exist no single component in population space of $LC$ to accept the best component chosen for $GC$ to build the current belief space. In such situations, we assume that the mission with current belief space is failed and classify this outcome of evaluation as futile.

- *Step 4: Vote Acceptance Function*

If the $Z = BLF(i)) \langle Z(BLF(i-1))$ when the objective function is to maximize $px_1 + qx_2$ then we stop the process and declare that $Z = BLF(i-1))$ is the optimal solution and $x_1$ and $x_2$ are the best components used for $GC$ and $LC$ in calculating $Z = BLF(i-1))$.

Similarly, If the $Z = BLF(i)) \rangle Z(BLF(i-1))$ when the objective function is to minimize $px_1 + qx_2$ then we stop the process and declare that $Z = BLF(i-1))$ is the optimal solution and $x_1$ and $x_2$ are the best components used for $GC$ and $LC$ in calculating $Z = BLF(i-1))$.

The vote acceptance function rejects the outcome of evaluation in all other circumstances except the above explained situations. Thus the function rejects the outcome of evaluation and suggest to reproduce the population space in all the following instances:

IF WE ARE WORKING WITH INITIAL BELIEF SPACE (THAT IS $BLF(i)$ WHERE $i = 0$)

IF THE OUTCOME OF EVALUATION OF $BLF(i)$ WAS FOUND TO BE FUTILE.

IF $Z(BLF(i)) \rangle Z(BLF(i-1))$ WHEN THE OBJECTIVE FUNCTION IS TO MAXIMIZE $Z = px_1 + qx_2$.

If $Z(BLF(i)) \langle Z(BLF(i-1))$ when the objective function is to minimize $Z = px_1 + qx_2$.

- *Step 5: Modification in the Belief Space*

When the vote acceptance function suggests to reproduce population space, we modify the current belief space $BLF(i)$ to get $BLF(i+1)$ by the modifying the $GC$. Note that the $GC$ is one fixed component and $LC$ lies in the set $B$ in the current belief space. We now modify this belief space by adjusting the component fixed to the $GC$. Based on this modified $GC$, we define once again the set $B$ where $LC$ lies, which along with fixed modified component of $GC$ forms new belief space $BLF(i+1)$. Thus the modification to $GC$ is brought as follows:

*Case I:* If the objective function is to maximize $px_1 + qx_2$ and the co-efficient of $GC$ in objective function is greater than zero or if the objective function is to minimize $px_1 + qx_2$ and the co-efficient of $GC$ in objective function is greater than zero, $GC$ is fixed with the immediate lower positive integer to the component fixed in the current belief space.

*Case II:* If the objective function is to maximize $px_1 + qx_2$ and the co-efficient of $GC$ in objective function is less than zero or if the objective function is to minimize $px_1 + qx_2$ and the co-efficient of $GC$ in objective function is less than zero, $GC$ is fixed with the immediate higher integer to component fixed in the current belief space.

Now we define the set $B$ based on the component fixed to $GC$, which together form the new belief space $BLF(i+1)$ as shown in the section *2.2.2 Building Belief Space.*

Now the new belief space $BLF(i+1)$ is $(x_1, x_2)$ such that $GC$ is the component codified above and $LC$ belongs to the set $B$ defined above. This modified belief space is then submitted to the promote influence function which will deicide the further course on whether this space is to be evaluated or not.

*a. Step 6: Promote Influence Function*

This modified belief space $BLF(i+1)$ shall be promoted for evaluation through *Step 3*, if the set $B$ of new space, where $LC$ belongs to, is not a null set. The set $B$ of $BLF(i+1)$ being null indicates that there exist no component in the population space to be $LC$, accepting the modified $GC$. When $B$ of $BLF(i+1)$ is found to be null, we stop the process and declare that $Z(BLF(i+1))$ is the optimal solution and $x_1$ and $x_2$ are components used as $GC$ and $LC$ in calculating $Z(BLF(i+1))$.

## III. Conclusion

A specific implementation of cultural algorithm is presented using the computational model of cultural evolution process to solve two variable IPPs. Identifying formative knowledge sources in IPPs, the belief space is built in addition to the traditional framework of genetic algorithms. Whilst the existing approaches like Gomery's and Branch and Bound need non-integer solution produced by traditional optimization methods like simplex algorithms, the proposed algorithm steer clear of the computational load of solving linear programming problem relaxations. Comparing the complexity of Gomery's and Branch and Bound approaches, searching global optimal solution using proposed cultural algorithm is especially slender. This implementation can be extended for solving IPPs with any number of decision variables and constraints with signed integers in future.

## References References Referencias


1. Beale, E.M.L. (1965) Survey of integer programming. *Operational Research Quarterly*, 16(2), pp. 219-228.
2. Beale, E.M.L. and Small, R. (1965) Mixed integer programming by a branch and bound technique. In: *Proceedings of the Third IFIP Congress*, pp. 450-451.
3. Carlos, A. C. C. and Licardo, L. B. (2002) A cultural algorithm for constrained optimization. In: *MICAI-Advances in Artificial Intelligence, Second Mexican, International Conference on Artificial Intelligence*, April 22-26. Merida, Yucatan, Mexico, Proceedings, Series-Lecture Notes in Computer Science, Springer-Verlag London, UK, 2313, pp. 98-107.
4. Chang, C. J. and Reynolds, R.G. (1996) A test bed for solving optimization problems using cultural algorithms. In: John R. M. and Peter, A. (eds.) Evolutionary Programming V, A Bradford Book, MIT Press, Cambridge Massachusetts, pp. 225-236.
5. Dakin, R. (1965) A tree search algorithm for mixed integer programming problems. *Computer Journal*, 8, pp. 250-255.
6. Dakin, R.J. (1964) A MIXED-INTEGER PROGRAMMING ALGORITHM. Basser Computing Laboratory, University of Sydney, Technical Report No. 31.
7. Driebeck, N.J. (1964) A method for the solution of mixed integer and non-linear programming problems by linear programming methods. The International Symposium on Mathematical Programming, London School of Economics.
8. Driebeck, N.J. (1966) An algorithm for the solution of mixed integer programming problems. *Management Science*, 12, pp. 576-587.
9. Forest, J.J.H., Hirst, J.P.H. and Tomlin, J.A. (1974) Practical solution of large mixed integer


Global Journal of Computer Science and Technology Volume XII Issue II Version I January 2012

programming problems with umpire. Management Science, 20(5), pp. 736-773.

10. (a) Gomory, R.E. (1963) All-integer integer programming algorithm. In: J.F. Muth and G.L. Thompson (eds), Industrial Scheduling, Prentice Hall, pp. 193-206. (First issued in IBM Research Center Research Report RC-189, 1960)
11. (b) Gomory, R.E. (1963) An algorithm for integer solutions to linear programs. In: R.L. Graves and P. Wolfe (eds), *Recent Advances in Mathematical Programming*, , McGraw-Hill, pp. 269-302. (First issued in Princeton-IBM Mathematics Research Project Technical Report No. 1, 1958)
12. (a) Gupta, O. K. and Ravindran, A. (1983) Nonlinear integer programming algorithms: a survey. OPSEARCH, 20(4), pp. 189-206.
13. (b) Gupta, O. K. and Ravindran, A. (1985) Branch and bound experiments in convex nonlinear integer programming. Management Science, 31(12), pp. 1533-1546.
14. Hansen, P. (1979) Methods of nonlinear 0-1 programming. Annals of Discrete Mathematics, New York.
15. Martin, G.T., Graves, R.L. and Wolfe, P. (1963) An accelerated euclidean algorithm for integer linear programming. *Recent Advances in Mathematical Programming*, McGraw-Hill, pp. 311-318.
16. Reynolds, R.G. (1994) An introduction to cultural algorithms. In: Proceedings of the Third Annual Conference on Evolutionary Programming, February 24-26, San Diego, California, pp. 131-139.
17. Tomlin, J.A. (1971) An improved branch and bound method for integer programming. *Operations Research*, 19, pp. 1070-1075.