

A Glorious Literature on Linear Goal Programming Algorithms

Ukamaka Cynthia Orumie, Daniel Ebong

Department of Mathematics/Statistics, University of Port Harcourt, Port Harcourt, Nigeria
Email: amakaorumie@yahoo.com, daniel.ebong@uniport.edu.ng

Received 16 October 2013; revised 16 November 2013; accepted 23 November 2013

Copyright © 2014 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In the last several years, there has been a marked improvement in the development of new algorithms for solving Linear Goal programming (LGP). This paper presents a survey of current methods for LGP.

Keywords

Linear Goal Programming; Algorithms; Current Methods

1. Introduction

Goal programming (GP) has been a popular theoretical method for dealing with multiple objective decision-making problems. In recent years, a number of linear goal programming algorithms have evolved based on the Dantzig [1]. Charnes *et al.* [2], Lee [3], Ignizio [4] and many others have been instrumental in the development of various algorithms of linear goal programming. Lin [5] presents an impressive list of articles which propose or apply goal programming.

However, a major setback in applying linear goal programming (LGP) method has been the lack of an algorithm capable of reaching optimality in reasonable time. Literature recorded that the most commonly used goal programming solution methods were introduced by Lee [3] and Ignizio [4] based on Dantzig's simplex method. Both methods require columns in the simplex tableau for positive and negative deviational variables. They also require separate $c_j - z_j$ rows for each priority level, all of which increase the computational time of the solution method. This paper presents a survey of current goal programming algorithms and their limitations.

The paper is organized as follows: Introduction and/general structure of the Linear Goal Programming Model is presented in Section Two. Goal Programming Assumptions and General Principles (Concepts) are presented in Section Three, whereas current literature on linear goal programming algorithm and its limitations are presented in Section Four. Section five presents the concluding remark.

2. Introduction and/General Structure of the Goal Programming Model

Goal programming can be considered as a branch of multi-objective optimization which itself is a part of multi-criteria decision analysis. Goal programming is one of the oldest multi criteria decision making techniques used in optimization of multiple objective goals by minimizing the deviation for each of the objectives from the desired target. The basic concept of goal programming is that whether goals are attainable or not, an objective will be stated in which optimization gives a result which come as close as possible to the desired goals (satisfied solution). The objective of goal programming is to minimize the achievement of each actual goal level. If non achievement is driven to zero, then it means that actual attainment of the goal has been accomplished. For a single goal problem, the formulation and solution is similar to linear programming with the exception that, if complete goal attainment is not possible, goal programming will provide a solution and information to the decision makers.

In general, the idea of goal programming is to convert original multiple objectives into a single goal. The resulting model yields a satisficing solution which may not be optimum with respect to all the conflicting objectives of the problem. That is GP yields only an efficient and satisfactory result rather than optimum, solution to the problem. This is because, it is uncommon to always satisfy every goal, so goal programming attempts to reach a satisfactory level of the multiple objectives under consideration. To avoid the possible bias effect of the solution to different measurement unit, goal normalization takes place.

The procedures for structuring Goal programming model are similar to those for a Linear Programming. The main difference between the LP and GP is that, LP maximizes or minimize (optimizes) a single objective functions whereas, GP minimizes the deviations between the target values of the objectives and the realized results (satisficing solution). The basic steps for structuring goal programming as stated in Rifai [6] are as follows;

*Goals are discovered and converted to constraints by introducing deviational variables.

*Examine the goals to determine the exact deviational variables needed for them, *i.e.*, whether d_i^- , d_i^+ , or both as summarized below in **Table 1**.

In the second objective goal (row 2 of **Table 1**), it implies that anything below the target value b_i is acceptable, so the over-achievement of the target d_i^+ should be minimized to 0. In row three, the objective goal is that anything below the target value b_i should be driven to zero while the over-achievement of the target d_i^+ should be accepted. The last objective goal implies that anything below or above the target value b_i is unacceptable, so the over-achievement of the target d_i^+ and under achievement of the goal d_i^- should be minimized to 0.

*Goals are ranked in order of importance and pre-emptive priority factor, p_i assigned to each of them.

*In case of ties in priority, assign weights to each of the deviational variables in the priority.

Once the above steps are completed, the problem can be quantified as a GP model.

Schniederjans and Kwaks [7] referred to the most commonly applied type of goal programming as “pre-emptive weighted priority goal programming” and a generalized model for this type of programming is as follows:

Minimize:

$$z = \sum_i^m w_i p_i (d_i^- + d_i^+) \quad (2.1)$$

s.t

$$\sum_j^n a_{ij} x_{ij} + d_i^- - d_i^+ = b_i \quad (i = 1, 2, \dots, m), \quad (2.2)$$

$$x_{ij}, d_i^-, d_i^+ \geq 0, w_i > 0 \quad (2.3)$$

$$(i = 1, 2, \dots, m; j = 1, 2, 3, \dots, n) \quad (2.4)$$

The above representation can be found in some recent publications such as Rifai [6], Ken and Perushek [8], Ken and Perushek [9], Sharma and Sharma [10], Jafari *et al* [11], Orumie and Ebong [12], Nabendu and Manish [13] among others.

For each of the objectives, a target value or goal would be given (b_i), which is needed to be achieved. Finally, the undesired deviations $d = (d_i^-, d_i^+)$ from the given set of targets (b_i) are minimized by using an achievement function (z). In effect, a deviational variable represents the distance (deviation) between the aspira-

Table 1. General structure of goal programming model.

Goal	deviation variable to be minimized(included) in z
$a_{ij}x_j \leq b_i$	d_i^+
$a_{ij}x_j \geq b_i$	d_i^-
$a_{ij}x_j = b_i$	$d_i^- + d_i^+$

tion level and the actual attainment of the goal. Hence, the deviation variable d is replaced by two variables: $d = d_i^- - d_i^+$ where $d_i^-, d_i^+ \geq 0$. The preceding ensures that the deviational variables never take on negative values. The constraint ensures that one of the deviation variables will always be zero. Finally, the unwanted deviational variables need to be brought together in the form of an achievement function whose purpose is to minimize them and thus ensure that a solution that is “as close as possible” to the set of desired goals is found. This solution is called a compromised (harmonized) solution rather than optimal and that is why it is called a satisficing technique.

Weighted goal programming and Lexicographic goal programming are the most popular variants of the GP model as shown in Schniederjans and Kwak [7]), Crowder and Sposito [14], Tamize *et al.* [15], Ken and Perushek [8], Tamiz and Jones [16], Charles *et al* [17], Alp and Erosy [18] and many others. The two methods do not generate the same solution and neither is one method superior to the other because, variant is designed to satisfy certain decision makers’ preferences. Rosenthal [19] represented Goal programming problems formulations in three categories. One is called weighted goal programming where weights are assigned to the goals that measure their relative importance and then finds a solution that minimizes the weighted sum of the deviations from the targets. The second approach is called preemptive goal programming which requires ranking the goals in order of importance. The higher prioritized goal is considered first before the next ranked goal. Finally the third one is called prioritized GP which is the combination of the weighted goal programming and preemptive goal programming above. These variants are discussed below.

2.1. Pre-emptive Goal Programming (Lexicographic Goal Programming)

In many situations, however, a decision maker may not be able to determine precisely the relative importance of the goals. *I.e.* apply pre-emptive goal programming, the decision maker must rank his or her goals from the most important (goal 1) to least important (goal m). Preemptive goal programming procedure starts by concentrating on meeting the most important goal as closely as possible, before proceeding to the next higher goal, and so on to the least goal *i.e.* the objective functions are prioritized such that attainment of first goal is far more important than attainment of second goal which is far more important than attainment of third goal, etc, such that lower order goals are only achieved as long as they do not degrade the solution attained by higher priority goal. When this is the case, pre-emptive goal programming may prove to be a useful tool as introduced by Ijiri [20], and developed by many others.

The achievement function for the general preemptive GP model is given as

$$\text{lexi min } z = \sum_i^k p_i (d_i^- + d_i^+) \quad (2.5)$$

such that Equation (2.1)-(2.2) holds.

The above is detailed in Orumie and Ebong [21].

2.2. Weighted Goal Programming (WGP)

Weighted Goal Programming (WGP), weights are attached to each of the objectives to measure the relative importance of deviations from their target. WGP handles several objectives simultaneously by establishing a specific numeric goal for each of the objectives and then find a solution that comes close to each of these goals. Popular choices for normalization constants are the goal target value of the corresponding objective (hence turning all deviations into percentages) or the range of the corresponding objective (between the best and the worst possible values, hence mapping all deviations onto a zero-one range).

The algebraic formulation of a WGP as presented in Ken and Perushek [8] is given as:

$$\min z = \sum_i^m (w_i^- d_i^- + w_i^+ d_i^+) \quad (2.6)$$

such that equation (2.2)-(2.4) holds.

Where w_i^- , w_i^+ are numerical weights associated with negative and positive deviational variable d_i^- and d_i^+ (≥ 0) respectively which denotes how far the decision is from the goal and how much the decision has exceeded the goal respectively. The above is detailed in Orumie and Ebong [21].

Tamiz *et al.* [15] reported that articles that used lexicographic and weighed goal programming in pre-1990 application papers is 75% and 25% respectively. Jones and Tamiz [22] reported that their split in the period 1990-2000 is 59% lexicographic and 41% weighted. The reasons for this are due to the greater flexibility by the weighted goal programming, and the aim of the decision makers to do more “trade-off” analysis and direct comparison between goals.

2.3. Prioritized Goal Programming

This is situation where both weighted and preemptive approaches are combined to form a model to a problem. This occurs when the goals can be categorised into groups where the goals within each group are of equal importance, but there are slight differences between the groups in their level of importance. In this kind of situation, weighted goal programming can be used within each group in turn while preemptive goal programming is being applied to deal with each group in order of importance. Each priority level (each group) has a number of unwanted deviations to be minimised. This means that minimisation of deviational variables placed in a higher priority level is assumed to be infinitely more important than that of deviational variables placed in a lower priority level (group). This is represented as in equations (2.1) to (2.4).

3. Goal Programming Assumptions and General Principles (Concepts)

Winston [23] stated the axioms of linear goal programming models as follows:

- **Additivity:** additivity assumption implies that the level of penalisation for undesired deviational variables from a target level does not depend on the levels of unwanted deviational variables from the other goals.
- **Proportionality:** Proportionality assumption in the goal programming model requires that the penalisation for an unwanted deviational variable from a target level is directly proportional to the distance away from the target level.
- **Divisibility:** This assumption implies that all the decision variables should be free to take any value within their stated range, *i.e.*, a decision variable cannot be forced to take an integer or a discrete value.
- **Certainty:** This assumption implies that all the data coefficients are known with certainty.

However, the use of goal programming is not necessarily impossible if any of the above axioms is violated. A nonlinear goal programme could be formulated if the additivity condition does not hold. In the case where the divisibility axiom does not hold, an integer or binary goal programming could be formulated. When the certainty axiom is not holding, then the method to be used will depend on the type of coefficients over which there exists uncertainty. A certain amount of uncertainty over weights and target values often exists and this can frequently be handled by good sensitivity or weight analysis techniques or an interactive method. Another good alternative is to use the fuzzy goal programming variant. If there is uncertainty over the technological coefficients then either the fuzzy goal programming variant or a combination with a technique such as simulation could be used.

Nevertheless, it is necessary to see and understand the general principles and concepts of goal programming to ensure that the goal programming variants are chosen correctly and the parameters set appropriately.

- **Satisficing**

Goal programming is primarily a satisficing technique. Simon [24] describes it as a behaviour in which decision makers aim to reach a set of defined goals. If they reach those goals, then they are satisfied. This is different from the concept of optimising. He argued that human beings are more interested and able to reach goals than in the abstract concept of optimising each outcome of the decision problem. Meeting goals as closely as possible is the main aim of the goal programming technique. Tamize and Jones [25] portrayed satisficing as the prime underlying philosophy of goal programming and that its solutions should be judged solely on how well they meet the goals of the decision maker and whether they produce a practical solution to the decision problem. Although goal programming can produce Pareto-inefficient solutions, this is mainly due to poor formulation and model-

ling of the decision maker's preferences and target levels by the analyst building the goal programme.

- **Optimising**

Optimization implies looking for the decision which gives the best possible value of some measure of performance from amongst the set of possible decisions. The theory of optimising in the presence of multiple objectives is defined by adapting a concept of Pareto optimality in a multi-objective model. According to Tamize as Jones [25], optimising philosophy has importance in Goal programming in the situations where: 1) If Pareto optimality detection and restoration take place then the goal programme has a mix of the satisficing and optimising philosophies. 2) If the goals are two-sided (*i.e.* a particular value is optimal rather than a "more is better" or "less is better" situation) then the satisficing and optimising philosophies can be thought of as coinciding for those goals.

- **Ordering or Ranking**

Ordering or ranking is important in the lexicographic goal programming because, it is assumed that the ranking of the goals in order of importance to the decision maker exists and is known or able to be estimated by the decision maker. In real-life situations, goals does not take place lexicographically, and in these cases the decision maker will explore the trade-offs or the balance between the goals. In these cases lexicographic goal programming should not be used and instead another goal programming variant should be chosen.

4. Literature Review on Linear Goal Programming Algorithms

Goal programming (GP) is a popular multiobjective optimization technique used in handling problems with multiple objectives.

There are several approaches available for solving linear GP problems in the literature.

Charnes and Cooper [2] developed goal programming model for linear system (linear programming problems) in which conflicting goals were incorporated as constraints. The model is represented as;

$$\text{lexi min } z = p_i (d_i^- + d_i^+) \text{ such that } a_{ij}x_j + d_i^- - d_i^+ = b_i, (i=1, 2, \dots, m).$$

Their model is limited to those models that employ only a single objective priority. Because of the conflicting goals, it was impossible to use the concept of linear programming. Ignizio [26] tried to apply this method to solving problems he faced in the U.S. Space program, but was faced with the problem of incommensurability of the deviational variables taken from different goals which made it impossible to find weighting factors for such variables so that a meaningful summation is possible in the objective function. For instance, most of the problems were nonlinear functions. The only benefit of this model is that it uses any existing linear programming packages for solution.

Ijiri [20] applied the concept of generalized inverse approach to develop a preemptive priority levels to handle goals in their order of importance. He presented an analysis of problems in which management can quantify the goals that they wish to obtain using goal programming approach.

Ignizio [27] developed a GP solution approach, known as sequential (linear) GP method. This approach was established on the remark by Huss [28] that linear goal programming model (problem) could be solved through a sequence of conventional L.P models. This method involves the solution to an interrelated sequence of conventional linear programming problems. Based upon the solution to the previous linear programming (LP), an augmented constraint is added to the subsequent models. The purpose of this augmented constraint is to ensure that all solutions to lower priority (subsequent LP models) do not violate solutions previously obtained for higher priority goals. It utilizes any existing linear programming package. This was letter coded (computerised) by Ignizio and Perlis [29].

Lee [3] presented a method that involves a modification of the standard simplex algorithm, capable of handling pre-emptive priority goals. It is an extension of two-phase simplex algorithm known as multiphase approach. This method approaches the entire goal programming problem as a single model. The Lee's modified simplex algorithm treats the full simplex tableau, expanding the evaluation section ($z_j - c_j$) row for every pre-emptive priority and the selection rules for this algorithm follow conventional primal linear programming as described by Olson [30]. Ignizio [31] described this method as an improvement over the sequential simplex technique since it requires fewer pivots (in general) and removes the need for the construction of new constraints at each sequence. It has been shown that this algorithm takes so much time to converge.

Schniederjans and Kwaks [7] described Lee [3] and Ignizo [4] based on Dantzig [1] simplex method as the most commonly used GP solution method; but have or require columns in the simplex tableau for positive and negative deviational variables. The separate objective function rows for each priority levels add to the computa-

tional time of the solution method.

Ignizio [4] developed a GP solution approach known as multiphase (linear) GP method that handled the challenges he faced using Lee's algorithm more efficiently. He emphasized the need for the construction of new constraints for the accomplished priority level of the deviation variables for each lower level priority model. At the first stage of this procedure, the only goals included in the LP model are the first-priority goals, and the simplex method is applied in the usual way. If the resulting optimal solution is unique, we adopt it immediately without considering any additional goals. However, if there are multiple optimal solutions with the same optimal value of Z (call it Z^*), move to the second stage and add the second-priority goals to the model. If $Z^* = 0$, all the variables representing the deviations from first-priority goals must equal zero (goals are fully achieved) for the solutions remaining under consideration. On the other hand, if $Z^* \neq 0$, then, proceed to the second-priority which becomes the objective function but then, it adds the constraint that the first solution obtained equals Z^* . Apply the simplex method again, if there exist multiple optimal solutions, repeat the same process for any lower priority goals. Ignizio [32] remarked that it is an efficient method but, there is a construction of new constraint at each sequence.

Authur and Ravindran [33] developed an efficient algorithm for solving linear goal programming problems that adopts the concepts of hierarchical structure of pre-emptive models using partitioning and elimination procedures. The algorithm took advantage of the definition of ordinal pre-emptive factors in the objective function inherent in most goal programming formulations. This algorithm starts by handling only those constraints that concerns the first priority goals. If there exists multiple optimal solutions to this first priority model, constraints affecting the next higher priority are added, and the new model solved. This procedure continues until a single optimal solution is obtained. The authors solved problems of various sizes and complexities to test its efficiency with the widely used goal programming algorithm by Lee [3]. In all tested problems, the partitioning algorithm consistently did much better than the algorithm by Lee [3], taking as little as 12% of Lee's time and never more than 60%. Hwang *et al.* [34] cited Arthur and Ravindran [33] method as being an efficient solution method. Olson [30] stated that in this method, computational economies are gained by considering only rows and columns affecting the most important unsatisfied goal and that for models with few priorities, or where all goals can be satisfied, little theoretical computational advantage is expected with this method. But Schniederjans and Kwaks [7] portrayed this computational procedure as being restricted to problems that are prioritized and/or do not have conflicting goal constraints that lose variables via the variable elimination process. The goal constraints that are later augmented to an already optimized tableau as described in their procedure without an iterative adjustment run the risk of violating the original G.P. problem as demonstrated by Schniederjans and Kwaks [7].

Ignizio [31] presented a general goal programming mathematical model that is different from the traditional linear programming formulation, but provided a real problem representation in practice. His model is represented thus:

Find $\bar{x} = x_1, x_2, \dots, x_j$, so as to minimize:

$$\bar{a} = \{g_1(\bar{n}, \bar{p}), g_2(\bar{n}, \bar{p}), \dots, g_k(\bar{n}, \bar{p})\} \quad (4.1)$$

such that:

$$f_i(\bar{x}) + n_i - p_i = b_i \text{ for all } i = 1, \dots, m \quad (4.2)$$

and

$$\bar{x}, \bar{n}, \bar{p} \geq 0 \quad (4.3)$$

where: x_j is the j^{th} decision variable, \bar{a} is denoted as the achievement function; a row vector measure of the attainment of the objectives or constraints at each priority level, $g_k(\bar{n}, \bar{p})$ is a function (normally linear) of the deviation variables associated with the objectives or constraints at priority level k , k is the total number of priority levels in the model, b_i is the right-hand side constant for goal (or constraint) i , $f_i(\bar{x})$ is the left-hand side of the linear or nonlinear goal or constraint i . where n_i represents the negative deviation from goal i and p_i is the positive deviation from goal i .

Ignizio [35] developed another approach for solving GPP that compresses tableau element with the use of the condensed simplex tableau alongside with the concept of column dropping. He maintained the negative deviational variables in the basis. The interior rows of the tableau are associated with the present set of basic variables and the interior columns with the present set of non-basic variables. The bottom row of the tableau, under the

non-basic variables, gives the shadow prices for the priority level under consideration. Check marks (\sphericalangle) are placed above the columns which are no longer eligible for entry into the basis. He also used the concept of reflected p -space to reduce storage. But, Schniederjans and Kwak [36] reported that, Ignizio [35] algorithm takes more computational time manipulation than the Schniederjans and Kwak [37] algorithm, and also fails to generate useful information that is commonly found in more popular G.P. algorithms. In particular, the Ignizio [35] algorithm does not provide any information for subsequent parameter sensitivity analysis. He concluded that, researchers who are interested in using a G.P. algorithm requiring the least number of tableau elements or readers who are interested in maximizing the informational value of a G.P. solution should not refer to Ignizio [35] method.

Schniederjans and Kwaks [7] provided a new approach for goal programming problems solution based on Baumol's [37] simplex method that fully utilized positive deviational variable, together with a step-by-step solution of an illustrative example. Their method yields a substantial reduction in the number of tableau elements computations. At that time no computer program existed using the proposed goal programming method and they recommended that such a computer program be developed by interested researchers. Olson [30] stated that Schniederjans and Kwak's solution procedure eliminates up to one half of the deviational variable columns in the simplex basis relative to Lee's full simplex approach, and does away with the $z_j - c_j$ section of the tableau, also, computational efficiency is gained by not maintaining the identity matrix column elements in the simplex tableau, and if the optimal solution includes a high proportion of positive deviational variables, this method can be expected to be relatively faster than other approaches. He also stated that the algorithm however, does not follow a path of guaranteed solution improvement if the solution contains a large proportion of negative deviational variables; this method requires extra computational time and cycling often occurred in models where an unsatisfied goal was found in the final solution (see Orumie and Ebong [21]). But Ignizio [38] argued that Schniederjans and Kwaks paper is incorrectly compared, since the correct comparison indicates that the new algorithm requires tableaux of the same sizes as are required of those for the older, elementary forms, and that the fact that there is no justification that the Baumol method for Linear Programming is more efficient than the simplex method disqualifies their argument.

Olson [30] compared four goal programming methods; Schniederjans and kwaks [7], Lee [3], Athur and Ravindran [33], and Olson [30]. And also developed a revised simplex algorithm (RSM) for solving LGP problem that adopts Schiederjans and Kwaks [7] dual simplex rules applied for the calculation of new tableau elements. The algorithm utilizes efficiency of not maintaining identity columns while gaining the systematic security of searching for optimal solution utilizing only feasible solution. Negative deviational variables are fully maintained, but other variables columns are developed only as necessary. The RSM works well when negative deviational variable is in the solution. In his result tests, the dual simplex method appears to have superior computational times for models with a large proportion of positive deviational variables in the solution, whereas the revised simplex algorithm appears more consistent in time and accuracy for general goal programming models. He summarized that the Lee algorithm proved accurate in models tested, although extra iterations were required in some models, whereas the Arthur and Ravindran code was not tampered with other than to increase dimensions owing to unfamiliarity with the specific program (see Ebong and Orumie [21]). According to him, accuracy was found to be a problem for the code in larger models tested. The Arthur and Ravindran code was competitive in time for those models where it obtained the correct solution, but for larger models run, incorrect solutions were obtained. The computational efficiency of the dual simplex code presented by Schniederjans and Kwak over the revised simplex code can be substantial for models involving solutions with a high proportion of positive deviational variables in the solution as described by the author.

Ignizio [32] developed another approach for solving GP problems called multidimensional dual simplex algorithm (MDD). The dual of a lexicographic GP is similar to that of a linear programming with the exception that the right hand sides of the dual are multi-dimensional and ranked in order of important. This follows from the fact that the achievement function in the primal are ranked lexicographically. However, since it has been remarked that the multi-dimensional dual is a linear programming problem with multiple and prioritized right-hand sides, each model is identical to one another, with the exceptions that: the right-hand side will be changed at each sequence and some constraints dropped depending on the solution obtained to the previous linear programmes. All non-binding dual constraints corresponding to primal variables are removed as illustrated by the author. According to the author, one might conjecture, that the solution to a multidimensional dual with, say, five right-hand sides would require about five times as long as that needed for the simplex solution to the initial

model and that an attractive feature of the dual simplex-based algorithm for obtaining the solution of the LGP problem is that it can be implemented easily with any conventional simplex software system (single-objective function). But Crowder and Sposito [14] disagreed with his method and argued that removal of non-binding constraints in the dual problem after obtaining the optimum for the dual problem associated with priority level i , coincides with the removal of non-basic variables in the LGP primal at priority level i . This implies that pre-emptive priority conditions are violated while solving MDD problem, otherwise wrong solution will be obtained. They supported their claim by solving a LGP problem as shown in Crowder and Sposito [14] which yields $a_0^1 = (0,1,2)$, $x_1^0 = 6$, $x_2^0 = 0$ when solved using SLGP. But, if the same problem is solved using the multidimensional-dual by Ignizio [32], changing right-hand sides and removing slack rows at each priority level as outlined, will lead to incorrect solution: $a_0^1 = (0,1,0)$, $x_1^0 = 4$, $x_2^0 = 0$.

Shim and Chun [39] showed that Resource Planning and Management Systems (RPMS) network approach can be used to solve goal programming (GP) problem. In RPMS-based GP, deviational variables are represented by resource nodes. These resources, with different priorities and weights, are used to attain maximum goal achievement, whereas resources are consumed to produce products at process nodes (decision variables). The nodes to be placed in the initial RPMS graph are process nodes, minimum/maximum nodes and resource nodes, except positive deviational variables. The interrelations (technological coefficient) are represented by arrows between process nodes and resource nodes, and between different resource nodes. The interrelations between resource nodes and the maximum node indicate the priority structure, while the interrelations between resource nodes and the minimum node stand for goal levels. Values below the x 's at resource nodes mean remaining resources for negative deviational variables and resources consumed for positive deviational variables, respectively. The positive deviational variables work as process nodes with priority assigned to it. But Ogryczak [40] argued that RPM defers from GP formulation since it utilizes negative weight and additional regularization of the min max aggregation. He argued also that RPM is an iterative technique and pointed out serious flow since practical large problem usually have multiple optimal solution.

Calvete and Mateo [41] used the ideas of primal dual algorithms for the minimum cost generalized network flow (MGNF) problem to develop a lexicographic optimization of multi objective generalized network flow (LGNF) problem. The author showed that their algorithm is efficient in reaching optimality, but difficult in labeling process due to several nodes, arcs, paths which result in multiple solutions.

Baykasoglu *et al.* [42] developed an algorithm for solving linear GP using multiple objective tabu search (TS). Their computational experimental result obtained showed that their method is efficient. This algorithm may recycle old solutions and become trapped in a loop as indicated by the author, which implies that it does not solve all kinds of goal functions and constraints. The procedure is very tedious as it works with more than one solution vector.

Baykasoglu [43] used the multiple objective tabu search (MOTS) algorithm, which was proposed previously by Baykasoglu *et al.* [42] to solve GP models. In the proposed approach, GP models are first converted to their classical Multi objective optimization (MOO) equivalent by using some simple conversion procedures. Then the problem is solved using the MOTS algorithm. The proposed approach also avoids the problem of weighting the goals in weighted GP and ordering the goals in pre-emptive GP. The results obtained from the test problems showed that MOTS can successfully find many alternative solutions to a given GP. To enable the TS algorithm to work with more than one objective, selection and updating stages of the basic TS are redefined.

Kasana [44] developed grouping algorithm for LGPP solution. This method considers all goals and real constraints together as one group with the objective function being the sum of all the unwanted deviations, and solves a sequence of LP sub problems, each using the optimal solution of the previous sub problems. This algorithm is being dominated by the partitioning method as indicated by the author. He indicated that it is good and performs well only if a large number of goals are satisfied. In other word, if an unsatisfied goal is in the final tableau, it is inefficient. It utilized sequential method (see Orumie and Ebong [21]).

Orumie and Ebong [12] developed an efficient method of solving a generalized GPP that reduced the computational time drastically when compared with the existing ones. Their method recognised the fact that a goal programming model may include rigid constraints like $\sum_j^n a_{ij}x_j \leq b_i$ *i.e.* Goal programming also allows for an addition of a set of linear programming style hard constraints whose violation will make the solution infeasible.

ble. Their new goal programming algorithm is formulated into initial tableau in the same format with that of simplex method of solving linear programming problems whether it is weighted, prioritized or generalized model. The different is that in the new algorithm, the deviational variables that appeared in the achievement functions with their priority or weight attached to them, together with slacks from the rigid constraints if exists forms the basis. If both d_i^v (positive and negative deviational variables) from the same row are in the achievement function z , then the one with highest priority or weight will be in the basis whereas the lesser one will be placed at the non bases. The column of the decision variables, the slacks variables from the rigid constraints, and the deviational variables from the achievement functions forms the non basic variables. The procedure considers the goal constraints (g_i) as both objective functions and the constraints. It starts by not including the deviational variables column that did not appear in the achievement function in the tableau (*i.e.* while searching for the optimal solution), but others can be augmented when necessary. This is because, positive deviational variables columns coefficient is the same as negative of the negative deviational column coefficient.

Iskander [45] suggested an approach for Solving Weighted Goal Programming Problem that reformulated the weighted goal program as a lexicographic goal program with two main goals. The first goal, which has the first priority, seeks to minimize the maximum weighted undesired normalized deviation. The second goal, having the second priority, minimizes the sum of the undesired normalized deviations. The proposed approach seeks to provide a solution in which the goals achievements are proportionally related to the relative weights.

Orumie and Ebong [21] separated their generalized algorithm of 2011 into lexicographic and weighted method. Their algorithm is very efficient in reaching optimality.

However, debated weakness of LGP is that goal-programming approach, regardless of the weighting structures (pre-emptive or Archimedean) and regardless of the goals (one-sided or two-sided), can lead to inferior (dominated or inefficient), suboptimal solutions which are not necessarily the “best” ones available to the decision-maker. But, in Ignizio [4], it was proved that the optimal solution obtained by the lexicographic problem is Pareto optimal. Thus, the lexicographic method is always adopted as an additional optimization approach in methods that can only guarantee weak optimality by themselves. Evans [46] described GP problem as a technique for finding that solution which minimizes the deviation over all feasible solutions; such a solution is called a best compromise solution and that under the assumption that more of each objective is preferred to less; a best compromise solution. Min and Storbeck [47] stated that GP is a technique not designed to find an “optimal point”, but to find an “acceptable range”, and advised that the dispute of GP dominance will continue unless the management scientist can accept goal programming’s satisficing principle and not being captivated by the principle of optimality. Miettinen [48] proved that GP technique yields nondominated solutions if the goal point is chosen in the feasible domain. However, in Goal programming, there is no method to determine if a solution is better than other. Nabendu and Manish [13] stated that the computational procedure in goal programming is to select a set of solutions which satisfies the constraints and providing a satisfactory goal, ranked in priority order since GP approach seeks satisficing solutions which come as close to the desired aspiration levels as possible. Antonio *et al.* [49] described Pareto dominance relation as the most commonly adopted method in multi objective optimization to compare solutions which, instead of a single optimal solution, leads to a set of alternatives with different trade-offs among the objectives. Their solutions are called Pareto optimal solutions or non-dominated solutions. Although there are multiple Pareto optimal solutions, in practice, only one solution has to be selected for implementation.

Hannan [50] and [51], and Romero [52] developed an approach for detection of Pareto inefficiency in a goal programming solution. These methods provide ways of restoring Pareto efficiency by calculating a Pareto-efficient solution that dominates the goal programming solution. These algorithms carry out a maximisation of some function of the wanted deviational variables (that are not penalized in the objective function) in the original goal programming, such that if the optimal solution from that of the original goal programme is altered indicated that the original goal programming solution was Pareto inefficient. Tamiz and Jone [53] separated detection and restoration procedure, and showed that Pareto detection can be performed without resorting to optimization in five tests as described in Tamiz and Jone [25]. In all the five steps, algorithm terminates with only degenerate iterations having taken place. Hence the solution point is still the same with the original goal programme solution. Tamiz and Jones [25] describe their approach as the most computationally efficient means of detecting Pareto inefficiency.

Hannan [50] method presented different possible ways of calculating a Pareto-efficient solution. These include; 1) A possible weighting scheme to give relative importance to the improvement of the objectives. 2) A

priority order in which the objectives should be improved. 3) A vector maximisation multiple objective programming model to find a set of efficient solutions that dominate the original goal programme's solution. But Romero [52] pointed out that the Hannan [50] formulations do not restrict the values of deviational variables which are not penalised in the original formulation correctly, and thus generates worst solution for such an objective, and hence does not dominate the original goal programme. Romero [52] modeled the original goal programming solution and Pareto restoration as a single lexicographic process.

Tamiz and Jones [53] pointed out a further complication that occurred in the case where one or more objectives are Pareto unbounded. In this case the restoration optimisation will become unbounded and an efficient solution is not possible. In this case the objectives that are unbounded will be discarded from the final priority level and the optimisation continued from that point. This step may continue iteratively as not all the unbounded objectives may be detected at that point. The method will terminate with all objectives classified as either Pareto efficient or unbounded. This is the closest to efficiency that can be reached for a Pareto-unbounded model. Tamiz and Jones [53] presented three ways of restoring Pareto efficiency: Straight restoration, Preference-based restoration and Interactive restoration.

Another debated issue on goal-programming approach is unbounded solution problem. But Ignizio [4], Marowski and Ignizio [54], and Schniederjans [55], argued that unbounded problems did not exist in GP because aspiration levels (target values or right-hand-side values) were associated with every objective and GP originally sought a satisficing solution which allowed for some flexibility in aspiration levels as stated in Min and Storbeck [47]. Evans [46] showed that GP solution cannot be bounded since it has only minimum values. Therefore, they concluded that GP problems could not generate unbounded solutions unlike in LP which does not allow any flexibility in determining right-hand-side values of feasibility constraints; hence an unbounded solution can occur in LP formulations. GP is much more flexible in determining right-hand-side values than is LP. The major appeal of GP as explained in Min and Storbeck [47] is that it considers the cognitive nature of much human behaviour which can be irrational or non-omniscient.

Another issue on goal programming approach is feasibility. It has been proved that when the goals are feasible, the solution given by the GP is efficient, but the efficiency of the provided solution when the goals are not feasible remains an open problem in general as illustrated by Ontario, *et al.* [56]. An efficient solution is a feasible solution, if there does not exist any other feasible solution which does at least as well on every single objective, and better on at least one objective. Weighted goal programming and preemptive goal programming provide pareto optimal solutions if the goals form a pareto optimal point or if all deviation variables, d_j^+ for functions being increased and d_j^- for functions being reduced, have positive values at the optimum as illustrated by Ken and Perushek [9]. This was repeated in Marler and Arora [57]. Major advantage of goal programming is that there always exists a solution to the problem, provided that it has feasible region and this is because of the inclusion of the deviational variables.

5. Conclusions

This paper presented a survey of current methods for LGP and limitation of LGP in general. The question that one would ask at this point is: "Which procedure is the best?" Orumie and Ebong (2013) have compared various LGP techniques. Their algorithm was compared in terms of accuracy and time requirements with existing algorithms by Lee [3] and by Ignizio [4] and Ignizio [35]. Computational times for 10 goal programming models of various sizes and complexities were compared. Number of iteration per problem, total entries per problems is used as benchmark for the comparisons. The new method by Orumie and Ebong (2011) have better computational times in all the problem solution and proved the best since there is a reduction in computational time in all the problems solved.

The controversies surrounding LGP mostly came from misconceptions about the principle of satisficing which underlies GP theories. It is almost impossible for the decision-maker to achieve "ideal" goals without the expense of other goals in optimization of multiple goals. In this sense, the multi-objective is an unfortunate name. Therefore, the efficiency of GP solutions is "problem-dependent" and "user-dependent". In the modelling and solution processes of GP, so much freedom is given to the decision-maker. If the decision-maker inadvertently sets unreasonable targets or assigns incorrect weights and/or priorities, a GP solution cannot provide the best available or efficient solution. Therefore, the limitations of GP, if any, are due mainly to errors of its users, not to the rationale behind GP theories. However, there is no doubt that further developments of GP theories are

no longer needed.

Hence, the basic concept of goal programming is that whether goals are attainable or not, an objective will be stated in which optimization gives a result which come as close as possible to the desired goals (satisfied solution). GP technique yields non-dominated solutions if the goal point is chosen in the feasible domain. GP solution cannot be bounded since it has only minimum values. It has been proved that when the goals are feasible, the solution given by the GP is efficient, but the efficiency of the provided solution when the goals are not feasible remains an open problem. Major advantage of goal programming is that there always exists a solution to the problem, provided that it has feasible region and this is because of the inclusion of the deviational variables.

References

- [1] Dantzig, G.B. (1948) Programming in a Linear Structure. Comptroller, United States Air Force, Washington DC.
- [2] Charnes, A. and Cooper, W.W. (1961) Management Models and the Industrial Applications of Linear Programming. John Wiley, New York.
- [3] Lee, S.M. (1972) Goal Programming for Decision Analysis. Auerbach, Philadelphia.
- [4] Ignizio, J.P. (1976) Goal Programming and Extensions. D. C. Heath and Company, Lexington.
- [5] Lin, W.T. (1980) A Survey of Goal Programming Applications. *Omega*, **8**, 115-117.
[http://dx.doi.org/10.1016/0305-0483\(80\)90047-X](http://dx.doi.org/10.1016/0305-0483(80)90047-X)
- [6] Rifai, A.K. (1996) A Note on the Structure of the Goal-Programming Model: Assessment and Evaluation. *International Journal of Operations and Production Management*, **16**, 40-49.
<http://dx.doi.org/10.1108/01443579610106355>
- [7] Schniederjans, M.J. and Kwak, N.K. (1982) An Alternative Method for Solving Goal Programming Problems: A Reply. *The Journal of the Operational Research Society*, **33**, 859-860.
- [8] Ken, W. and Perushek, D.E. (1996) Linear Goal Programming for Academic Library Acquisitions Allocations.
http://trace.tennessee.edu/utk_libfpubs/26
- [9] Ken, W. and Perushek, D.E. (2000) Goal Programming as a Solution. http://trace.tennessee.edu/utk_libfpubs
- [10] Sharma, H.P. and Sharma, D.K. (2006) A Multi-Objective Decision-Making Approach For Mutual Fund Portfolio. *Journal of Business & Economics Research*, **4**, 13-24.
- [11] Jafari, H., Koshteli, R. and Khabiri, B. (2008) An Optimal Model using Goal Programming for Rice Farm. *Applied Mathematical Sciences*, **2**, 1131-1136.
- [12] Orumie, U.C. and Ebong, D.W. (2011) An Alternative Method of Solving Goal Programming Problem. *Nigerian Journal of Operations Research*, **2**, 68-90.
- [13] Nabendu, S. and Manish, N. (2012) A Goal Programming Approach to Rubber Plantation Planning in Tripura. *Applied Mathematical Sciences*, **6**, 6171-6179.
- [14] Crowder, L.J and Sposito, V.A. (1987) Comments on “An Algorithm for Solving the Linear Goal-Programming Problem by Solving Its Dual”. *The Journal of the Operational Research Society*, **38**, 335-340.
- [15] Tamiz, M., Jones, D.F. and El-darzin, E. (1995) A Review of Goal Programming and Its Applications. *Annals of operation Research*, **58**, 39-53.
- [16] Tamiz, M. and Jones, F. (1997) Interactive Frame Works for Investigating of Goal Programming Models. Theory and Practice. *Journal of Multi-Criteria Decision Analysis*, **6**, 52-60.
[http://dx.doi.org/10.1002/\(SICI\)1099-1360\(199701\)6:1<52::AID-MCDA124>3.0.CO;2-3](http://dx.doi.org/10.1002/(SICI)1099-1360(199701)6:1<52::AID-MCDA124>3.0.CO;2-3)
- [17] Charles, H.F., Boaz, G. and Hussein, N. (2005) Modelling Tradeoffs in Three-Dimensional Concurrent Engineering: A Goal Programming Approach. *Journal of Operations Management*, **23**, 389-403.
<http://dx.doi.org/10.1016/j.jom.2004.09.005>
- [18] Alp, S., Yavuz, E. and Ersoy, N. (2011) Using Linear Goal Programming in Surveying Engineering for Vertical Network Adjustment. *International Journal of the Physical Sciences*, **6**, 1982-1987.
- [19] Rosenthal, R. E. (1983) Goal Programming—A Critique. *NZOR*, **11**, 8.
- [20] Ijiri, Y. (1965) Management Goals and Accounting for Control. Rand-McNally, Chicago.
- [21] Orumie, U.C. and Ebong, D.W. (2013) An Efficient Method of Solving Lexicographic Linear Goal Programming Problem. *International Journal of Scientific and Research Publications*, **3**, 1-8.
- [22] Tamiz, M. and Jones, D.F. (2002) Goal Programming in the Period 1990-2000. In: Ehrgott, M. and Gandibleux, X., Eds., *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, Kluwer, 129-170.
- [23] Winston, W. (2004) Operations Research: Applications and Algorithms, Duxbury Press, Pacific Grove.

- [24] Simon, H.A. (1957) *Models of Man*. Wiley & Sons, New York.
- [25] Tamiz, M. and Jones, D.F. (2010) *Practical Goal Programming*. International Series in Operations Research & Management Science, Springer, New York. <http://www.springer.com/series/6161>
- [26] Ignizio, J.P. (1966) Adaptive Antenna Array Study. Boeing Company, RWA-5557.
- [27] Ignizio, J. P. (1967) A FORTRAN Code for Multiple Objective I.P. North American Aviation Internal Memorandum.
- [28] Huss, P. (1967) Telephone Communications of January.
- [29] Ignizio J.P. and Perlis, J.H. (1979) Sequential Linear Goal Programming: Implementation via MPSX. *Computers and Operations Research*, **6**, 141-145. [http://dx.doi.org/10.1016/0305-0548\(79\)90026-1](http://dx.doi.org/10.1016/0305-0548(79)90026-1)
- [30] Olson, D.L. (1984) Comparison of Four Goal Programming Algorithms. *Journal of the Operational Research Society*, **35**, 347-354.
- [31] Ignizio, J.P. (1978) A Review of Goal Programming: A Tool for Multiobjective Analysis. *The Journal of the Operational Research Society*, **29**, 1109-1119.
- [32] Ignizio, J.P. (1985) An Algorithm for Solving the Linear Goal-Programming Problem by Solving Its Dual. *Journal of operational Research Society*, **36**, 507-5 15.
- [33] Arthur, J.L. and Ravindran, A. (1978) An Efficient Goal Programming Algorithm Using Constraint Partitioning and Variable Elimination. *Management Science*, **24**, 867-886.
- [34] Hwang, C.L., Masud, A.S.M., Paidy, S.R. and Yoon, K. (1980) Mathematical Programming with Multiple Objectives: A Tutorial. *Computers & Operations Research*, **7**, 5-31.
- [35] Ignizio, J.P. (1982) *Linear Programming in Single and Multiple Objective System*. Prentice Hall, Upper Saddle River, 408-410.
- [36] Schniederjans, M.J. and Kwak, N.K. (1982) An Alternative Method for Solving Goal Programming Problems: A Reply. *The Journal of the Operational Research Society*, **33**, 859-860.
- [37] Baumol, W.J. (1965) *Economic Theory and Operations Analysis*. 2nd Edition, Prentice-Hall, Englewood Cliffs.
- [38] Ignizio, J.P. (1983) A Note on Computational Methods in Lexicographic Linear Goal Programming. *The Journal of the Operational Research Society*, **34**, 539-542.
- [39] Shim, J.P. and Chun, S.G. (1991) Goal Programming: The RPMS Network Approach. *The Journal of the Operational Research Society*, **42**, 83-93.
- [40] Ogryczak, W. (2001) Comments on Romero, C., Tamiz, M. and Jones, D.F. (1998) Goal Programming, Com- promise Programming and Reference Point Method Formulations Linkages and Utility Interpretations. *The Journal of the Operational Research Society*, **52**, 960-962.
- [41] Calvete, H.I. and Mateo, P.M. (1998) Lexicographic Optimisation in Generalised Network Flow Problems. *Journal of the Operational Research Society*, **49**, 519-529.
- [42] Baykasoglu, A., Owen, S. and Gindy, N. (1999) Solution Of Goal Programming Models Using a Basic Taboo Search Algorithm. *Journal of Operational Research Society*, **50**, 960-973.
- [43] Baykasoglu, A. (2001) Goal Programming Using Multiple Objective Tabu Search. *The Journal of the Operational Research Society*, **52**, 1359-1369. <http://dx.doi.org/10.1057/palgrave.jors.2601229>
- [44] Kasana, H.S. (2003) Grouping Algorithm for Linear Goal Programming Problems. *Asia Pacific Journal of Operational Research*, **20**, 191-220.
- [45] Iskander, M.G. (2012) A Suggested Approach for Solving Weighted Goal Programming Problem. *American Journal of Computational and Applied Mathematics*, **2**, 55-57. <http://dx.doi.org/10.5923/j.ajcam.20120202.10>
- [46] Evans, G.W. (1984) An Overview of Technique for Solving Multiobjective Mathematical Programs. *Management Science*, **30**, 1268-1282.
- [47] Min, H. and Storbeck, J. (1991) On the Origin and Persistence of Misconceptions in Goal Programming. *Journal of the Operational Research Society*, **42**, 301-312.
- [48] Miettinen, K.M. (1998) *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston. <http://dx.doi.org/10.1007/978-1-4615-5563-6>
- [49] Antonio, L.J., Martinez, S.Z. and Coello, C.A. (2009) *An Introduction to Multiobjective Optimization Techniques*. Nova Science Publishers, Inc., Hauppauge, 1-26.
- [50] Hannan, E.L. (1980) Nondominance in Goal Programming. *INFORMATION*, **18**, 300-309.
- [51] Hannan, E.L. (1981) On Fuzzy Goal Programming. *Decision Sciences*, **12**, 522-531. <http://dx.doi.org/10.1111/j.1540-5915.1981.tb00102.x>
- [52] Romero, C. (1991) On Misconceptions in Goal Programming. *The Journal of the Operational Research Society*, **42**,

927-928.

- [53] Tamiz, M. and Jones, D.F. (1996) Goal Programming and Pareto Efficiency. *Journal of Information and Optimization Sciences*, **17**, 291-307. <http://dx.doi.org/10.1080/02522667.1996.10699283>
- [54] Markowski, C.A. and Ignizio, J.P. (1983) Theory and Properties of the Lexicographic Linear Goal Programming. *Large Scale System*, **5**, 115-121.
- [55] Schniederjans, M.J. (1984) Linear Goal Programming. Petrocelli, Princeton.
- [56] Larbani, M. and Aouni, B. (2007) On the Pareto Optimality in Goal Programming. ASAC, Ottawa.
- [57] Marler, R.T. and Arora, J.S. (2004) Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, **26**, 369-395. <http://dx.doi.org/10.1007/s00158-003-0368-6>