PERGAMON

Omega, Int. J. Mgmt Sci. 27 (1999) 179–188

omega
The International Journal of Management Science

Extensions of Pareto efficiency analysis to integer goal programming

M. Tamiz *, S.K. Mirrazavi, D.F. Jones

University of Portsmouth, Portsmouth, UK

Received 1 February 1998; accepted 1 May 1998

Abstract

This paper focuses on the design, development and implementation of new Pareto efficiency detection and restoration techniques for integer goal programming. The design of the algorithms and their implementation issues within (an otherwise continuous) goal programming system are detailed. The differences between continuous and integer goal programming regarding Pareto efficiency detection and restoration analysis are described. The integer Pareto efficiency techniques have been applied to a selection of problems from different industrial contexts in order to assess their computational performance. Finally, Pareto restoration and detection techniques are applied to an integer goal programming problem to illustrate the methodology. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Multi-objective programming; Integer goal programming; Pareto efficiency; Intelligent analysis tools

1. Introduction

Goal programming (GP) is a multi-objective programming technique first developed by Charnes et al. in 1955 [1] and more explicitly defined by Charnes and Cooper in 1961 [2]. GP can be considered as a mathematical programming method and a member of the multi-criteria decision making (MCDM) family, and is known as a distinguished and effective method of problem solving in this field. Research on the development of the theoretical and operational aspects of GP is extensive. A small subset of key references are Refs. [6–11, 15–18, 21, 22].

This research is concerned with weighted and lexicographic integer GP.

● Weighted integer GP: minimises a weighted sum of unwanted deviations from the decision maker's set of targets for a number of goals (criteria). All goals are therefore considered simultaneously.

The general mathematical representation of a weighted GP model has the form;

$$\text{Min } z = \sum_{i=1}^{k}(u_i n_i + v_i p_i) \quad (1)$$

subject to,

$$f_i(\mathbf{x}) + n_i - p_i = b_i, \quad i = 1, \ldots, k \quad (2)$$

$$\mathbf{x} \in \mathbf{C}_s \quad (3)$$

$$\mathbf{n}, \mathbf{p} \geq 0 \quad (4)$$

$$\mathbf{x} \geq 0 \text{ and integer} \quad (5)$$

where $f_i(x)$ is a linear function(objective) of $\mathbf{x}$, and $\mathbf{x}$ is the set of decision variables to be determined. b_i is the target value for objective i. n_i and p_i represent the negative and positive deviations from this target value, respectively. u_i and v_i are the respective non-negative weights attached to these deviations in the achievement function z. $\mathbf{C}_s$ is an optional set of hard constraints.

● Lexicographic integer GP: minimises a ranked vector (in order of decision makers importance) of unwanted deviations from a set of targets for a num-

* Corresponding author. Tel.: +44-1705-843108; Fax: +44-1705-843106; E-mail: mehrdad.tamiz@port.ac.uk.

PII: S0305-0483(98)00038-3

ber of objectives, where different goals are grouped into several levels of priorities. There exists a natural ordering amongst the goals. Goals in the higher priority levels are satisfied as closely as possible and it is only then that goals in the lower priority levels are considered, i.e., a sequential minimisation of priority levels with no degradation in the value of higher priority levels. The mathematical representation of an integer lexicographic GP model has the form;

$$\text{Lex Min } \mathbf{a} = \left[\sum_{i=1}^{k} (u_i^1 n_i + v_i^1 p_i), \ \ldots, \ \sum_{i=1}^{k} (u_i^L n_i + v_i^L p_i) \right] \tag{6}$$

Subject to,

$$f_i(\mathbf{x}) + n_i - p_i = b_i, \quad i = 1, \ \ldots, k \tag{7}$$

$$\mathbf{x} \in \mathbf{C}_s \tag{8}$$

$$\mathbf{n}, \mathbf{p} \geq 0 \tag{9}$$

$$\mathbf{x} \geq 0 \text{ and integer} \tag{10}$$

where $f_i(x)$ is a linear function(objective) of $\mathbf{x}$, and $\mathbf{x}$ is the set of decision variables to be determined. b_i is the target value for objective i. n_i and p_i represent the negative and positive deviations from this target value, respectively. This model has L priority levels, and k objectives. $\mathbf{a}$ is an ordered vector of L priority levels. u_i^l and v_i^l are the respective weights attached to the deviations in the lth priority level of the achievement function. $\mathbf{C}_s$ is the optional set of hard constraints.

GPSYS [20] and IGPSYS [13] are the goal programming and integer goal programming systems used in this research, respectively.

A well known fact that has caused much debate over recent years is that GP has the ability to produce Pareto inefficient or dominated solutions. A standard GP optimum solution (initial optimal solution) is not guaranteed to be Pareto efficient, which according to Zeleny [24] is probably the most contentious quality of GP.

Vilfredo Pareto [14] introduced the concept of Pareto optimality in the field of economics in 1896. According to his definition, a society is Pareto optimal (Pareto efficient) when no member of that society can improve their condition without lowering the condition of another member.

The concept of Pareto optimality can be applied to GP in order to build succinct tools to overcome these intrinsic deficiencies. In a GP environment, Pareto optimality/efficiency is defined as the state in which no objective can be improved without degrading another objective. Improvement can be thought of as obtaining a better level of satisfaction of the objective irrespective of the target value, conversely degradation implies a worsening of the satisfaction level. The reason for the fact that GP models can produce Pareto inefficient solutions is that the decision maker may set target values which are too pessimistic, i.e., objectives which are easily achieved with respect to the restrictions (constraints and conflicting objectives) imposed. This disadvantage has, in the past, caused great concern and doubt regarding the use of GP, as detailed in Refs. [23, 24].

To overcome this drawback, Hannan [4, 5] proposed a remedy to restore Pareto efficiency. His method is based on the production of a set of efficient points which dominate the standard inefficient GP optimisation solution point. Further developments to Hannan's method were carried out by Romero [15], in order to generate efficient solutions, while preventing the degradation of any objective's achieved value from the standard inefficient GP solution point. Tamiz and Jones [19] propose an alternative technique for Pareto efficiency and inefficiency detection and implement it within a GP optimisation package GPSYS [20]. This technique consists of a series of tests which are designed to categorise objectives into Pareto efficient, inefficient or unbounded states [19]. These tests investigate the possibility of improving the objectives from the initial optimal solution in order to detect efficiency or inefficiency. The examination and inspection of the GP optimal solution using mathematical programming simplex tableaux theory takes place in order to perform the tests. Only degenerate simplex iterations are performed and thus the initial GP solution remains unchanged and no movement occurs.

Fig. 1 shows a simple GP problem illustrating the case of Pareto efficiency in the continuous case. The problem has two objectives, OBJ1 and OBJ2, and a hard constraint. The shaded area represents the feasible region and z^* is the initial GP optimal solution for the achievement function of

$$\text{Min } z = n_1 + n_2 \tag{11}$$

It is clear that both objectives can be improved without degrading the other, resulting ultimately in detection/restoration of efficient feasible points z_1 and z_2. Thus, point z^* is classified as a Pareto inefficient point dominated by a set of Pareto efficient points (Pareto efficient boundary). Due to the characteristics of the simplex algorithm, only the two extreme points z_1 and z_2, can be located.

Tamiz and Jones [19] also investigate the restoration of Pareto efficient points whereby the decision maker may have a preference for the restoration of objectives, i.e., he/she may be more concerned about improving certain Pareto inefficient objectives than the others.

The methods proposed in [5, 15, 19] overcome Zeleny's major criticism of continuous GP. However

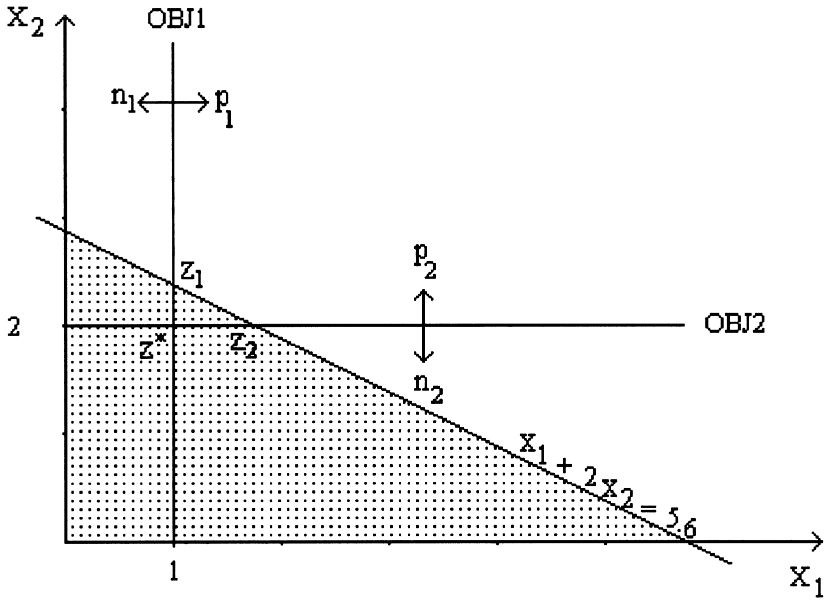

Fig. 1. Pareto efficiency in a continuous case.

the case of detection and restoration of efficient solutions in integer goal programming has not yet been fully addressed or resolved.

In integer GP an objective is Pareto inefficient if a different *integer* solution can be found that improves that objective without degrading the value of any other objective. If no such point exists, then that objective is termed Pareto efficient. A variant of a Pareto inefficient objective is when the objective can be increased to an infinitely high value without degrading other objectives i.e., Pareto unbounded. An integer GP is said to be Pareto inefficient or unbounded when one or more objectives are inefficient or unbounded, respectively. It follows that in a Pareto efficient integer GP, every objective must be Pareto efficient.

The methods developed for the continuous case do not guarantee the identification of inefficient *integer* GP objectives, let alone the restoration of an integer efficient point. For example in Fig. 1, z^* is a GP Pareto inefficient point in the continuous case, but it is an integer Pareto efficient point since there are no other integer points in the $z^*z_1z_2$ domain. The technique in Ref. [19] examines and inspects the simplex solution space and is based on performing degenerate iterations only. The existence of non-degenerate simplex iterations proves the inefficiency of the solution. To detect an improved integer point, simplex iterations as well as branch and bounding [13] must be performed to confirm the existence or non-existence of an integer point. In this paper, methods in Refs. [5, 15, 19] are thus built upon to develop a new algorithm to overcome the integer requirements of integer GP inefficiency [13].

To clarify the points mentioned above, consider the integer GP problem depicted in Fig. 2. Point z^* is the integer GP inefficient point. Continuous Pareto efficiency detection techniques would not be able to detect the integer efficiency or inefficiency of z^*, since the selection of dominating points in the continuous case might not contain any integer point. In Fig. 2, z_2, not being an extreme point, would not be recognised as a dominating integer point when the continuous Pareto efficiency tests are applied.

2. Detection

A new algorithm has been developed and implemented in IGPSYS to detect and categorise the Pareto state (efficient, inefficient, unbounded) of the objectives for integer GP problems [13]. The state of every objective will be reported at the initial optimal integer GP solution. This classification helps the decision maker to find out the state of each objective in order to make rational decisions in the real life problems, especially in large scale integer GP models where the shear amount of data renders it impossible to visually scan for such details.

To be able to detect the Pareto efficiency status of an integer GP optimum solution, other integer points must be found in the feasible dominating area. In the case where there are no other integer points in the feasible dominating area, the optimum integer GP

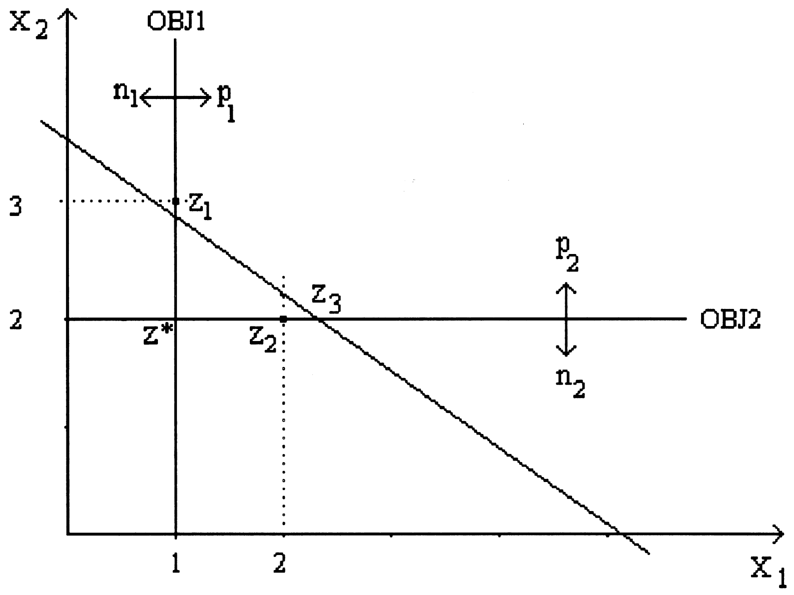

Fig. 2. Pareto efficiency in an integer case.

point is classified Pareto efficient. An integer optimisation technique, such as branch and bound with different state-of-the-art branching facilities and speed ups [13] is employed for the detection process. Unlike the continuous case, depending on the structure of the model, there are movements from the initial integer point to at least one, and in most cases, several other integer points in the dominating area. Every time that a new integer point is detected, the status of every objective is examined at that point. If there are objectives with status unknown, the original integer optimum point is retrieved and branch and bound is employed to maximise a new achievement function. This results in the movement towards and the reaching of a new integer point. The status of the remaining unknown objectives are then examined.

2.1. Notation

The following notation is used for the explanation of the detection and restoration algorithms described in this paper.

- *InfCnt* = The Pareto inefficient objective counter
- L_{NWV_i} = A lower bound placed on NWV_i
- NWV_i = The deviational variable of one-sided objective i which is not in the achievement function
- U_{WV_i} = The upper bound placed on WV_i
- $Value_{NWV_i}$ = The value of NWV_i at the initial optimal point
- $Value^*_{NWV_i}$ = The value of basic NWV_i at the new optimal point
- $Value_{WV_i}$ = The value of WV_i at the initial optimal point
- WV_i = The deviational variable of one-sided objective i which is in the achievement function

2.2. Integer GP Pareto efficiency detection procedure

The following algorithm outlines a new integer Pareto efficiency detection procedure developed and implemented in IGPSYS. It is used to detect the Pareto state of the optimum point by the maximisation of the non-weighted deviational variables, *NWV*s [13].

The steps of the algorithm are now stated:

(1) Initialise the state of each objective to 'Pareto Efficient' if both deviational variables are in the achievement function (two-sided objective), or otherwise to 'Unknown'. Note: The reason for the above classification is that if both deviational variables of an objective, say OBJ1, are in the achievement function then they are at their minimum value in the optimal solution. Any change will cause the objective to move from its optimal value. Thus, OBJ1 is classified as Pareto efficient.

(2) If a WV_j is in the basis, an upper bound is placed on it to stop possible degradation. The upper bound is the value of WV_j at the optimal solution. That is ($U_{WV_j} = Value_{WV_j}$). WV_js outside the basis are fixed to zero.

(3) Set up an achievement function of NWV_js from those objectives with status 'unknown'.

(4) If NWV_j is in the basis, a lower bound is placed on it to stop possible degradation. The lower bound is the value of NWV_j at the optimal solution point. That is ($L_{NWV_j} = Value_{NWV_j}$).

(5) Solve the integer GP by performing the Branch and Bound algorithm, maximising the new achievement function.

$$\text{Max } z = \sum_{j \subseteq J'} NWV_j \tag{12}$$

subject to,

$$f_i(\mathbf{x}) + n_i - p_i = b_i, \quad i = 1, \ldots, k \tag{13}$$

$$WV_j^b \leq Value_{WV_j} \tag{14}$$

$$NWV_j^b \geq Value_{NWV_j} \tag{15}$$

$$\mathbf{x} \in \mathbf{C}_s \tag{16}$$

$$\mathbf{n}, \mathbf{p} \geq 0 \tag{17}$$

$\mathbf{x} \geq 0$ and integer

where J' is the set of objectives with status 'unknown'. WV_j^b and NWV_j^b are the basic WV_j and NWV_j, respectively, with the optimum values of $Value_{WV_j}$ and $Value_{NWV_j}$. The integer GP optimum solution point is the starting point for this maximisation problem. The newly created integer GP maximisation problem is solved using IGPSYS. Different Branch and Bound strategies could be used, depending upon the structure of the problem [13].

(6) Perform the following procedure:

InfCnt = 0
For each objective, j = 1 to J, Do
If ($Value^*_{NWV_j} > L_{NWV_j}$) Then
(OBJ_{NWV_j}) is 'Pareto Inefficient'
InfCnt = InfCnt + 1
Endif
Continue (For loop)
If (InfCnt = 0) Then
'Unknown' objectives are 'Pareto Efficient' STOP
Else
Retrieve original optimum integer GP, GOTO 2
Endif

In step 6, the values of the deviational variables in the new objective function are monitored. If $Value^*_{NWV_j}$ is greater than L_{NWV_j} then there has been improvement. Therefore, the objective whose deviational variable is NWV_j is classed as 'Pareto inefficient'. If the Pareto inefficiency counter has not been incremented, i.e., no improvement has occurred ($InfCnt$ = 0), then objectives with unknown Pareto efficiency state are classified as 'Pareto efficient'. In the case where the Pareto inefficiency counter is incremented ($InfCnt \geq 1$), the original integer GP problem is retrieved and a new achievement function, from the deviational variables of the objectives which still maintain status 'unknown', is formed and the resultant new integer GP problem is maximised.

This is a filtering process; at each iteration undetermined objectives are examined and the process terminates when efficiency or inefficiency of all objectives are determined. This guarantees the classification of each objective and guarantees the convergency of the method.

2.3. An illustrative example

The filtering process of integer GP detection technique is illustrated by a hypothetical example on a simple integer GP problem with four objectives where, n_i and p_i represent the negative and positive deviational variables for objective i, respectively. Let the original achievement function be of the form,

$$\text{Min } z_0 = n_1 + p_2 + n_3 + p_4 \tag{19}$$

After solving the integer GP problem, in order to detect the Pareto efficiency status of the solution, a new achievement function is constructed from the non-weighted deviational variables, that is,

$$\text{Max } z_1 = p_1 + n_2 + p_3 + n_4 \tag{20}$$

The new solution is then compared with the initial optimal solution and the objectives are then given the relevant Pareto state. Table 1 shows the hypothetical value of the deviational variables during the process of integer Pareto detection, where

$$\begin{cases} 0<\alpha_1 \\ 0<\beta_1<\beta_2 \\ 0<\gamma_1<\gamma_2 \end{cases}$$

Thus, p_1 and p_3 have improved from their original values at z_0 to α_1 and γ_2, respectively, at z_1, whereas values of n_2 and n_4 remain unchanged. Thus objectives 1 and 3 are classified as Pareto inefficient. Table 2 shows the Pareto status of all objectives during the Pareto detection process.

Pareto efficiency status of all objectives are unknown at z_0 whereas objectives 1 and 3 are categorised as Pareto inefficient at z_1. Subsequent categorisation of objectives with as yet undefined status takes place by maximisation of z_2 and z_3, respectively. That is

$$\text{Max } z_2 = n_2 + n_4 \tag{21}$$

and

Table 1
Process of integer GP detection algorithm

OBJ	z_0	z_1	z_2	z_3
1	$p_1=0$	$p_1=\alpha_1$	–	–
2	$n_2=\beta_1$	$n_2=\beta_1$	$n_2=\beta_2$	–
3	$p_3=\gamma_1$	$p_3=\gamma_2$	–	–
4	$n_4=0$	$n_4=0$	$n_4=0$	$n_4=0$

$$\text{Max } z_3 = n_4 \tag{22}$$

It is clear that objectives 1 and 3 have both been improved as a result of the integer point found when maximising z_1, thus, classed as Pareto inefficient. In maximising z_2, objective 2 is improved at an integer point and is therefore categorised as Pareto inefficient. Finally, objective 4 is Pareto efficient since no improved integer points could be found when maximising z_3.

3. Restoration

If one or more objectives in an integer GP model are found to be Pareto inefficient, the decision maker would be advised to investigate and preferably restore efficiency. Three different restoration algorithms; straight, preference based and interactive, have been developed and implemented in IGPSYS. These algorithms help the decision maker to restore Pareto efficiency within the subset of the feasible region that dominates the inefficient point found. In the case where there exist multiple Pareto efficient points on the efficient boundary that dominate the original solution, the different mechanisms and structure of the three restoration techniques developed in this research provides added flexibility for the decision maker [13].

Techniques in the continuous case for restoring Pareto efficient points in the dominating feasible area cannot be applied to the integer case. The algorithms developed for the continuous case can only restore non-integer points. These points are on the continuous Pareto efficient boundary. Whereas, the integer Pareto efficient points are normally in the interior of the dominating area and are located on a different and *integer* Pareto efficient boundary.

The original integer optimisation causes movement from the initial continuous optimum point to the integer optimum point. The Pareto efficiency status of objectives at this point are then investigated. In the case where inefficient objectives are detected, restoration techniques are employed to achieve efficiency. The Pareto efficient solution for an integer GP model is obtained by using one of the variants of the Pareto restoration analysis techniques. The restoration of Pareto efficient integer points in the dominating feasible area are performed using any of the algorithms available in the integer optimisation system developed in Ref. [13].

In most problems, there are usually more than one Pareto efficient integer solution. Movement towards and reaching different integer Pareto efficient solutions in the dominating feasible area may depend on the decision maker's use of a particular restoration technique variant.

3.1. Straight restoration

This is similar to the detection process. Maximisation of the sum of the non-weighted deviational variables (NWV_j) is performed i.e.,

$$\text{Max } z = \sum_{j \subseteq J''} NWV_j \tag{23}$$

where J'' is the set of NWV_j's of the objectives found to be inefficient by the detection process. This is equivalent to the imposition of an additional priority level to the integer GP. In the case of detecting one or more Pareto unbounded objectives, a Pareto efficient solution cannot be reached. The steps of this algorithm can be stated as follows:

(1) Call Detection

(2) If all objectives are efficient, Stop. Otherwise continue.

(3) Place an upper bound on WV_js ($U_{WV_j} = Value_{WV_j}$). This prevents degradation of WV_js during the maximisation process.

(4) Set up an achievement function of NWV_js from those objectives with status 'Inefficient'.

(5) Place a lower bound on NWV_js ($L_{NWV_j} = Value_{NWV_j}$). This prevents degradation of NWV_js during the maximisation process.

(6) Solve the integer GP, maximising the new achievement function. This process is effectively the same as that of a lexicographic integer GP where the last priority is made up of a summation of the NWV_js.

There is usually more than one Pareto efficient integer point in the dominating area. This technique restores the integer point with the best unweighted improvement, i.e., that which offers the greatest improvement considering all objectives equally.

3.2. Preference based restoration

This is achieved by the addition of one or more extra priority levels to the original integer GP. In this case, the weights of the deviational variables in the original achievement function and their priority levels for the lexicographic GP case, are taken into account, maintaining a preference structure. For the weighted

Table 2
Table of status classification

OBJ	z_0 status	z_1 status	z_2 status	z_3
1	Unknown	Inefficient	Inefficient	Inefficient
2	Unknown	Unknown	Inefficient	Inefficient
3	Unknown	Inefficient	Inefficient	Inefficient
4	Unknown	Unknown	Unknown	Efficient

GP the new achievement function will be of the form

$$\text{Lex Min } a = \left[\sum_{j=1}^{J} (w_j \cdot WV_j), \sum_{j=1}^{J} - (w_j \cdot NWV_j) \right] \quad (24)$$

For the lexicographic GP, the mathematical representation of the new achievement function is of the form;

$$\text{Lex Min } a = \left[\left(\sum_{j=1}^{J} w_{1j} WV_j \right), \left(\sum_{j=1}^{J} w_{2j} WV_j \right), \ldots, \left(\sum_{j=1}^{J} - w_{1j} NWV_j \right), \left(\sum_{j=1}^{J} - w_{2j} NWV_j \right), \ldots \right] \quad (25)$$

where w_{lj} is the weight of WV_j in the objective function associated with NWV_j and priority level l (l = 1, ..., L). This model has $2L$ priority levels.

This method has an advantage over straight restoration in that it considers the relative degree of importance of each objective at each priority level to the decision maker. In this technique, priority levels are dealt with in their order of importance when considering lexicographic integer GPs. The following is the algorithm for the preference based technique designed to handle integer GP models:

1. Call Detection
2. If all objectives are efficient, Stop. Otherwise continue.
3. Place upper bounds on WV_js
($U_{WV_j} = Value_{WV_j}$)
4. For i = 1 to Number-of-Priority-Levels. Set up an achievement function
of NWV_js from those objectives with status 'Inefficient' in priority
level i with the corresponding weights of w_{ij}. Solve the integer GP,
maximising the new achievement function.
If ($i>1$, and NOT-last-priority-level) Then
Place lower bounds on NWV_js
($L_{NWV_j} = Value_{NWV_j}$)
Else
Exit.
Endif
Next i.

3.3. Interactive restoration

The first step of the restoration phase consists of providing the decision maker with the set of inefficient objectives. The decision maker then chooses the objective that he/she wishes to improve. Each time a single achievement function containing only one NWV_j is maximised:

$$\text{Max } z = NWV_j, \quad j \in J \quad (26)$$

where J is the set of objectives found to be inefficient by the detection process. Once the achievement function is set up, the new problem is then optimised by performing the integer optimisation using a branch and bound algorithm. A new report is then constructed stating the efficiency or inefficiency of each of the set of objectives at the new integer point. This interactive process is repeated until all objectives are returned efficient. The maximisation and thus improvement of a singleton objective during each iteration of the algorithm guarantees the convergency of the method.

Depending on decision maker's choice of objectives to be improved during this interactive process, optimisation causes movement towards and ultimately reaching the best integer solution for his/her purpose.

If any one of the objectives is however found to be Pareto unbounded, it cannot be made Pareto efficient and is thus categorised as Pareto unbounded. The interactive restoration approach is implemented using the following steps:

(1) Call Detection. If every objective is 'Efficient', Stop. Otherwise continue.

(2) Set up a new achievement function of the Kth objective as selected by the decision maker with inefficient status, NWV_k

(3) Solve the integer GP, maximising the new achievement function by performing integer optimisation

(4) Place a lower bound on NWV_k. ($L_{NWV_k} = Value_{NWV_k}$).

(5) Goto 1.

4. Computational experiments

In order to evaluate the computational performance of the detection and restoration analysis tools, a set of integer GP problems from different industrial contexts are analysed [12]. A 486 PC with 33 MHz Intel co-processor was used for this purpose. All times given are in CPU seconds. A summary of the analysis is given in Table 3.

Models 2 and 7 are taken from MIPLIB, library of real life integer programming problems, and converted to integer goal programming models. The most time

Table 3
Performance of integer Pareto detection and restoration analysis tools on some industrial contexts

Model	Application	Type	NOB	DTM	NIO	SRT	PRT
1	Set partitioning	WGP	6	0.441	4	0.055	0.059
2	MIPLIB	WGP	12	0.715	7	0.109	0.109
3	Oil distribution	WGP	20	2.914	6	1.703	1.703
4	Diet planning	WGP	20	2.582	19	0.330	1.648
5	Capital budget	LEX (8)	24	0.386	4	0.054	0.054
6	Transportation	LEX (4)	39	0.277	0	0.0	0.0
7	MIPLIB	WGP	59	2.527	39	0.328	0.328
8	Education	LEX (4)	86	0.604	12	0.221	0.109
9	Basket selection	LEX (3)	181	17.967	3	9.230	9.121
10	Petro-Chem-Plants	LEX (4)	319	253.574	123	40.219	6.922

The explanation of the column headers are as follows: Type = The type of the GP model under consideration: WGP = Weighted GP; LEX(n) = Lexicographic GP with n priority levels; NOB = Number of objectives; DTM = Detection time; NIO = Number of Inefficient objectives in the initial optimal solution; SRT = Straight restoration time; PRT = Preference based restoration time.

consuming model regarding the detection time is model 10 with the highest number of inefficient objectives. Model 6 has no inefficient objectives and therefore the detection time is minimum. The straight and preference based restoration times are the same or with little difference in most models apart from models 4 and 10 with 19 and 123 inefficient objectives, respectively. The detection and restoration analysis processing times, except for model 6, are approximately less than 10% of the execution time for finding the optimal solution.

5. An example

In this section, an example is given to demonstrate the Pareto detection and restoration algorithms developed to overcome GP inefficiency with respect to a model's integer requirements. The mathematical representation of this example has the form

$$\text{Min } z = 2n_1 + n_2 + n_3 \tag{27}$$

Subject to,

$$x_1 + n_1 - p_1 = 6.5 \tag{28}$$

$$x_2 + n_2 - p_2 = 7.5 \tag{29}$$

$$2x_1 + 3x_2 + n_3 - p_3 = 7.5 \tag{30}$$

$$x_1 \leq 10.5 \tag{31}$$

$$0.6x_1 + x_2 \leq 20.5 \tag{32}$$

$$\mathbf{n}, \mathbf{p} \geq 0 \tag{33}$$

$$\mathbf{x} \geq 0, \text{ and integer} \tag{34}$$

Fig. 3 illustrates the above example diagrammatically. The shaded area OABC represents the feasible region for the model with three objectives and two hard constraints. Point F(6.5, 7.5) is the initial GP optimum solution. DBEF is the feasible dominating area of this solution. It contains substantial number of integer points, few of which are marked in Fig. 3. By applying branch and bound algorithm, the initial integer GP optimum solution, point G(7, 8), is obtained.

Applying the detection algorithm, the new achievement function will have the form;

$$\text{Max } z' = p_1 + p_2 + p_3 \tag{35}$$

Maximisation of z' for the remaining of this section is subject to the set of goals and constraints as set out in Eqs. (28)–(34). This will result in the detection of point H(10, 14) in Fig. 3 which means improvement of objectives 1, 2 and 3, i.e., all objectives are classified as Pareto inefficient.

A Pareto detection status report is given to the decision maker by IGPSYS once the status of each objective is classified, as shown in Table 4.

- The straight restoration algorithm in the integer case is similar to the detection process. Maximisation of deviational variables, as detailed in Eq. (35), is performed. This results in the movement towards a new position on the Pareto efficient boundary and finally settling on B(10, 14.5). By performing the integer optimisation, the integer Pareto efficient point, H(10, 14), is obtained where all objectives are improved.
- The preference based restoration method considers the decision maker's preferences. The weights in the original integer GP are used when the maximisation of

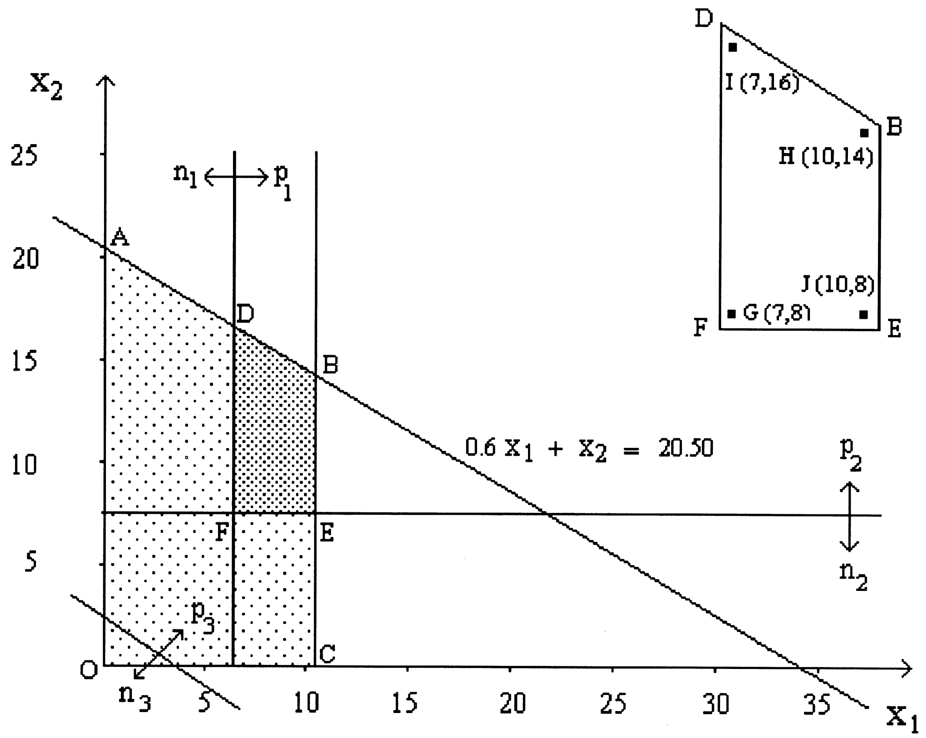

Fig. 3. Detection and restoration of integer pareto efficiency.

NWV_js take place. For the example above the maximisation problem will have the form

$$\text{Max } z' = 2p_1 + p_2 + p_3 \tag{36}$$

The weight on objective 1 causes movement towards and finally reaching the integer Pareto efficient point H(10, 14). If objective 2 had a greater weight, e.g., the maximisation problem was of the form

$$\text{Max } z' = p_1 + 2p_2 + p_3 \tag{37}$$

then the Pareto efficient point I(7, 16) would have been obtained. Therefore, in the preference based restoration technique the original structure of the integer GP is maintained, satisfying the decision maker's preferences.

- The final variant of the restoration technique is the interactive case where the decision maker has the option of choosing objectives to be improved during the restoration process. At each stage the Pareto efficiency status of every objective is reported to the decision maker.

Choosing the interactive restoration technique for the above example, the decision maker has a choice of improving any of the 3 Pareto inefficient objectives. If the decision maker chooses objective 2 to be improved then, point I is reached and objectives are reported Pareto efficient. If objective 3 is chosen to be improved, the simplex and branch and bound algorithms will cause movement towards the Pareto efficient integer boundary and finally residing at point H, where all objectives are Pareto efficient. Point J(10, 8) is obtained when objective 1 is chosen to be improved first. The status of objectives are then reported to the decision maker, as shown in Table 5, where objectives 2 and 3 are inefficient. At this point the decision maker

Table 4
Pareto detection status report

Integer Pareto detection report
Objective 1: Inefficient
Objective 2: Inefficient
Objective 3: Inefficient

Table 5
Pareto restoration status report

Integer Pareto restoration report
Integer point: ($X_1 = 10$, $X_2 = 8$)
Objective 1: Efficient
Objective 2: Inefficient
Objective 3: Inefficient

can either choose objective 2 or objective 3 to be improved. In either case, H(10, 14) is the final Pareto efficient integer point.

6. Conclusion and discussions

This paper has reported the development of a novel approach to the design and implementation of unique specialised integer Pareto efficiency detection and restoration analysis tools to detect and restore Pareto efficiency [13].

The algorithm developed in this research enables the practitioners to model and solve real life integer GP problems and find the corresponding Pareto efficient solutions by using the integer GP analysis tools developed in Ref. [13]. This can easily be implemented in other integer optimisers. This research has thus overcome the criticism and doubt [24] that GP/Integer GP faces in the possible production of inferior solutions.

A further research would enable the new Pareto efficiency analysis tools to be adopted with new heuristic search methods such as Tabu search and Genetic algorithms [3] in order to find Pareto efficient solutions.

References

[1] Charnes A, Cooper WW, Ferguson R. Optimal estimation of executive compensation by linear programming. Management Science 1955;1:138–51.

[2] Charnes A, Cooper WW. Management models and industrial applications of linear programming. New York: Wiley, 1961.

[3] Glover F, Kelly JP, Laguna M. Genetic algorithm and Tabu search. Computers and Operations Research 1995;22(1):111–34.

[4] Hannan EL. Non-dominance in goal programming, INFOR. INFOR, Canadian Journal of Operational Research and Information Processing 1980;18:300–9.

[5] Hannan EL. An assessment of some of the criticisms of goal programming. Computers and Operations Research 1985;12:525–41.

[6] Ijiri Y. Management goals and accounting for control. Amsterdam: North Holland, 1965.

[7] Ignizio JP. Goal programming and extensions. Lexington, Mass, Heath (Lexington Books), 1976.

[8] Ignizio JP. Linear programming in single and multiple objective systems. Englewood Cliffs, NJ: Prentice-Hall Inc., 1982.

[9] Ignizio JP. An algorithm for solving the linear goal programming problem by solving its dual. Journal of the Operational Research Society 1985;36:507–15.

[10] Ignizio JP, Cavalier T. Linear programming. Prentice Hall, 1994.

[11] Lee SM. Goal programming for decision analysis. Auerback, Philadelphia, 1972.

[12] Mathematical Programming Group. A library of real life goal programming and integer goal programming models. Department of Computer Science and Mathematics, University of Portsmouth, UK, 1997.

[13] Mirrazavi SK. Investigation and development of efficient integer and integer goal programming systems. Ph.D. Thesis, University of Portsmouth, UK, 1997.

[14] Pareto V. Course d'economie politique. Rouge, Lausannes, 1896.

[15] Romero C. Handbook of critical issues in goal programming. Pergamon Press, 1991.

[16] Tamiz M, Jones DF. Expanding the flexibility of goal programming via preference modelling techniques. Omega, International Journal of Management Science 1995;23(1):41–8.

[17] Tamiz M, Jones DF. Algorithmic extensions to the method of Martel and Aouni. Journal of the Operational Research Society 1995;46:254–7.

[18] Tamiz M, Jones DF, El-Darzi E. A review of goal programming and its applications. Annals of Operations Research 1995;58:39–53.

[19] Tamiz M, Jones DF. Goal programming and Pareto efficiency. Journal of Information and Optimisation Sciences 1996;17(2):1–17.

[20] Tamiz M, Jones DF, Mirrazavi SK. Intelligent solution and analysis of goal programmes: the GPSYS system. Technical Report. School of Computer Science and Mathematics, University of Portsmouth, UK, 1997.

[21] Tamiz M, Jones DF. An example of good modelling practice in goal programming: means for overcoming incommensurability. In: Cabellero R, Ruiz F and Steuer R, editors. Advances in multiple objective and goal programming. Lecture Notes in Economics and Mathematical Systems. *Springer*, Vol. 455, pp. 29–37, 1997.

[22] Tamiz M, Jones DF. Interactive frameworks for investigation of goal programming models: theory and practice. Journal of Multi-Criteria Decision Analysis 1997;6:52–60.

[23] Zeleny M, Cochrane JL. A prior and a posteriori goals in macroeconomic policy making. In: Cochrane JL and Zeleny M, editors. Multiple criteria decision making. University of South Carolina Press, Columbia, 1973. pp. 373–391.

[24] Zeleny M. The pros and cons of goal programming. Computers and Operations Research 1982;8:357–9.