Operations Research Applied to Forestry Management

By

Maikel Sianturi

A thesis presented to the University of Manitoba in partial fulfillment of the requirements for the degree of

Master of Science

in Applied Mathematics

Winnipeg, Manitoba, Canada

September 2000

0-612-53253-4

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

Operations Research Applied to Forestry Management

BY

Maikel Sianturi

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of Master of Science

Abstract

People want to use forests for their benefits as much as possible but environmental impacts of their actions should be minimized. This leads to difficult land management problems with multiple, conflicting objectives. Forest land management analysts have developed and utilized sophisticated planning methods to address complex issues involving multiple objectives. An intensive literature review of these techniques is presented. The most popular multiobjective technique among forester is Goal Programming. Multiobjective Genetic Algorithms are relatively new optimization techniques which have not yet been used in forestry. Two multiobjective forestry problems are solved using a Multiobjective Genetic Algorithm and the results are compared to Goal Programming solutions. It is shown that the Multiobjective Genetic Algorithm can find solutions with better tradeoffs between conflicting objectives.

Acknowledgements

I am blessed that I have the opportunity to thank some people at this time. I thank my supervisor Dr. Peter Aitchison for guiding me during the research and keeping my motivation alive during my difficult time due to the religions strife in my hometown, Ambon, Indonesia.

As well, I thank the Canadian International Development Agency (CIDA) project for all the financial support and the CIDA staff for giving me great encouragement to complete my study. I also thank my fellow students, especially Tim Rogalsky and Allan Hedilberg, for the help during my study and the Department of Mathematics.

I dedicate this thesis to people of my hometown and my youngest sister, Mawan Sianturi, who suffer from religious persecutions. This thesis is also dedicated to my mother who always inspires me to pursue higher education.

Table of Contents

Chapter 1

Introduction

Consideration of alternative uses of the forest and its products nearly always raises the question, "What is the best way?" Because of the number of alternatives, the complexity of the product interactions, and the conflicting desires of the public, an optimal answer may be impossible to find. Some help has been provided for forest managers by the decision tool known as Operations Research (OR).

Over the past few decades several factors have altered the practice of forest land management. As population and resource development increase, many forest-based outputs are approaching or exceeding sustainable levels of use. People are increasingly aware of the need to preserve the forest ecosystem, to sustain threatened and endangered species, wildlife habitat, scenic beauty, and biodiversity. As a result, forest land manager – especially on public lands - are shifting their emphasis from the production of goods and services towards the maintenance of forest health, biodiversity, and productivity. On private timberlands this trend is tempered by the concurrent need to remain competitive in the global market place.

Both classes of ownership have created new challenges for the OR community. On public land, the shift toward an ecology system model has necessitated the development of a new set of OR models that incorporate spatial relationship, ecological relationships, resource protection issues, and consideration of a wide spectrum of natural resources beyond timber. In the private sector, the increase in open, global markets has encouraged forest products companies to improve productivity and managerial efficiency while being cognizant of environmental and ecological values.

In the late 1950's the most common OR method in forestry was Linear Programming (LP), in which objectives and constraints are expressed in the form of linear equations. A few applications were suggested in the 1950's and early 1960's. Many of these early suggestions were for problems such as optimizing harvest schedules, production mixes, and product distribution. In the 1970's, applications expanded rapidly, and models based on variations of LP were developed. The U.S Forest Service uses LP in both timber management and land-use planning.

LP models work quite well in allocating resources for market-value goods. However, they may involve unrealistic assumptions when applied to the kind of complicated decision making situations common in multiple-use forest management. The manager is charged with obtaining a desired mix of goods and services from limited resources, and will usually have several alternative courses of action open to him. To choose among them, he must know both the tradeoffs between one course of action and another, and the relative desirability of the goods and services. Goal Programming (GP) provides a way of allocating resources efficiently in decision making situations that involve multiple goals.

To use GP to solve multiobjective problems, a target level of each goal is required. Decision makers usually specify these target levels. However, the target levels are sometimes not optimal in the sense that all or some of the target levels are still upgradable without sacrificing any goal. Even if target levels are optimal, the solutions generated by GP are sometimes extreme; that is, one or more goals are achieved but the other(s) might be very far from the target level specified. This implies that good tradeoff solutions are not captured by GP. However, in real problems good tradeoff solutions are usually preferable.

A relatively new technique, effective at finding tradeoff solutions, is the Multiobjective Genetic Algorithm (MOGA). Genetic Algorithms (GAs) are useful search methods loosely based on ideas from population genetics. The output from a MOGA contains many solutions, each of which represents a good tradeoff between the (possibly) conflicting goals. In this thesis, it is shown that MOGAs are potentially very useful in solving forestry problems.

In chapter 2, several OR techniques commonly used in forestry management are discussed. First Linear Programming, along with its variations, is examined. However, this can only optimize one objective, so multiobjective optimization techniques are needed to solve multiobjective problems which are common in forestry. The most common technique used in forestry management is Goal Programming. Another technique, used to capture uncertainties in forestry problems is called fuzzy optimization. The shortest path algorithm (Dijkstra's algorithm) is described as a tool to determine the shortest route in a road network in a forest. Finally, Dynamic Programming is described, a technique usually used to accommodate forest products such as tree bucking.

In chapter 3 a literature review is presented discussing the OR techniques that have been proposed to solve forestry problems. Four broad areas of forestry are discussed: resource allocation, spatial concerns, road construction, and forest products. Proper allocation of resources, such as labour, equipment, and land management, can save a lot of money and time. A spatial concern arises when considering the harvest of adjacent land parcels in the same time period. This spatial concern is imposed in forestry models in order to ensure ecosystem stability. In the construction of a road network in a forest, economic and environmental impacts must be considered. One of the main issues when considering forest products is to obtain maximal profit from selling the tree. In this regard, cutting patterns must be carefully designed in advance.

In chapter 4, Genetic Algorithms are discussed. Genetic Algorithms are relatively new optimization techniques based on a model of biological evolution. When using a cycle of evaluation, selection, and genetic changes, iterated for many generations, the overall fitness of the population generally improves. The individuals in the population represent improved solutions to whatever problem was posed. In GAs, the individuals converge to a single optimal value (single objective). Multiobjective Genetic Algorithms are an extension of Genetic Algorithms designed such that the individuals in the population converge to optimal solutions representing tradeoffs among many objectives.

In chapter 5, two conflicting multiobjective forestry problems are solved by Goal Programming and Multiobjective Genetic Algorithm, and the results are compared. In the first problem the decision makers specify a target level for each goal and in the second problem decision makers do not specify either the target levels or the priorities of the goals. Some MOGA solutions are comparable to GP solution, but other MOGA solutions represent better tradeoffs among the goals. Thus MOGA has excellent potential for forestry management.

Chapter 2

Operations Research Methods Commonly Used in Forestry

Operations Research (OR) is the professional discipline that deals with the application of information technology to informed decision making. OR professionals aim to provide rational bases for decision-making by seeking to understand and structure complex situations. This understanding is used to predict system behavior and to predict decisions which give improved system performance. Much of this work is done using analytical and numerical techniques to develop and manipulate mathematical algorithms which model the organizational systems composed of people. machines. and procedures.

The main methods of OR are linear programming, integer and mixed programming, dynamic programming, fuzzy programming, stochastic programming, goal programming, etc. Using OR methods can save a lot of goods. money and time as OR methods can provide us with information which can help make efficient decisions. OR methods have also been used to solve forestry problems as forestry problems become more complex.

Forests provide many natural resources that benefit people. give shelter for animals and contain much of the world's biodiversity. This last topic being very important to the integrity of the earth's biosphere.

Back in 1849, Faustman (1849) first proposed a conceptually correct analysis of optimal rotation length for "even-aged" timber stands. He treated timber as a maturing asset and located the optimum rotation-age for identifying harvesting (a fixed amount each year.) Andersen (1976) used optimal control theory to study the problem and derived a model identical to Faustman's. Amidon et al. (1968) found an optimal solution for the joint stocking-rotation decision for an even-aged stand using dynamic programming. Grevatt at al. (1967) developed two linear programming models to aid in nursery planning.

In the sections below, OR methods commonly used in forestry problems are described and cited so that the reader can form a general idea about main OR methods that have been intensively applied to the problems. We will encounter these methods frequently in the remaining chapters.

2.1. Linear Programming

Increasing complexity in the forestry industry characterizes the evolution of the planning problems perceived by agency analysts, planners and managers. They are pressured by society to take into concerns about threatened species and endangered species, wilderness and old growth preservation, water quality and road construction. Consequently, forest planners have to follow a systematic planning procedure proposed by Cortner et al. (1983), which

1. Defines objectives or values to be optimized;
2. Identifies the full range of possible alternatives for achieving the desired objectives;
3. Comprehensively evaluates the physical, environmental, social, and economic consequences of each alternatives; and
4. Chooses the course of action which best realizes the stated objectives.

To best satisfy all the issues, linear programming (LP) has been utilized. Navon (1971) used LP to develop a timber resource allocation model (timber RAM) that saw widespread use throughout the forest community. It was designed primarily to address timber production, but the many other forest values such as recreation and wildlife were addressed by way of constraints on harvest and regeneration activities.

LP is an optimization method applicable for the solution of problems in which the objective function and the constraints appear as linear functions of decision variables. The constraint equations in a LP problem may be in the form of equalities or inequalities.

The general LP problem can be stated in the following standard form (Grossman, 1991)

$$Minimize \quad f(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j$$

subject to constraints

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1, 2, \ldots, m$$

$$x_j \geq 0, \quad j = 1, 2, \ldots, n$$

where c_j, b_j, and a_{ij} $(i = 1, 2, \ldots, m;\ \mathrm{j} = 1, 2, \ldots, n)$ are known constants, and x_j are decision variables.

Any LP problem can be put in the standard form by the use of the following transformations.

1. The maximization of a function $f(x_1, x_2, ..., x_n)$ is equivalent to the minimization of the negative of the same function.

2. In most real optimization problems, decision variables, x_j, have to be nonnegative. However, a variable may be unrestricted in sign in some problems. In such a case, an unrestricted variable (which can take a positive, negative or zero value) can be written as the difference of two nonnegative variables. Thus, if x_j is unrestricted in sign, it can be written as $x_j = x'_j - x''_j$ where $x'_j \geq 0$ and $x''_j \geq 0$.

3. If a constraint appears in the form of a "less than" type of inequality it can be converted into the equality by adding a nonnegative slack variable. Similarly, if the constraint is in the form of a "greater than" type of inequality it can be converted into the equality form by subtracting a variable known as surplus variable.

The standard simplex solution method is available in many computer software packages such as LINDO and LINGGO (LINDO Systems Inc.) which can be used to solve this LP problem.

Example 2.1;

LP is used to solve a simple reforestation planning and budgeting problem. Assume that the activity alternatives available to the forest manager include the following four land classes:

Class 1: Site II type B bare land with a north aspect in seed zone 53
Class 2: Site III type B bare land with a south aspect in seed zone 52

Class 3: Site IV type B bare land with a north aspect in seed zone 51

Class 4: Site II substocked land with a south aspect in seed zone 52

Assume further that there are 100 acres in each land class and that any portion of the total may be scheduled for treatment. Resources available are: budget-$8000; seedlings-30,000, 55,000 and 35,000 for zones 51, 52, and 53, respectively; and seeds – 25lb for each zone.

Assume that 0.75 lb of seed or 600 seedlings are required to treat 1 acre of bare land, and in the case of interplanting, 500 seedlings per acre. If necessary, seed and seedlings may be transferred between adjacent zones. Thus seedling supply in zone 51 is 85,000 trees; in zone 52, 120,000 trees; and in zone 53, 90,000 trees.

The alternative activities, capital requirements, and activity values are shown in the following table. There are two alternative activities – seeding or planting – for bare land situations, while interplanting is the only possible treatment for substocked plantations. In the paragraph above there are in total 30,000+55,000 + 35,000 = 120,000 seedlings available. Zone 51 and 52 are adjacent and hence zone 51 could have 30,000 + 55,000 or 85,000 seedlings if zone 52 used none.

Decision Variable	Activity	Resources requirements			Present net
		Capital ($)	Seedlings	Seeds	Worth ($)
x_1	Acres planted in land class 1, seed zone 53	31.50	600	0	163.20
x_2	Acres seeded in land Class 1, seed zone 53	10.80	0	0.75	104.10
x_3	Acres planted in land class 2 seed zone 52	31.50	600	0	58.90
x_4	Acres seeded in land class 2 seed zone 52	10.80	0	0.75	19.30
x_5	Acres planted in land class 2 seed zone 51	31.50	600	0	6.30
x_6	Acres seeded in land class 2 seed zone 51	10.80	0	0.75	-1.35
x_7	Acres interplanted in land class 4, seed zone 52	24.00	500	0	73.80

The objective function of this problem is to maximize present net worth of acres planted, that is,

Maximize

$$(163.20x_1 + 104.10x_2 + 58.90x_3 + 19.30x_4 + 6.30x_5 - 1.35x_6 + 73.80x_7)$$

Constraints are

1. Budget constraint

$$31.50x_1+10.80x_2+31.50x_3+10.80x_4+31.50x_5+10.80x_6+24.00x_7 \le 8{,}000$$

2. Seedling constraints

$$600x_1 \le 90{,}000$$

$$600x_5 \le 85{,}000$$

$$600x_1 + 600x_3 + 600x_5 + 500x_7 \le 120{,}000$$

3. Seed constraints

$$0.75x_2 \le 50$$

$$0.75x_4 \le 75$$

$$0.75x_6 \le 50$$

$$0.75x_2 + 0.75x_4 + 0.75x_6 \le 75$$

4. Area constraints

$$x_1 + x_2 \le 100 \text{ (land class 1)}$$

$$x_3 + x_4 \le 100 \text{ (land class 2)}$$

$$x_5 + x_6 \le 100 \text{ (land class 3)}$$

$$x_7 \le 100 \text{ (land class 4)}$$

5. Nonnegative solutions, $x_i \ge 0$ for $i = 1, \ldots, 7$.

Solving the above problem using LINGO gives results as follows. Objective value = \$26290.00, $x_1 = 100.00$, $x_2 = 0$, $x_3 = 16.67$, $x_4 = 83.33$, $x_5 = 0$, $x_6 = 0$, $x_7 = 100.00$.

2.2. Integer Linear and Mixed Integer Linear Programming

Forestry problems have become more complex due to considerations of environmental impacts, recreational and other needs from the forests. Spatial consideration is one of the central issues that foresters have to take account in their decisions. This means that harvesting is restricted to a certain area. Therefore, forests should be blocked into contiguous areas so that, for example, adjacent blocks cannot be harvested in the same period of time. To accommodate this concern, a model can be creating using integer 0-1 programming, with 1 indicating harvesting and 0 indicating not harvesting. Similarly in resource allocation problems, integer variables can be used to indicate decisions such as the amount of equipment to buy or the amount which must be operated in order to achieve the required goals.

Integer linear programming is similar to linear programming except that all variables can only take integer values. Usually to solve this integer linear programming the equivalent linear programming is solved first and then the integer constraints are introduced using certain methods such as the commonly used branch and bound method (Winston, 1987). Mixed Integer Linear Programming is similar in that some of the variables are integer variables, but the remaining variables are continuous variables.

2. 3. Goal Programming

Multiple-use forest resource problems involve a consideration of multiple conflicting goals and objectives such as: increased net revenue from timber resources, improved water quality, protection of wildlife, preservation of natural beauty, and increased recreational opportunities. Managing multiple-use resources requires more complicated decision making. Managers are charged with obtaining a desired mix of goods and services using limited resources, and will usually have several alternative courses of action open to them. To choose among them, they must know both the tradeoff between one course of action and another, and the relative desirability of the goods and services. For example, if the decision makers want to provide 20 % more recreation in the forests, what quantity of timber products (if any) must be relinquished, and is there enough money and land to provide both the desired recreation and timber products ? (Schuler, et al. 1975).

Finding the best solution to multiple-use forest resources is very hard because some problems (goals) are complementary. For instance, Some timber harvesting helps wildlife by improving habitat, providing good forage, but full utilization of forage reduces timber yield. Since it is not possible to meet all objectives (goals), a good balancing (tradeoff) solution among the goals is preferable. The best tradeoff solutions are often considered to lie on the non-dominated (Pareto-optimal) set.

Pareto Optimal

In multi criterion or multiobjective problems, there is typically no solution that is "better than" all others, but rather tradeoffs must be made between the various objective functions.

Suppose, without loss of generality, that the objective functions form the vector function

$$f = (f_1, f_2, \dots, f_n)$$

with

$$f_i : S \to \mathbf{R}$$

for each component f_i, and assume further that each function is to be minimized. A solution $x \in S$ is now said to dominate another solution $y \in S$ if it is no worse with respect to any component than y and is better with respect to at least one. Formally

$$x \text{ dominate } y \quad \Leftrightarrow \quad \forall i \in \{1,2,\ldots,n\}: f_i(x) \leq f_i(y)$$
$$\text{and} \quad \exists j \in \{1,2,\ldots,n\}: f_j(x) < f_j(y)$$

A solution is said to be *Pareto Optimal* in S if it is not dominated by any other solution in S, and the *Pareto optimal* set or *Pareto optimal front* is the set of such nondominated solutions, defined formally as

$$S^{*} = \{x \in S \setminus \text{no } y \in S : y \text{ donimates } x\}.$$

Multiobjective problems are usually formulated as covering problems, with the goal being to find either the entire Pareto optimal set, or a number of different points near it.

Goal programming (GP) provides a way of allocating resources efficiently in decision making situations that involve mutiobjectives. Field (1973) is the first researcher introducing goal programming to solve multiobjective forestry problems.

A GP decision situation is generally characterized by multiple objectives. Some of these objectives may be complementary, while others may be conflicting in nature. GP allows the decision maker to specify a target for each objective. A solution of the complete problem minimizes the total deviations from the prescribed set of target values. The method for minimizing this deviation is called the method of distance function (Srinivas, N. et al. 1994). Our usage of the term goal function is synonymous with that of the objective function.

The team of Charnes and Cooper (1961) is generally credited with introducing the method to industrial problems. It may be noted that the initial purpose of developing the method was not multiple-objective decision making, but its subsequent use justifies the credit generally given to Charnes and Cooper for pioneering in the field. Lee (1972) has applied GP to problems in production planning, financial decisions, academic planning, and medical care, to mention a few. More recently, Kendall and Lee (1980) have applied the technique to the design of the operating policy of a blood bank. A text by Ignizio (1976) deals with GP, exclusively, as it extends the general formulation to linear integer, and nonlinear forms; it also offers a computer code with a cutting-plane option. Werczberger (1976) uses GP for industrial-location analysis involving environmental factors, and Bres et al. (1980) analyze military-manpower problems using this approach.

One form of GP model can be stated as follows:

$$\min \sum_{i=1}^{p} \left| F_i(\mathbf{x}) - T_i \right| \qquad (1)$$

Subject to $\mathbf{x} \in \mathbf{X}$

where T_i denotes the target or goal set by the decision maker for the ith linear objective function $F_i(\mathbf{x})$, and $\mathbf{X}$ represents the feasible region defined by a system of linear inequalities/or equalities. A more general formulation of the GP objective function is a weighted sum of the pth power of the deviation $\left|F_i(\mathbf{x}) - T_i\right|$ (Haimes et al., 1975). Such a formulation has been called generalized GP (Ignizio, 1976, 1981; Szidarovszky, 1979).

Returning to formulation (1) above, the objective function is nonlinear and the simplex method, with its many inherent advantages, cannot be applied directly. However, it is possible to transform (1) into a linear form, thus reducing GP to a special type of linear programming. The transformation (Charnes and Cooper, 1961) defines new slack variables d_i^+ and d_i^- such that

$$d_i^+ = \frac{1}{2}\{F_i(\mathbf{x}) - T_i| + [F_i(\mathbf{x}) - T]\} \tag{2}$$

$$d_i^- = \frac{1}{2}\{F_i(\mathbf{x}) - T_i| - [F_i(\mathbf{x}) - T_i]\} \tag{3}$$

Examination of 2 reveals that d_i^+ is the positive deviation from the ith target for the ith objective (i.e., overachievement of a goal). The second slack variable d_i^- is the negative deviation from the ith target for the ith objective (i.e., underachievement of a goal).

Adding (2) to (3), it is seen that

$$d_i^+ + d_i^- = |F_i(\mathbf{x}) - T_i|$$

Thus the objective function in formulation (4-1) can be replaced by an equivalent linear relationship. Furthermore, by subtracting (2) from (3), we get

$$F_i(\mathbf{x}) - T_i = d_i^+ - d_i^-$$

It is also required that d_i^+ and d_i^- be nonnegative, that is $d_i^+ , d_i^- \geq 0$. and, since it is not possible to have both underachievement and overachievement of a goal simultaneously, then one or both of the deviational slack variables must have a zero value; that is

$d_i^+ . d_i^- = 0.$

Fortunately, this constraint is automatically fulfilled by the simplex method. This is because the objective function will drive either (or perhaps both) d_i^+ or d_i^- to zero for all i.

Thus, an equivalent linear formulation of (1) is

$$\min \; W_0 = \sum_{i=1}^{p} \left(d_i^+ + d_i^-\right) \tag{4}$$

Subject to $\mathbf{x} \in \mathbf{X}$

$$F_i(x) - d_i^+ + d_i^- = T_i$$

$$d_i^+ , d_i^- \geq 0 \, , \; i = 1, \ldots, p$$

Once the GP model is formulated as in (4), the computational procedure can make use of the simplex method as in linear programming method described in section 1.1.

In formulation (4), both d_i^+ and d_i^- appear in the objective function and are assigned equal weights. This form of the model will attempt to achieve the goal exactly; but, if exact achievement is not possible, no preference for overachievement or underachievement of a goal is built into the model. Nor is any goal in formulation (4) given any particular weight. However, it is possible to assign priority factors and weights to goals. Only a slight modification of formulation (4) is required.

Assigning Priority Factors (Ordinal Ranking and Weights (Cardinal Ranking) to Goals

To express preference for deviations, the decision maker can assign relative weights w_i^+, w_i^- to positive and negative deviations, respectively, for each target, T_i. Since we are minimizing, choosing the w_i^+ to be larger than w_i^- would be expressing preference for underachievement of a goal (for example, such may be the case when overachievement would result in an overtime requirement).

In addition, GP allows flexibility needed to deal with cases with conflicting multiple goals. Essentially, goals can be ranked in order of importance. That is, a priority factor, $P_i (i = 1,\dots, p)$ is assigned to the deviation slack variables associated with the goals. These factors, P_i, are conceptually different from weights, as will be illustrated in the next section. It is assumed that the priorities are ordered so that for i = 1,..., p, $P_i > P_{i+1}$. Another possibility is $P_i >> P_{i+1}$ which is equivalent to stating that goal i has absolute priority over goal i + 1.

Thus, our GP model is now formulated as:

$$\min\ S_0 = \sum_{i=1}^{p} P_i (w_i^+ d_i^+ + w_i^- d_i^-) \tag{5}$$

Subject to

$$\mathbf{x} \in \mathbf{X}$$

$$F_i(x) - d_i^+ + d_i^- = T_i$$

$$d_i^+, d_i^- \geq 0,\ i = 1,\dots,p$$

Solution Method

The simplex method can also be applied to solve the problem by making some modification to the GP. Goal 1 (priority 1) is solved first by ignoring the other lower priorities. Putting this resulting goal 1 as a constraint, goal 2 (priority 2) is solved by ignoring the other lower priorities and this procedure is continued until the lowest priority is solved. There are many commercial software packages available to solve this GP. Usually before analysts solve the GP problem, the project manager has to specify priorities and weights. In practice it is often very difficult to determine appropriate priorities and weights in a specific problem.

There are many techniques proposed by researchers to resolve these problems of assigning of weights and priorities in GP models. Rustagi et al (1987) describe in their paper titled "resolving multiple goal conflicts with interactive goal programming" how "interactive goal programming" is used. In this method the problem is first solved with initial target levels and weights. On the basis of this solution, the project manager would revise the target levels and weights and the process would be repeated until an acceptable compromised is found.

Dyer et al., (1979) consider GP with "preemptive priorities", where weights are not included. Preemptive priorities are not rigidly determined and the method attempts to reorder the priorities so as to get the optimal result.

Kangas et al., (1992) suggest that the project manager's judgment of priorities is most of the time not very accurate. They give a method for determining priorities by using analytic hierarchy process. In the next example we will see how GP can be used to best satisfy our preferences in a small forestry problem.

Example 2.3

Jackson has 24 acres of fallow land available and wants to use it to increase his/her income. He can either plant fast-gro hybrid Christmas tree transplants that mature in one year, or he can fatten steers by putting part of his acreage in pasture. The trees are planted and sold in lots of 1,000. It takes 1.5 acres to grow a lot of trees and 4.0 acres to fatten a steer. The farmer is busy and only has 200 hours per year to spend on this enterprise. Experience shows it takes 20 hours to cultivate, prune, harvest, and package one lot of trees and also 20 hours per steer. There is a $1,200.00 operating budget available for the year and annual expense are $30 per lot of trees and $240 per steer. At current prices, Christmas trees will return a net revenue of $0.50 each and steers will return a net revenue of $1,000 each.

For other reasons, he/she wanted to use all of the budget allocation and that he truly hoped for an even mix of 5 steers and 5 tree lots. Achieving these three goals is more important than maximizing income, and and achieving the budget goal is at least twice as important as either of the other 2 goals.

Let x_1 = number of steers fattened per year

x_2 = number of 1000-tree lots of fast-gro-Christmas trees grown per year

The formulations of the goals are:

(1) achieving 5 steers;

$$x_1 = 5$$

(2) achieving 5 tree lots;

$$x_2 = 5$$

(3) spending all the budget;

$$240x_1 + 30x_2 = 1{,}200$$

Constraints;

(1) Land 24 acres available, 4 acres per steer, 1.5 acres per tree lot. So we get

$$4x_1 + 1.5x_2 \le 24$$

(2) Budget: \$ 1,200 available, \$ 240 per steer, \$ 30 per tree lot. So we gete

$$240x_1 + 30x_2 \le 1{,}200$$

(3) Labor 200 hours available, 20 hours per steer, 20 hours per tree lot. So we get

$$20x_1 + 20x_2 \le 200$$

Then we introduce additional variables to represent deviation from the goals.

Let d_1^+ = positive deviation (amount of overachievement) from the 5 steer goal.

d_1^- = negative deviation (amount of underachievement) from the 5 steer goal.

d_2^+ = positive deviation from the 5 tree goal.

d_2^- = negative deviation from the 5 tree goals.

d_3^+ = positive deviation from the \$ 1,200 budget goal

d_3^- = negative deviation from the \$ 1,200 budget goal

So our objective now is to:

$$Minimize\,(d_1^+ + d_1^-) + (d_2^+ + d_2^-) + 2\,(d_3^+ + d_3^-)$$

Subject to:

$$4x_1 + 1.5\,x_2 \le 24$$

$$20x_1 + 20\ x_2 \ \le 200$$

$$240x_1 + 30x_2 + d_3^- - d_3^+ = 1{,}200$$

$$x_1 + d_1^- - d_1^+ = 5$$

$$x_2 + d_2^- - d_2^+ = 5$$

$$x_1, x_2 \ \ge 0$$

Solving this formulation using LINGGO gives us the results summarized below;

Variable	value	row	slack	dual
d_1^+	0			
d_1^-	0.5	land	0	0.875
d_2^+	0	labor	30	00
d_2^-	0	budget	0	-0.0
d_3^+	0	steers	0	-0.5
d_3^-	0	trees	0	-1.0
x_1	4.5			
x_2	4.0			

We can see from this result that the solution minimizes the deviations at $x_1 = 4.5$ and $x_2 = 4.0$. All of the budget and land are used. By implementing this result he will have profit of \$ 4.0(1000x0.5) + \$4.5x1000 = \$6500.

2.4. Fuzzy Optimization

Allen et al. (1986) described forest planning or systems not only as complex but also as wicked systems. This is due to the diversified nature of the forest itself as well as the different biological, physical, and economic processes within and outside the forest ecosystem. In view of the inherent complexity of the forests, planning for their efficient use and the effective management has become an increasingly difficult task.

Concerns about the use of LP models have also been raised (Bare et.al, 1987). The main criticisms deal with the inherently deterministic nature of LP models, and their use of precise coefficients. In traditional LP models, the coefficients or parameters are assumed to be known with certainty, but in many real world forest planning situations it is very unlikely that this assumption is valid. For example, forest managers often have to deal with insufficient or imperfect information due to the inherent complexity of the system as described above. In this case, the forest managers have to be able to capture the uncertainties in their decisions

The term "uncertainty" has been widely used to describe several phenomena. It has been used to represent risks, imprecision, randomness, inaccuracy, ambiguity or inexactness. In our discussion here in this thesis, uncertainty is used to reflect any phenomena other than those regarded as random or probabilistic in nature. There are several reasons for incorporating uncertainty in forest planning. First, forest planning involves long planning horizons (e.g. several decades) with accurate long-term projections generally difficult to make and are at best only educated guesses of future outcomes. Future timber prices, for instance, are highly dependent on several variables making them difficult to predict. Moreover, most forest lands covering large diverse geographical areas produce multiple goods and services which are valued differently by forest users. Some of these uses can be adequately measured while others are inherently qualitative and difficult to quantify. Finally, forest planning often requires the incorporation of human subjectivity which is both difficult to elicit and express in quantitative terms. Therefore, the use of optimization models that can incorporate imprecise information, has become a prerequisite to comprehensive planning, particularly in complex planning environments, such as forestry. A relatively new approach called fuzzy programming may be better suited under these environments.

Basics of Fuzzy Set Theory

In this section, a formal treatment of fuzzy logic is provided by considering membership or indicator functions for fuzzy sets (objective targets) and fuzzy members for imprecise values of the technical coefficients in the decision model. This background constitutes the formal foundation for the fuzzy programming.

Fuzzy sets and membership functions

An element x of X is assigned to an ordinary (crisp) set A via the characteristic function μ_A such that: $\mu_A(x)=1$ if $x \in A$, and $\mu_A(x)=0$ if $x \notin A$. The valuation set for the function is the pair of points $\{0,1\}$. A fuzzy set $\tilde{A}$ is also described by a characteristic function, the difference being that the function now maps to all points in the closed interval $[0,1]$.

Formally, a fuzzy set $\tilde{A}$ of the universal set X is defined by its membership function $\mu_{\tilde{A}} : \mathrm{X} \rightarrow [0,1]$, which assigns to each element $x \in X$ a real number $\mu_{\tilde{A}}(x)$ in the interval [0,1], where the value of $\mu_{\tilde{A}}$ at x represents the grade or degree of membership of x in $\tilde{A}$ (Sakawa 1993). While membership functions can take on a variety of functional forms, linear specifications are often employed.

As an example of fuzzy membership, consider the set of "natural forests". It is clear that old-growth forests belong to this set, they have a degree of membership equal to 1. As we consider progressively heavier logged forests, the descriptor "natural" becomes less apt. Is a selectively logged forest "natural"?. To capture the uncertainty surrounding their membership in the set of "natural forests", partly logged forests are assigned a partial degree of membership, something less than one. This is an example of a one-sided fuzzy set. Membership in this set approaches zero as the exploitation pressure increases.

In this regard, fuzzy set theory can be used to deal with unclear objectives. This will be illustrated with an example. An objective of the land-use decision model developed below will be to preserve wilderness by setting land aside as protected areas. The question is: how much land should be protected? According to some government guidelines 15% of the land should be protected. Since "undershooting" of this goal will be politically sensitive, it can be argued that 15% serves to define the lower limit and a lower percentage of the land base as wilderness will be unacceptable and have a membership value of 0. On the other hand, there are many who would argue that more land should be set aside. Claims up to 35% have been put forward. If we adopt 35% as a perfectly satisfactory level of forest protection, then the linear membership function describing the fuzzy set for a forest x is:

$\mu_i(x) = 1,$ if PA $\geq$ 35%

$\mu_i(x) = (\text{PA} - 15)/(35 - 15),$ if 15% $\leq$ PA $\leq$ 35%

$\mu_i(x) = 0,$ if PA $\leq$ 15%

where PA refers to the percentage of the land base that is to be protected. If the solution to the optimization problem allocates 25% of the land base to protected areas, $\mu_i(x) = 0.50$.

The preceding definitions have employed the concept of a normalized fuzzy set. A fuzzy set A, defined over a finite interval, is said to be normal if there exists an $x \in X$ such that $\mu_A(x) = 1$, and $\mu_A(x) \leq 1 \;\; \forall x \in X$.

Set theoretic operations are defined for fuzzy sets. Among these are the concepts of containment, complement, intersection and union. A fuzzy set A is contained in the fuzzy set B ($\tilde{A}$ is a subset of $\tilde{B}$), if and only if the membership function of $\tilde{A}$ is less than or equal to that of $\tilde{B}$ everywhere on X:

$\tilde{A} \subseteq \tilde{B} \Leftrightarrow \mu_A(x) \leq \mu_B(x)$ for all $x \in X$.

The complement of $\tilde{A}$ (written as $\overline{A}$) is defined as:

$\mu_{\overline{A}}(x) = 1 - \mu_{\tilde{A}}(x)$.

The intersection of fuzzy set $\tilde{A}$ and $\tilde{B}$ is defined as:

$\tilde{A} \cap \tilde{B} \Leftrightarrow \mu_{(\tilde{A} \cap \tilde{B})} = \min\{\mu_A(x), \mu_B(x)\}$ for all $x \in X$, and the union as:

$\tilde{A} \cup \tilde{B} \Leftrightarrow \mu_{(\tilde{A} \cup \tilde{B})} = \max\{\mu_A(x), \mu_B(x)\}$ for all $x \in X$.

Hence, the intersection $\tilde{A} \cap \tilde{B}$ is the largest fuzzy set contained in both $\tilde{A}$ and $\tilde{B}$, and the union $\tilde{A} \cup \tilde{B}$ is the smallest fuzzy set containing both $\tilde{A}$ and $\tilde{B}$.

While both union and intersection of fuzzy sets are commutative, associate and distributive, as is the case for ordinary or crisp sets, fuzzy logic deviates from crisp logic because, if we do not know $\tilde{A}$ with certainty, then its complement $\overline{A}$ is also not known with certainty. Thus, $\overline{A} \cap \tilde{A}$ does not produce the null set as is the case for crisp sets (where $A^c \cap A = \Phi$). Thus, fuzzy logic violates the "law of non-contradiction". It also violates the "law of the excluded middle" because the union of a fuzzy set and its complement does not equal the universe of discourse – the universal set X. Thus, $\tilde{A}$ is properly fuzzy iff $\tilde{A} \cup \overline{A} \neq X$ (Kosko 1992).

Another concept required for model building with fuzzy sets is that of the α-level set. The α-level set A_α is simply that subset of $\tilde{A}$ for which the degree of membership exceeds the level α, and is itself a crisp set (an element either meets the required level of α or it does not).

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\},\ \alpha \in [0,1].$$

A_α is an upper level set of $\tilde{A}$. The use of α-level sets provides a means of transferring information from a fuzzy set into a crisp form. Defining an α-level set is referred to as taking an α-cut, cutting off that portion of the fuzzy set whose members do not have the required membership or possibility value. It can be argued that the level of the α-cut is a measure of the faith that the decision maker has in the reliability of the imprecise coefficient. The more the decision makers' confidence, the higher the α-cut is set.

Fuzzy Linear Programming

Fuzziness can be modeled in several ways depending upon the nature of imprecision, the context in which uncertainty occurs, and how it is accommodated in the problem. For instance, in a mathematical programming setting, fuzziness can be restricted to the constraints, the objective function, or both; and fuzziness may be manifested as fuzzy numbers (i.e., coefficients in the objective function or constraints) or as fuzzy sets (i.e., the objective function or constraints).

Before formally defining the fuzzy LP, note that the classical LP problem can be restated as follows(Sukawa, 1993):

$$\begin{aligned} &\text{minimize} \quad z = \mathbf{cx} \\ &\text{subject to} \quad \mathbf{Ax} \le \mathbf{b} \\ &\qquad\qquad\quad \mathbf{x} \ge \mathbf{0} \end{aligned} \tag{2.4.1}$$

where $\mathbf{c} = (c_1,...,c_n)$, $\mathbf{x} = (x_1,...,x_n)^T$, $\mathbf{b} = (b_1,...,b_m)^T$, $\mathbf{A}$ = m x n matrix.

Zimmermann (1976) proposed to soften the rigid requirements of the decision maker to strictly minimize the objective function and to strictly satisfy the constraints. In other words, the goal of the decision maker can be expressed as a fuzzy set and the solution space is defined by constraints that can be modeled by fuzzy sets. In such situation a better model than (2.4.1) would be:

Find $\mathbf{x}$ such that

$$\begin{aligned} &\mathbf{cx} \lesssim z \\ &\mathbf{Ax} \lesssim \mathbf{b} \\ &\mathbf{x} \ge \mathbf{0} \end{aligned} \tag{2.4.2}$$

where the symbol "$\lesssim$" denotes a relaxed or fuzzy version of the ordinary inequality "$\leq$". These fuzzy inequalities mean that the objective function $\mathbf{cx}$ should be essentially smaller than or equal to z, vaguely specified by decision makers or maximum value of the crisp systems of the problem, and the constraints $\mathbf{Ax}$ should be essentially smaller than or equal to $\mathbf{b}$, respectively.

By substituting $\begin{pmatrix} \mathbf{c} \\ \mathbf{A} \end{pmatrix} = \mathbf{B}$ and $\begin{pmatrix} z \\ \mathbf{b} \end{pmatrix} = \mathbf{d}$, (2.4.2) becomes:

Find $\mathbf{x}$ such that

$$\begin{aligned} &\mathbf{Bx} \lesssim \mathbf{d} \\ &\mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.4.3}$$

Each of the $(m+1)$ rows of (2.4.3) shall now be represented by a fuzzy set, whose membership functions are $\mu_i(\mathbf{x})$, $i = 1,\ldots,m+1$. $\mu_i(\mathbf{x})$ can be interpreted as the degree to which $\mathbf{x}$ fulfills (satisfies) the fuzzy inequality $(\mathbf{Bx})_i \lesssim (\mathbf{d})_i$.

Denote the ith fuzzy inequality $(\mathbf{Bx})_i \lesssim (\mathbf{d})_i$, $i = 0, 1, \ldots, m$, Zimmerman used the following linear membership function:

$$\mu_i\left((Bx)_i\right) = \begin{cases} 1 & ;(Bx)_i \leq d_i' \\ 1 - \dfrac{(Bx)_i - d_i}{p_i} & ;d_i \leq (Bx)_i \leq d_i + p_i \text{ , } i = 1,\ldots,m+1 \\ 0 & ;(Bx)_i \geq d_i + p_i \end{cases}$$

(2.4.4)

where each p_i is a subjectively chosen constant expressing the limit of the admissible violation of the ith inequality. This ensures that the ith membership function should be

1 if the ith constraint is well satisfied, 0 if the *i*th constraint is violated beyond its limit p_i, and linear in between.

Following the fuzzy programming method of Bellman and Zadedh (1970) to choose x^* such that

$$\mu(x^*) = \max_{x\geq 0} \min_{i=0,\ldots,m} \{\mu((Bx)_i)\}. \tag{2.4.5}$$

In other words, the problem is to find the $x^* \geq 0$ which maximizes the smallest membership function values(i.e. try to minimize the deviation of the inequalities from being fully satisfied). Substituting (2.4.5) to (2.4.4) yields, after some rearrangements (Zimmermann 1976),

$$\max_{x\geq 0} \min_{i=0,\ldots,m} \left\{1 - \frac{(Bx)_i - d_i}{p_i}\right\} \tag{2.4.6}$$

Introducing one new variable λ, this problem can be transformed into the following equivalent conventional LP problem:

$$maximize\ \lambda$$

subject to

$$\begin{aligned} &\lambda\, p_i + (Bx)_i \leq d_i + p_i \quad i = 1,\ldots m+1 \\ &0 \leq \lambda \leq 1 \end{aligned} \tag{2.4.7}$$

$$x \geq 0.$$

If the optimal solution to (2.4.5) is the vector (λ^*, x^*) then x^* is the maximizing solution (2.4.5) of model (2.4.2) . We should realize that the maximizing solution can be found by solving one standard (crisp) LP.

2.5. Shortest Paths

Transportation systems in forests are one of the most crucial decisions that have to be made. The determination of the shortest route (or path) through a network of available routes is often an important step in planning transportation. A system of forest routes may be described as a network, a collection of interconnected segments or links. Each link describes a unique path between two adjacent nodes. A node is any feature that might be treated as the point of departure or destination of some path through the network, such as a landing or mill (Carson et al. 1978). Nodes are also commonly used to indicate points at which road design standards change or there is a marked change in grade or curvature. Such changes would be expected to influence costs of hauling logs (Byrne et al. 1960) and may therefore be of interest in the solution of many transportation problems.

Planning a network for transportation in a forest is also very important since we can reduce transportation costs by having an efficient network. Before deciding on a permanent system of forest roads, we usually create a network road plan and put a cost value on each road segment in the network. Then we analyze the network road planning to get a more efficient alternative road network. The shortest path algorithm is commonly utilized for this purpose (Carson, et. al 1978).

The shortest path can be found by using a linear optimization method or an efficient graphical solution procedure (Mandl 1979). We will first use the formulation of shortest distances (paths) as a linear optimization model.

Assume a directed network given by $N = (X, A)$ has a set of nodes X and set of arcs A. Suppose that each arc $j \in A$ has a length or other cost measurement c_j. If we want to find the shortest distance and route from nodes $s \in X$ to node $t \in X$ then this can be formulated as a linear optimization model as follows: let x_j be a variable which has value one if the arc j is used on the route from s to t, and is otherwise zero. The problem becomes then

$$\text{minimize} \quad \sum_{j \in A} c_j x_j \tag{1}$$

subject to

$$\sum_{\substack{j \in A \text{ with initial} \\ \text{vertex } k \in X}} x_j - \sum_{\substack{j \in A \text{ with terminal} \\ \text{vertex } k \in X}} x_j = \begin{cases} 1 \text{ for } k = s \in X \\ 0 \text{ for all other } k \in X \\ -1 \text{ for } k = t \in X \end{cases} \tag{2}$$

$$x_j \geq 0 \text{ for all } j \in A \tag{3}$$

One convenient property of this problem is that there is always a solution (Mandl 1979) in which the variables all have values 0 or 1, even if the variables are continuous. Hence, there is no need to specify this condition. Equations (2) are called the conservation equations and simply state that if a route enters a node then it must also leave the node, unless this node is the origin or destination.

Equation (2) may be written as

$$Bx = e \tag{4}$$

where B denotes the network incidence matrix, x the flow vector and e the right-hand side vector of equation (2).

The entries of B are defined by: $b_{ij} = \begin{cases} 1 \text{ if arc } j \text{ starts at vertex } i \\ 0 \text{ if arc } j \text{ neither starts nor ends at vertex } i \\ -1 \text{ if arc } j \text{ ends at vertex } i. \end{cases}$

Obviously, both problem (1) – (3) may be solved with the simplex algorithm. However due to the special structure of the incidence matrix, faster algorithms are available. One of the algorithms is called Dijkstra's algorithm (Winston 1987). For this algorithm it must be assumed that the cost $c_j \geq 0$ for all arcs $j \in A$. However, for problems we are considering this is not a restrictive assumption, because negative costs do not have a practical meaning. The algorithm is divided into two parts: first the shortest distance are found, and, secondly, the associated shortest paths.

Algorithm D (Dijkstra's algorithm for shortest distances)

To each node $x \in X$ a value $v(x)$ is assigned, which at the end will denote the shortest distance from some node $s \in X$. This value $v(x)$ may be temporary, indicating that $v(x)$ could still be reduced, or permanent, indicating that this value denotes the shortest distance from s to node x.

D1 [Initialization]. Set $v(s) \leftarrow 0$ and mark this value as permanent. Set $v(x) \leftarrow \infty$ for all $x \in X$ and $x \neq s$ and mark these values as temporary. Set $p \leftarrow s$, the current working node.

D2 [Updating the values]. For all nodes x which have temporary values $v(x)$ and which are connected by an arc from p, set $v(x) \leftarrow \min_x [v(x), v(p) + c(p,x)]$, where $c(p,x)$ is the length from node p to node x.

D3 [Fixing a value as permanent]. Among all nodes x with associated temporary values $v(x)$ choose a node y for which $v(y) = \min v(x)$. Mark the value $v(y)$ as permanent and set $p \leftarrow y$. If no such nodes x exist (so y cannot be found), go to D4.

D4 [Termination]. If the shortest distance from node s to node t is wanted and if $p = t$, then the algorithm terminates with $v(t)$ as the answer. If $p \neq t$ we return to D2. If the shortest distance from node s to all other nodes are wanted then the algorithm terminates if all values $v(x)$ are permanent; otherwise return to D2.

Algorithm $\bar{D}$ (algorithm for shortest routes)

To apply this algorithm, algorithm D has first to be solved and the values $v(x)$ are used as an input to algorithm $\bar{D}$.

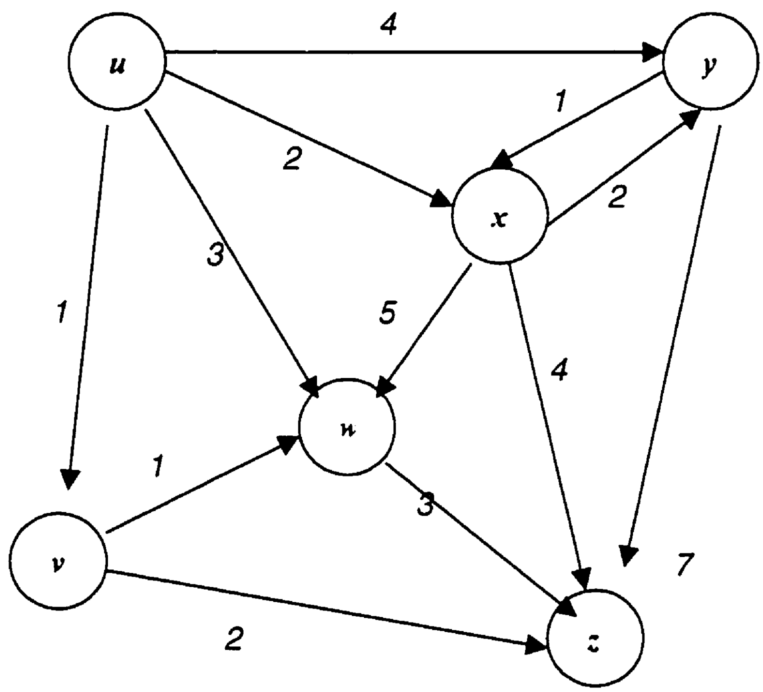
$\bar{D}1$ [Initialization]. Set $p \leftarrow t$, where t is the node for which a shortest path from s is required. The value $v(t)$ from algorithm D must be permanent. p is the current working node.

$\bar{D}2$ [Iteration]. From among all nodes $x \in X$ from which there is an arc from x to node p find the node y, for which $v(y) + c(y,p) = v(p)$, where arc $c(y,p)$ connects node y with node p. Store arc $c(y,p)$ as belonging to the shortest route.

$\bar{D}3$ [Termination]. Set $p \leftarrow y$. If $p = s$ the algorithm terminates with the sequence of arcs belonging to the shortest s to t route as the result. If $p \neq s$ return to $\bar{D}2$.

Example 1.5;

Consider the following network transportation problem, $(X = \{u, v, w, y, z\})$ with the associated costs, and structure shown in the diagram below. Node u is a landing and node z is the mill. We want to find the most efficient route to drive logs from the landing to the mill through the existing network transportation.



We now want to find the shortest path from node u to node z, and therefore first apply algorithm D.

D1: $v(u) \leftarrow 0$, $v(v) \leftarrow v(w) \leftarrow v(x) \leftarrow v(y) \leftarrow v(z) \leftarrow \infty$, $p \leftarrow u$.

$v(u)$ is permanent.

D2: $v(y) \leftarrow \min[\infty, 0 + 4] = 4$ $\quad$ $v(w) \leftarrow \min[\infty, 0 + 3] = 3$

$$v(x) \leftarrow \min[\infty, 0+2] = 2 \qquad v(v) \leftarrow \min[\infty, 0+1] = 1$$

D3: $v(v)$ is permanent. $p \leftarrow v$

D4: Return to D2.

D2: $v(w) \leftarrow \min[3, 1+1] = 2$, $\qquad v(z) \leftarrow \min[\infty, 1+2] = 3$

D3: $v(x)$ or $v(w)$ is permanent. Choose $v(x)$. $p \leftarrow x$.

D4: Return to D2.

D2: $v(y) \leftarrow \min[4, 2+2] = 4 \qquad v(w) \leftarrow \min[2, 2+5] = 2$

$$v(z) \leftarrow \min[3, 2+4] = 3$$

D3: $v(w)$ is permanent. $p \leftarrow w$

D4: Return to D2.

D2: $v(z) \leftarrow \min[3, 2+3] = 3$

D3: $v(z)$ is permanent. $p \leftarrow z$.

D4: Terminate. The shortest distance from u to z is 3.

Now we have to use algorithm $\bar{D}$ for computing the shortest route.

$\bar{D}1$: $p \leftarrow z$.

$\bar{D}2$: $v(y) + c(y, p) = 4 + 7 \neq 3$, $v(v) + c(v, p) = 1 + 2 = 3$. $c(v, p)$ belongs to the shortest route.

$\bar{D}3$: $p \leftarrow v$. Return to $\bar{D}2$.

$\bar{D}2$: $v(u) + c(u, p) = 0 + 1 = 1$. $c(u, p)$ belongs to the shortest route.

$\bar{D}3$: $p \leftarrow u$. Terminate. The shortest route is the arc sequence $c(u,p) - c(v,p)$. When the shortest distance and routes between all pairs of nodes of a network are required, a feasible way for obtaining the information is to apply algorithm D and $\bar{D}$ for each node of the network.

2.6. Dynamic Programming

Operations Research techniques can be used to tackle the increased complexity of resource management and resource management and resource problems entail decisions which are sequential, risky and irreversible. Dynamic programming (DP) is a versatile technique with considerable scope for helping to solve such problems.

The ability of DP to decompose big problem into small problems, where the small problems interrelate to each other sequentially, makes it become a very useful tool for optimization. Many people have successfully used this technique to solve not only resource allocation (resource management) but also production problems such as bucking tree problem. Pnevmaticos et al (1972) show how DP can be applied to select the optimal bucking patterns for single logs, assuming uniform taper, with no defective stem sections, and probabilistic grading of wood quality. Haight et al.(1985) show that the incorporation of stand growth and yield simulators, whether they involve whole stands or single trees, and whether they are free of, or dependent on, either distance or diameter, into DP algorithms has improved the analysis of silvicultural investment decisions for even-age stand management by allowing the simultaneous determination of optimal timing and intensity of thinning and rotation age.

To solve resource allocation problems in which limited resources must be allocated among several activities, people usually use LP. To use LP to do resource allocation, three assumptions must be satisfied:

Assumption 1: The amount of a resource assigned to an activity may be any non-negative number.

Assumption 2: The benefit obtained from each activity is proportional to the amount of the resource assigned to the activity.

Assumption 3: The benefit obtained from more than one activity is the sum of the benefits obtained from the individual activities.

Even if assumptions 1 and 2 do not hold, DP can be used to solve resource allocation problems efficiently when assumption 3 is valid and when the amount of the resource allocated to each activity is a member of a finite set.

DP is an approach to problem solving that permits decomposing one large mathematical model, that may be very difficult to solve, into a number of smaller problems that are usually much easier to solve (Schmidt et al. 1981). Moreover, the DP approach allows us to break up a large problem in such a fashion that once all the smaller problems have been solved, we are left with an optimal solution to the large problem. We shall see that each of the smaller problems is identified with a stage of the DP solution procedure. As a consequence, the technique has been applied to many decision problems that are multi stage in nature. Often, multiple stages are created because a sequence of decisions must be made over time. For example, a problem of determining an optimal decision over a 100-year horizon might be broken into 10 smaller stages , where each stage requires an optimal decision over a 1-decade horizon. In most cases, each of these smaller problems cannot be considered to be completely independent of the others, and this is where DP is helpful.

DP is an approach that can be used fruitfully in the modeling and analysis of many diverse operational problems. As a modeling tool it provides a framework for building mathematical relationships that describe the operational behavior and performance of multistage decision processes. As an analysis tool, it provides a structure whereby a large problem (in terms of the number of decision variables) can be decomposed into a series of interrelated small problems. These small problems are solved sequentially utilizing their interrelationships until, ultimately, the solution to the large problem is obtained. Each of the small problems is associated with a stage in the solution process. This staging implies that the problem is separable, that is, can be validly decomposed into such stages.

There are several basic features associated with using a DP rationale to define an optimal solution to a mathematical programming problem. They are

1) The problem can be divided into stages with a policy decision required at each stage.
2) Each stage has a number of states associated with it.
3) The effect of the policy decision at each stage is to transform the current state into a state associated with the next stage.
4) Given the current state of the system in a particular stage, an optimal policy for subsequent stages is independent of the policy adopted in previous stages.
5) The solution procedure begins by finding the optimal policy for each state of the last stage.

6) A recursive relationship is available which identified the optimal policy for each state with $N-k$ stages remaining ($k = 0,1,\ldots,N-1$).

7) Using this recursive relationship, the solution procedure moves backward stage-by-stage, each time

Finding the optimal policy for each state of that stage, until it finds the optimal policy when starting at the initial stage.

These basic features provide the framework through which a dynamic programming solution is implemented. Having indicated that the problem is to be decomposed into stages, it is important to identify specifically how a typical stage is represented. A typical stage (here denoted the ith stage) can be represented by Fig. 2.6.1 and is characterized by five fundamental factors (Schmidt, 1981):

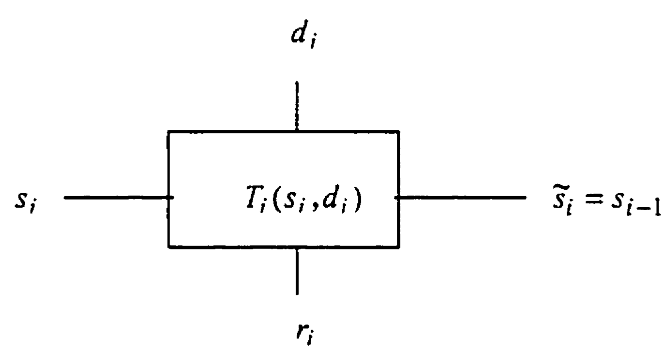


Fig. 2.6.1. Typical stage diagram

(1) an input stage s_i, which gives all relevant information about inputs to the stage; s_i is called the initial stage of stage i as it gives a description of the system at the beginning of the stage;

(2) stage transition functions $T_i(.)$, sometimes called the stage-coupling functions, which express each component of the output state as a function of the input state and stage decisions;

(3) an output state $\tilde{s}_i$, which gives all relevant information about outputs from the stage; $\tilde{s}_i$ is called the final state of stage i as it gives a description of the system at the end of the stage: $\tilde{s}_i = T_i(s_i, \mathrm{d}_i) = s_{i-1}$;

(4) a decision d_i, which controls the operation of the stage;

(5) a stage return r_i, which is a variable that measures the utility or performance of the stage as a function of the input state and decision:

If the objective function $F(d)$ is separable into individual stage returns r_i which are additive in their effect on the total objective, that is, $F(d) = \sum_{i=1}^{N} r_i$ then the basic optimization principle of dynamic programming can be stated in a maximization context as follows:

(a). For every possible input state value s_1, in the first stage of analysis, the optimum decision d_1, will maximize $f_1(s_1) = R_1(s_1, \mathrm{d}_1)$ and for each of the other stages.

(b). For every possible input state value s_k in stage k of the analysis, the optimal decision d_k will maximize $f_1(s_k) = R_k(s_k, \mathrm{d}_k) + f_{k-1}^*(s_k, d_k)$ where $f_k(.)$ is the cumulative return for stage k and $f_{k-1}^*(.)$ is the optimal cumulative return from stage $k-1$ given in terms of each input state to stage k. The key to formation of the cumulative return function is recognizing that each input state of stage $k-1$ for which f_{k-1}^* is defined can be associated with a specific input state-and-decision pair at stage k. This relationship is explicitly defined by the stage coupling function at stage k. That is $s_{i-1} = \tilde{s}_i = T_i(s_i, \mathrm{d}_i)$.

Example.

A private forestry company has \$6000 to invest in growing three types of trees. If d_j dollars (in thousands) are invested to grow trees type j (investment j), then a net present value (in thousands) of $r_j(d_i)$ is obtained, where the $r_j(d_i)$'s are as follows:

$$r_1(d_1) = 7d_1 + 2 \qquad (d_1 > 0),\ r_1(0) = 0$$

$$r_2(d_2) = 3d_2 + 7 \qquad (d_2 > 0),\ r_2(0) = 0$$

$$r_3(d_3) = 4d_3 + 5 \qquad (d_3 > 0),\ r_3(0) = 0$$

The amount placed in each investment must be an exact multiple of \$1000. In order to maximize the net present value obtained from the investments, how should the company allocate the \$6000?.

The return on each investment is not proportional to the amount invested in it (for example, $16 = r_1(2) \neq r_1(1) = 18$). Thus, LP cannot be used to find an optimal solution to this problem.

Mathematically, the company's problem may be expressed as

$$\max \{r_1(d_1) + r_2(d_2) + r_3(d_3)\}$$

such that $d_1 + d_2 + d_3 = 6$

d_j non-negative integer (j = 1, 2, 3).

To formulate the company's problem as a DP problem, we begin by identifying the stage. The stage should be chosen so that when one stage remains the problem is easy to solve. Then, given that the problem has been solved for the case where one stage remains, it should be easy to solve the problem where two stages remain, etc. Clearly, it would be easy to solve the problem in which only one investment was available, so we define stage t to represent a case where funds must be allocated to investments t, $t+1$, ...,3.

For a given stage, what must we know to determine the optimal investment amount? Simply how much money is available for investments t, $t+1$, ...,3. Thus, we define the state at any stage to be the amount of money (in thousands) available for investments t, $t+1$, ...,3. Since we can never have more than $6000 available, the possible states at any stage are 0,1,2,3,4,5, and 6. We define $f_t(d_t)$ to be the maximum net present value (NPV) that can be obtained by investing d_t thousand dollars in investments t, $t+1$, ...,3. Also define $x_t(d_t)$ to be the amount that should be invested in investment t in order to attain $f_t(d_t)$. We start to work backwards by computing $f_3(0)$, $f_3(1)$, . . ., $f_3(6)$ and then determine $f_2(0)$, $f_2(1)$, . . ., $f_2(6)$. Since $6000 is available for investment in investments 1,2, and 3, we terminate our computations by computing $f_1(6)$. Then we retrace our steps and determine the amount that should be allocated to each investment .

Stage 3 computations

We first determine $f_3(0)$, $f_3(1)$, . . ., $f_3(6)$. We see that $f_3(d_3)$ is attained by investing all available money (d_3) in investment 3. Thus,

$f_3(0)=0$ $\qquad x_3(0)=0$

$f_3(1)=9$ $\qquad x_3(1)=1$

$f_3(2)=13$ $\qquad x_3(2)=2$

$f_3(3)=17$ $\qquad x_3(3)=3$

$f_3(4)=21$ $\qquad x_3(4)=4$

$f_3(5) = 25$ $\qquad$ $x_3(5) = 5$

$f_3(6) = 29$ $\qquad$ $x_3(6) = 6$

Stage 2 computations

To determine $f_2(0), f_2(1), \ldots, f_2(6)$ we look at all possible amounts than can be placed in investment 2. To find $f_2(d_2)$, let x_2 be the amount invested in investment 2. Then an NPV of $r_2(x_2)$ will be obtained from investment 2, and an NPV of $f_3(d_2 - x_2)$ will be obtained from investment 3. Since x_2 should be chosen to maximize the net present value earned from investments 2 and 3, we write

$$f_2(d_2) = \max_{x_2} \{r_2(x_2) + f_3(d_2 - x_2)\}$$

where x_2 must be a member of $\{0, 1, \ldots, d_2\}$. The computations for $f_2(0), f_2(1), \ldots,$ $f_2(6)$ and $x_2(0), x_2(1), \ldots, x_2(6)$ are given in Table 2.6.1.

Stage 1 computations

Following from stage 2, we write $f_1(6) = \max_{x_1} \{r_1(x_1) + f_2(6 - x_1)\}$ where x_1 must be a value from $\{0,1,2,3,4,5,6\}$. The computations for $f_1(6)$ are given in Table 2.6.2.

Determination of the optimal investment

Since $x_1(6) = 4$, the company invests \$4000 in investment 1. This leaves 6000-4000 = \$2000 for investment 2 and 3. Hence the company should invest $x_2(2) =$ \$1000 in investment 2. Then \$1000 is left for investment 3, so the company chooses to invest $x_3(1)$ =\$1000 in investment 3. Hence the company can attain a maximum net present value of $f_1(6) =$ \$49,000 by investing \$4000 in investment 1, \$1000 in investment 2, and \$1000 in investment 3.

Table 2.6.1. Computation for $f_2(0), f_2(1), \ldots, f_2(6)$

d_2	x_2	$r_2(x_2)$	$f_3(d_2 - x_2)$	NPV from investments 2,3	$f_2(d_2)$ $x_2(d_2)$
0	0	0	0	0*	$f_2(0)=0$ $x_2(0)=0$
1	0	0	9	9	$f_2(1)=10$
1	1	10	0	10*	$x_2(1)=1$
2	0	0	13	13	$f_2(2)=19$
2	1	10	9	19*	$x_2(2)=1$
2	2	13	0	13	
3	0	0	17	17	
3	1	10	13	23*	$f_2(3)=23$
3	2	13	9	22	$x_2(3)=1$
3	3	16	0	16	
4	0	0	21	21	
4	1	10	17	27*	
4	2	13	13	26	$f_2(4)=27$
4	3	16	9	25	$x_2(4)=1$
4	4	19	0	19	
5	0	0	25	25	
5	1	10	21	31*	$f_2(5)=31$
5	2	13	17	30	$x_2(5)=1$
5	3	16	13	29	
5	4	19	9	28	
5	5	22	0	22	
6	0	0	29	29	
6	1	10	25	35*	$f_2(6)=35$
6	2	13	21	34	$x_2(6)=1$
6	3	16	17	33	
6	4	19	13	32	
6	5	22	9	31	
6	6	25	0	25	

Table 2.6.2. Computation for $f_1(6)$

d_1	x_1	$r_1(x_1)$	$f_2(6-x_1)$	NPV from investments 1 - 3	$f_1(6)$ $x_1(6)$
6	0	0	35	35	$f_1(6)=49$
6	1	9	31	40	$x_1(6)=4$
6	2	16	27	43	
6	3	23	23	46	
6	4	30	19	49*	
6	5	37	10	47	
6	6	44	0	44	

Chapter 3
Literature Review

Forests cover approximately 31% of the earth's land surface (Sedjo and Lyon 1990). They provide many natural resources that benefit individuals, corporations and governments, and they contain much of the world's biodiversity that is so essential to the integrity of the earth's biosphere. Forest management, once the sole domain of the professional forester who attempt to regulate forests to maximize the value of the timber and other natural resources extracted from the forest, has taken center stage as many powerful interests compete aggressively to have forests managed to satisfy their often conflicting objectives.

Forest management has changed greatly over the last few decades. Initially, relatively simple stand rotation decision-making was performed (i.e. deciding when to cut individual stands to maximize the present net value of the timber). Then industrial agricultural approach was adopted for the production of timber from large forest management units while attempting to reconcile conflicting demands for non-timber resources. In the current era, environmental concerns have become a major factor in resource exploitation in many forested areas. It is therefore imperative that foresters and operational researchers seek methods to solve those problems in order to get optimal results.

Some foresters and operational researchers have attempted to use mathematical approaches to forestry problems. Some of these are described in this chapter, focusing on four aspects of forestry: resource allocation, spatial consideration, road construction and forest products.

3.1. Resource Allocation

3.1.1. Multiobjective

When allocating resources in forestry practice there are usually multiple objectives. The objectives are often measured in different measurement units, and the goals are incommensurable. Forest analysts usually utilize goal programming to resolve this problem.

Goal Programming

Goal programming (GP) is a very popular method in forestry problems where there are multiple and conflicting objectives. Goal programming was introduced into forestry management for the first time by Field (1973). He was motivated to introduce this method because of the two major weaknesses of ordinary linear programming, which had dominated in forestry problems up to that time.

Firstly, linear programming yields an optimal solution to a quantitative allocation problem only if a feasible solution exists. Feasibility is assured if the requirements specified by the analyst and the constraints imposed by the problem environment are all mutually consistent. But, inconsistencies are not always readily apparent. For example, it may not be obvious, prior to the analysis, that limited resources preclude the simultaneous satisfaction of a minimum desired timber yield goal and a watershed management objective. In contrast, the objectives specified in a goal programming format are approached as closely as possible but need not all be met completely. This flexibility allows the specification of a problem in terms of multiple conflicting goals and the allocation of resources according to subjective priorities.

Secondly, even if feasible solutions exist, in linear programming there can be only one optimization criterion. Whatever measure is associated with the objective specified by this criterion, the outcomes of several conflicting activities must be included and must be expressed in the common units of measurement. This requirement has two particularly serious effects. First, analysts attempting to apply linear programming to problems involving incommensurable values are tempted to search for, inaccurate but easily computed, indirect measures of relatively intangible results. Thus, for example, vacation expenditures are used as a surrogate gauge of outdoor recreation benefits, and a wilderness preserve is valued in terms of timber harvests foregone. Second, even when a clearly valid relationship between the optimization criterion standard and a particular activity does exist, that relation may be very difficult to specify. For example, Goal programming allows not only the simultaneous consideration of resources allocation to activities whose outcomes cannot be valued in like terms, but it also permits the analyst to specify directly activities whose levels can be associated with a common measure. For example, the consequences of a shortage of pulpwood at a mill can be expressed in cords rather than requiring the difficult estimate of overall dollar impact of such a shortage on the firm's operating costs and sales revenue.

Field used GP in advising a small woodland owner how best to satisfy immediate and long-range goals. The objective function is expressed as a weighted sum of the deviations from the goals. The weights are priority factors that reflect the priorities of the different goals. The general form of the GP proposed is:

$$\textit{minimize } z = \underline{w} . \underline{d}^{+} + \underline{w} . \underline{d}^{-}$$

subject to

$$\underline{A}\,\underline{x} - \underline{d}^{+} + \underline{d}^{-} = \underline{b}$$
$$\underline{B}\,\underline{x}\ \{\leq, =, \geq\}\underline{h}$$
$$x_j,\, d_k^{+},\, d_k^{-} \geq 0 \; for\; j = 1,\ldots,n;\;\; k = 1,\ldots,m$$

At most only one of d_k^{+} and d_k^{-} is non-zero, $\underline{w}$ is a $1\,x\,m$ vector of priority factors, $\underline{d}^{+}$ and $\underline{d}^{-}$ are $m \times 1$ vectors representing, respectively, positive and negative deviation from goals, where d_k^{+} is the kth entry in $\underline{d}^{+}$, d_k^{-} is the kth entry in $\underline{d}^{-}$, $\underline{A}$ is an $m \times n$ matrix which expresses the technical relationship between goals and sub-goals, $\underline{x}$ is an $n \times 1$ vector of decision variables called subgoals, $\underline{b}$ is an $m \times 1$ vector of desired goal attainment levels, $\underline{B}$ is a $p\ x$ n matrix describing the relationship between subgoals and specified constraints on subgoals, and $\underline{h}$ is a $p \times 1$ vector of constraint levels imposed on subgoals. The proposed procedure for determining the priority factors and weights is as follows:

(1) Formulate the problem with no priority factors or weights and solve. If all goals are met, stop. If one or more goals are not met, go to step 2.

(2) Define priorities for the set of goals and establish the weights w_k.

Another example of using GP to reconcile conflicting objective in a forestry problem us given by Kao and Brodie (1979). Some Managers accept even-flow harvest as a necessary feature of harvest control. That is, exactly the same amount of timber is cut during each period in the planning horizon. A fully regulated sustained yield should provide a constant flow of forest production, as well as allegedly more stable income and employment. To be fully regulated, the forest must have a normal age class distribution; that is, each age-class must have the same area, the number of age-classes must equal the rotation period, and no age classes can be older than the rotation period. For a given planning period, if an equal amount of volume is cut each cutting cycle, then the final age-class distribution may not be regulated. If we want the final age-class distribution to be regulated, then the harvest at each cutting cycle usually cannot be the same. GP is a good technique to compromise the conflict of the three goals: even-flow harvesting, regulating the stand, and maximizing the present net worth from the harvest. The specific form of the Goal Programming model is developed in detail in the next five pages.

Let

X_{ij} = acres harvested in age-class j in the *ith* period

P_{ij} = unit price of stumpage in age-class j harvested in the *ith* period. (Stumpage is timber in unprocessed form as found in the woods. Normally it means the physical content of standing trees, within a contiguous area, whether live or dead.)

V_j = volumes per acre of age-class j

a_j = initial area of age-class j

n = oldest age-class in the initial stand

A = total area = $\sum_{j=1}^{n} a_j$

p = cutting cycle (the interval between harvests in an uneven-aged stand, of the planning period.)

N = rotation age (the interval between one regeneration harvest and the next regeneration harvest.)

m = the maximum age-class the stand can reach during the planning period.

Constraints

Certain constraints must be maintained or the problem is infeasible. Harvesting must be limited to the initial forest and its subsequent growth. That is, the area of trees in age class *j* that are cut in the first period cannot exceed the initial area of age-class *j*. The area cut in age-class 1 in the second period cannot exceed that cut in the first period from all age-classes. The area cut in age-class *j*, *j*>*2* in the second period cannot exceed the initial area of age-class *j-1* left after the first cut, and so on.

For the first cut:

[1] $X_{1,j} \le a_j$ where $j = 1,2,3,\ldots,n$

For the second cut,

[2a] $X_{2,j} \le a_{j-1} - X_{1,j-1}$ where $j = 2,3,\ldots,n+1$

For the third cut,

$$X_{3,1} \le \sum_{j=1}^{n+1} X_{2,j}$$

$$X_{3,2} \le \sum_{j=1}^{n} X_{1,j} - X_{2,1}$$

$$X_{3,3} \le a_1 - X_{1,1} - X_{2,2}$$

$$X_{3,j} \le a_j - 2 - X_{1,j-2} - X_{2,j-1}, \qquad j = 3,4,\ldots,n+2 = m$$

$$\vdots$$

and for the *pth* cut

$$X_{p,1} \le X_{p-1,1} + X_{p-1,2} + \ldots + X_{p-1,m}$$

[3a] or $X_{p,1} \le \sum_{l=1}^{m} X_{p-1,j}$

$$X_{p,2} \le \sum_{l=1}^{m} X_{p-2,j} - X_{p-1,1}$$

$$X_{p,k} \le \sum_{l=1}^{m} X_{p-k,j} - X_{p-k+1,1} - X_{p-k+2,2} - \cdots - X_{p-1,k-1}$$

$\vdots$

[3b] $X_{p,p-1} \le \sum_{j=1}^{m} X_{1,j} - X_{2,1} - X_{3,2} - \cdots X_{p-1,p-2}$

$$X_{p,p} \le a_1 - X_{1,1} - X_{2,2} - \ldots - X_{p-1,p-1}$$

$$X_{p,p+1} \le a_2 - X_{1,2} - X_{2,3} - \cdots X_{p-1,p}$$

$$X_{p,p+2} \le a_3 - X_{1,3} - X_{2,4} - \cdots X_{p-1,p+1}$$

$$X_{p,p+k} \le a_{k+1} - X_{1,k+1} - X_{2,k+2} - \cdots X_{p-1,p+k-1}$$

$\vdots$

[3c] $X_{p,m} \le a_n - X_{1,n} - X_{2,n+1} - \ldots - X_{p-1,m-1}$

Constraints of form 1 to 3 restrict every possible age class that could develop over the planning period.

Goal constraints arise from objectives the manager would like to achieve as closely as possible, but for which some deviation is tolerable. The deviation is permitted and feasibility is ensured through use of the d^-, d^+ variables. Different goals in harvest scheduling are accommodated in these three constraint sets.

Regulating the Stand Constraints

At period $p+1$, each age-class has the same area, *C (=A/N)*

Thus we need

$$[4]\quad \sum_{k=1}^{m} X_{p,k} + d_{1,1,1}^{-} - d_{1,1,1}^{+} = C$$

$$[5]\quad \sum_{k=1}^{m} X_{p-i,k} - \sum_{k=1}^{i} X_{p-i+k,k} + d_{1,1,i+1}^{-} - d_{1,1,i+1}^{+} = C \quad i = 1,2,\ldots,N-1$$

Equation [4] states that the total area cut in the pth period should be C so age-class 1 at period ($p+1$) wll have area C. Equation [5] states that the total are cut in ($p-i$)th period subtracted from the area cut in subsequent periods (from ($p-i+1$) to p) should be C so the area of age-class ($i+1$) at period ($p+1$) will be C.

At period ($p+1$), age-classes from N up to m should not contain any area thus

$$[6]\quad \sum_{k=1}^{m} X_{p-j,k} - \sum_{i=1}^{j} X_{p-j+i,i} + d_{1,2,j-N+1}^{-} - d_{1,2,j-N+1}^{+} = 0$$

where $j = N, N+1,\ldots,m-1$

$$[7]\quad a_j - \sum_{i=1}^{P} X_{i,j+i-1} + d_{1,2,p-N+j}^{-} - d_{1,2,p-N+j}^{+} = 0;\ j = 1,2,\ldots,N+1$$

The terms in the first summation in equation [6], or the first term in equation [7], are those age-classes that will grow to age-class ($j+1$) in equation [6] or ($p+i$) in equation [7]. The terms in the second summation are the same age-class cut in the subsequent periods. Setting term to 0 ensures that no age-classes older than N will remain at period ($p+1$).

Even-flow harvesting Constraints

For the same harvest each cutting cycle of the planning period requires these constraints:

$$[8] \quad \sum_{j=1}^{m} X_{i,j}V_j - \sum_{j=1}^{m} X_{i+1,j}V_j + d_{2,i}^- - d_{2,i}^+ = 0; \; i = 1,2,\ldots,p-1$$

$$[9] \quad \sum_{j=1}^{m} X_{1,j}V_j - \sum_{j=1}^{m} X_{p,j}V_j + d_{2,p}^- - d_{2,p}^+ = 0$$

Equation [8] states that the volume cut in cutting cycle i should be as close as possible to that cut in period $(i+1)$. Equation [9] states that the cut in cutting cycle p should be as close as possible to the cut in cutting cycle 1. Hence the chain relation forms a closed loop that prevents all the deviation from occurring in a single cutting cycle.

Maximizing the Present Net Worth Constraints

Because we want to maximize the present net worth, we can set equal to a large number M. M is usually subjectively determined by decision makers.

$$[10] \quad \sum_{i=1}^{p} \sum_{j=1}^{m} X_{i,j}V_j P_{i,j} + d_3^- = M$$

and try to minimize the underachievement.

All variables of type X and d should be nonnegative.

The objective function depends on the priority: if even-flow harvesting and regulating the stand area are equally important, and both are more important than the income from the harvests in the planning period, the objective function is

$$[11] \text{ minimize } P_1\left[\sum_{i=1}^{N}\left(d_{1,1,i}^- + d_{1,1,i}^+\right) + \sum_{i=1}^{m-N+1} d_{1,2,i}^- + \sum_{i=1}^{P}\left(d_{2,i}^- + d_{2,i}^+\right)\right] + P_2 d_3^-$$

where P_1, P_2 are ordinal weights (ranking goals 1,2,3,...).

Different Cardinal weights (using real numbers to measure priorities of goals relating to one another) can also be introduced to the objective function if w units of deviation in even-flow harvesting and 1 unit of deviation in the final age-class distribution are equally important, then the objective function becomes

$$[12]\ \text{minimize } P_1\left[w\left(\sum_{i=1}^{N}\left(d_{1,1,i}^{-}+d_{1,1,i}^{+}\right)+\sum_{i=1}^{m-N+1}d_{1,2,i}^{-}\right)+\sum_{i=1}^{P}\left(d_{2,i}^{-}+d_{2,i}^{+}\right)\right]+P_2d_3^{-}$$

Useful information Kao and Brodie found here is that good compromised result of three conflicting objectives can be achieved using these objective function formulations. The three conflicting objectives are economic (maximization of present net worth), even-flow harvest, and a normal age-distribution; that is, each age-class must have a similar area, the number of age-classes must equal the rotation age, and no age-classes can be older than the rotation age.

According to theoretical and empirical studies, the preferences of forestry decision makers vary considerably from one decision maker to another (Hyberg and Holthousen 1989). There are some crucial problem when utilizing goal programming which include: specifying the target levels of the goal, determining the weights used in the objective function and making goals measured with different units commensurable. Prior information on the requirements of the decision maker is needed to formulate the problem appropriately. Because there is usually no single overriding management goal in multiple-use forestry, but a set of more or less conflicting objective having certain trade-offs, cardinal weighting is recommended instead of ordinal weighting.

Specifying a set of priori relative weights for goals is often found to be difficult. Kangas, et al. (1992) showed that the Analytic Hierarchy Process (AHP) can be utilized in the estimation these weights. AHP is a mathematical method for analyzing complex deviation problems with multiple criteria. It was originally developed by Saaty (1977).

Basically, the AHP is a general theory of measurement, having both mathematical and psychological features.

For estimating a priori weights of proportional deviations from the target levels, pairwise comparisons between decision criteria (goals) are carried out. When making the comparisons, the question is: which of the two factors has a greater weight in decision making, and how much greater? Verbal comparisons are connected into the numerical form ${}^{w_i}/_{w_j}$, measuring the relative priority of goal i with respect to goal j. Reciprocal matrices of pairwise comparisons are constructed as in the matrix below:

$$A = (a_{ij}) = \begin{bmatrix} 1 & {}^{w_1}/_{w_2} & \cdots & {}^{w_1}/_{w_n} \\ {}^{w_2}/_{w_1} & 1 & \cdots & {}^{w_2}/_{w_n} \\ \cdots & \cdots & \cdots & \cdots \\ {}^{w_n}/_{w_1} & {}^{w_n}/_{w_2} & \cdots & 1 \end{bmatrix}$$

Using the pairwise comparisons as input, the relative weights of elements are computed by using an eigenvalue method (Anderson et al., 11994). The resulting weights represent the decision maker's perception of the relative importance, or preference, of the criteria.

Based on properties of reciprocal matrices, a consistency ratio can be calculated. The consistency ratio measures the coherence of the pairwise comparisons. In human decision making, some inconsistencies can be expected. As a rule of thumb, a consistency ratio value of 10 percent or less is consider acceptable (Saaty, 1980). Otherwise, all or some of the comparisons should be repeated to resolve the inconsistencies of the pairwise comparison. For more details on the AHP theory and the estimation for relative priorities, readers are referred to Saaty(1977, 1980) and Saaty and Kearns (1985).

Disadvantages of GP

Although GP has become very popular lately there have been a lot of issues and criticisms for using this GP as a tool to get optimal solutions to multiobjective forestry problems. Those issues are related to GP with pre-emptive (pre-assigned) target levels, priorities or weights of the objective functions.

Some features and assumptions of GP that are often considered its major weaknesses are infinite trade-offs between goals of different priority levels and the possibility of generating a dominated solution. Infinite tradeoff occurs when one goal is satisfied (or nearly satisfied), but others are not satisfied at all. The disturbing implications of this possibility have been pointed out by Dyer et al. (1979) and Mendoza (1987) and are described briefly by considering a simple case involving two objectives.

Assume that the production frontier (i.e., nondominated solution set) for objectives 1 and 2 is shown in Fig. 3.1. If the goals or target levels for each objective are set at G_1^1 and G_2^1, respectively, and objective 1 is the first priority, GP will generate point A as the optimal solution. This solution implies a maximum of objective 1 and nothing of objective 2. In fact, point A will be the solution generated by GP regardless of the target level set for objective 2. Hence, the only possibility that an optimal solution generated by GP is along the production frontier between point A and point B (a more realistic situation) is when the prespecified levels for objectives 1 and 2 are less than A and D, respectively. For instance, if objective 1 is the first priority and the target level for this objective is specified at G_1^2, GP will generate an optimal solution that is equal to E_0, provided that the specified target level for objective 2 is greater than or equal to G_2^2. If the target level for objective 2 is less than G_2^2, say G_2^3, a dominated solution at E_1 will be generated by GP.

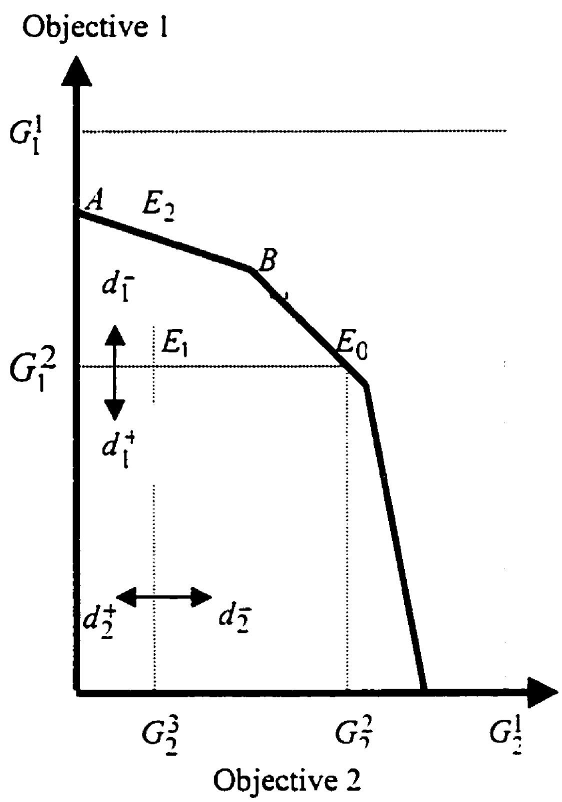


Figure 3.1. Objective space showing the non-dominated solution set of a two-objective problem

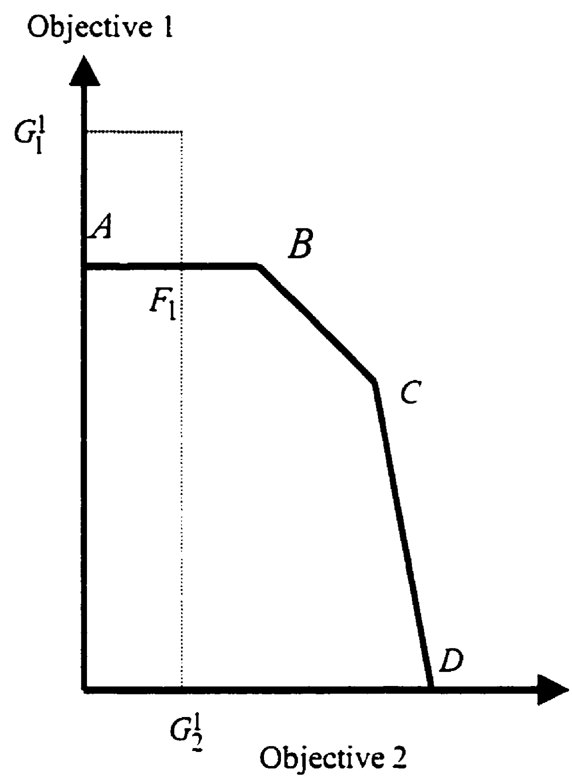


Figure 3.2. Objective space showing the non-dominated solution set of a two-objective problem with alternate optima

Another situation where GP may generate a dominated solution is described in Fig. 3.2. Assume that the target levels for objective 1 and objective 2 are set at G_1^1 and G_2^1, respectively. If objective 1 is the first priority, GP will generate the solution equal to F_1, which gives the maximum possible value for objective 1. However, F_1 is a dominated solution (dominated by B). Note that F_1 is dominated , even though it yielded the highest value of the objective with the highest priority.

Some concerns have also been raised on the specification of weights in Weighted Goal Programming (WGP). Zeleny (1982) and Kluyver (1979) have noted that care should be exercised in assigning weights to the various deviations. This specifically applies to problems with noncommensurable goals and the objective function is formulated as the sum of absolute deviations from targets measured in different units that may not be comparable. As Romero (1985) has pointed out, when the approach is used with goals that are not commensurable, the straightforward objective function aggregating all the deviational variables is meaningless.

The problem of weighted deviations also has some direct implications in the solutions generated, particularly when the goals are expressed in different measurement units (e.g., some goals are in thousand units while others are in decimal units.). In this situation, the goals expressed with the highest magnitude will drive or dictate the solution generated by WGP. Hence, some normalizing or scaling system must be used in calculating weights assigned to each deviation variable. Zeleny (1982) and de Kluyver (1979) have suggested ways of normalizing or calculating scaled weights but Zeleny has also noted that dominated solution may be generated by WGP.

Nondominance in GP solutions

Nondominance is often a desirable characteristic of any solution to a multiobjective problem. Intuitively, the selection of a nondominated solution is appealing because from standpoint of rational decision making, no other solution leads to better attainment of the stated objectives.

Critiques of GP have raised concern about the possibility that a dominated or inefficient solution may be generated by GP. Figures 3.1 and 3.2 describe two situations where this possibility may occur. Zeleny (1982) has also described situations where dominated solutions may be generated under GP or WGP formulations.

Intuitively, it is not a difficult task to detect whether a dominated solution may have been generated. A closer look at Fig. 3.1 indicates that whenever a GP solution yields a zero value in any one or more of the deviation variables it is possible that this solution may be dominated. For instance, at solution E_1 all the deviational variables are zero because the two goals are met. However, E_1 is a dominated solution and it can be improved by increasing the deviational variables d_1^- and d_1^+ from their current zero values. This can be done by increasing the goal levels for both objectives. In these circumstances any point on the line segments E_2B and BE_0 would give a nondominated solution. Along these two lines, either one or both of the deviational variables are greater than zero.

Ignizio (1981) has developed a procedure that can be used to generate a subset of nondominated solutions in GP. His procedure involves a parametric increase of the target level(s) whose deviational variable(s) is(are) equal zero. His technique could be adopted to WGP. Hannan (1980) has also described a mathematical technique to test dominance of GP solutions.

Multi-Attribute Decision Theory

Hyber, T.B. (1987) resolved these conflicting objectives by implementing multi-attribute decision theory. He presented the procedures required to implement multi-attribute decision theory. These procedures are quite different from other methods dealing with resolution of conflicting objectives. The procedures result in a utility function that can accommodate the conflicting objectives. He helped a non-industrial private forest landowner manage his forest to maximize timber harvest revenue while maintaining an attractive site.

Multi-attribute decision theory is a procedure where the manager can structure a problem for analysis, quantify the relative advantages of the available options, and arrive at the preferred solution (Keeney and Raiffa, 1976). The procedure incorporates the expected utility framework developed by von Neumann and Morgenstern (1974) with the decision analysis technique of Raiffa (1968).The general description of multi-attribute decision theory is as follows.

Multi-attribute decision theory presents a decision maker with a series of choices requiring an assessment of preference. This can be best describe in terms of a special kind of lottery. Considering a situation in which a person will receive either a reward r_1 with probability p_1 or a reward r_2 with probability p_2. This is denoted as the lottery L($p_1, r_1; p_2, r_2$). Decision theory is a method of choosing between lotteries. Consider, for example, timber revenue. Suppose that with certainty, lottery L_1 yields \$10,000. Lottery L_2 has a 0.5 probability of yielding \$30,000 and 0.5 probability of yielding \$0. The decision maker must decide whether he prefers $L_1(1,\$10{,}000)$ or $L_2(0.5,\$30{,}000;0.5,\$0)$. He is said to be indifferent between L_1 and L_2 if there is no preference.

In order to rank more than two lotteries, a utility is defined for each possible reward. First identify the most favorable and least favorable rewards that can occur,

r_{best} and r_{worst}, respectively. The utility, u_i, of each of the remaining possible rewards, r_i, is defined to be the probability, q_i, such that the decision maker is indifferent between lottery (1, r_i) and lottery (q_i, r_{best}; $1-q_i, r_{worst}$). The decision maker must then determine a utility function. This estimates the utility of any given reward r, $r_{worst} < r < r_{best}$, with the utility of r_{worst} chosen to be 0 and that of r_{best} to be 1. To clarify the following discussion the decision maker will be assumed to be interested in only two commodities – timber revenue and aesthetics.

Once the decision maker's utility functions for both attributes(i.e. revenue and aesthetics) have been estimated in the above manner, three additional questions are asked. These questions are used to define a multi-attribute utility function. This describes the utility as a function of the levels of both attributes. The first question is "Given the choice between (1) a state with the maximum quantity of aesthetics and the minimum quantity of timber revenue, and (2) a state with the minimum amount of aesthetics and the maximum quantity of timber revenue, which would you choose?" This question asks the decision maker to decide, given an either/or situation, which commodity he or she would choose.

Assume for this example the landowner selects option 2, indicating that he or she would rather have the maximum income over the maximum aesthetics. Given this response, the landowner is asked, "What amount of timber revenue, with no aesthetics, would make you indifferent to a trade for the maximum amount of aesthetics with no timber revenue?" The response to this question provides the dollar amount of landowner would require in order to sell the total esthetic value of his forest. With this value, a functional relationship between the two goods can be developed.

Finally the decision maker is asked, "What probability of success in a lottery involving the maximum amount of both goods versus the minimum amount of both goods would make you indifferent to a choice of participating in the lottery or receiving the maximum income with no aesthetics?" This question allows the estimation of a weighting coefficient for the utility from timber income. It also assesses the individual's aversion to the worst possible outcome. These three questions are used to define constants $K_A,\ K_\$,\ and\ K_{A\$}$, given below.

With the utility function estimated through a series of lottery and an assumption of utility independence, the following equation represents the decision maker's utility.

$$U(a,b) = K_A U_A(a) + K_\$ U_\$(b) + K_{A\$} U_A(a) U_\$(b)$$

where a = level of aesthetics, b = level of revenue

$U(a,b)$ = total utility

$U_A(a)$ = utility from aesthetics at level a

$U_\$(b)$ = utility from timber revenue at level b

$K_A,\ K_\$,\ and\ K_{A\$}$ are scaling parameters

K_A is the probability such that the decision maker is indifferent between lottery $L_1(1, u(a_{best}, b_{worst}))$ and $L_2(K_A, u(a_{best}, b_{best}); 1 - K_A, u(a_{worst}, b_{worst}))$.

$K_\$$ is the probability such that the decision maker is indifferent between lottery $L_1(1, u(a_{worst}, b_{best}))$ and $L_2(K_\$, u(a_{best}, b_{best}); 1 - K_\$, u(a_{worst}, b_{worst}))$.

$K_{A\$} = 1 - K_A - K_\$$

Interactive Multicriteria Programming

Korhonen, P. et al. (1986) introduced a visual interactive approach to multicriteria mathematical programming that shares the advantages of goal programming while providing more effective means of interaction between the decision maker and the computer than traditional goal programming. Interactive use of computer graphics plays a central role in this approach. It gives the decision maker the ability to see the big picture of the problem at hand and enables him to evaluate any part of the Pareto optimal set.

The decision maker's targets are often impossible to achieve simultaneously. However, the decision maker may be interested only in nondominated solutions (the Pareto-optimal set). If there exists a feasible solution for which his targets are attained, there may also exist feasible solutions that are better in all respects. This means that some rules must be established for selecting attainable solutions that bear some relation to the decision maker's targets. Such rules are often defined using an achievement function. The term "achievement function" to refers to all techniques for projecting a given solution onto the Pareto optimal set. There are several ways to specify an achievement function (Ignizio,1983).

Interactive multicriteria programming methods generate a sequence of attainable solutions for the decision maker's evaluation until a satisfactory solution is found. Attainable solutions are often generated using some kind of achievement functions. Visual representation enables the decision maker to evaluate a continuum of solutions simultaneously. Besides the use of visual aids, this approach has two particular desirable features. Firstly, the decision maker is free to examine any part of the Pareto optimal set he pleases at any moment, i.e., he is not confined to evaluating only extreme point solutions, nor is his freedom limited by his earlier behavior during the interactive process. Secondly, the approach needs no specific assumptions concerning the decision maker's underlying utility function (discussed in multi-attribute section above) during the interactive process. The utility function can be even assumed to be changing due to learning and changes of mind during the process

The features of this approach are very useful for solving complex problems in forestry but this approach is only efficient for a certain number of objective functions (at most 10) due to complexity of the approach. However, forestry problems in real life are always complex involving a lot of goals.

Bare et al. (1988) illustrated the potential use of multicriteria (multiobjective) programming in land management planning by solving a demonstrative example using an interactive technique called the STEM method. Among the interactive approaches, the STEM method is applicable to forest land management planning because it can computationally accommodate problems of the size commonly encountered and is easy to understand. Further, it uses the highly efficient simplex method from linear programming which is familiar to most forest planners. This method seeks to identify the best compromise solution by presenting sequential compromise solutions to the decision maker, each reflecting the decision maker's preferences.

3.1.2 Uncertainty

Uncertainty in forest planning is pervasive, entering in the form of a lack of information, imprecision or inaccuracies in estimating model parameters, and inexact or imperfect data. All of these cause uncertainties that must be incorporated in any planning model. Besides imprecision, forest planning is also inherently multiple objective, mainly due to the multiple use nature of forest management. Hence, forest planning models should also address multiple objective concerns in forest management. During the last few decades, mathematical programming models have been used extensively in forest planning, with linear programming and multiobjective linear programming being the most commonly used methods. However, concerns about the use of these models have also been raised. The main criticisms deal with inherently deterministic nature of models, and their use of precise coefficients. In traditional linear programming and multiobjective linear programming models, the coefficient or parameters are assumed to be known with certainty. In many real world forest planning problems, however, it is very unlikely that this assumption is valid. For example, forest managers often have to deal with insufficient or imperfect information due to the inherent complexity of the system (Allen, et. al (1986)). Hence to enhance model utility, it is necessary to be able to incorporate uncertain information (fuzziness) into the model (Mendoza, et al., 1993). In some cases, fuzzy formulations are actually able to provide improvements in all goals (Pikens and Hof, 1989).

In recognition of some of these problems, Mendoza and Sprouse (1989) described a procedure that is particularly suited for complex forest planning problems such as multiple-use forestry. The procedure they proposed is a two stage approach or method. The first stage uses fuzzy models for generating alternative solutions. These models offer some desirable features. First, they allow a more robust generation of widely different alternative solutions. Second, they provide a convenient framework for accommodating a certain amount of fuzziness, uncertainty, vagueness, or ambiguity in the modeling and decision making processes. The second planning stage and its corresponding methodology deal with the evaluation and prioritization of alternatives. One of the goals is to derive a global priority ranking of the alternatives by explicitly considering pairwise comparison of the different alternatives which respect to each criterion. The AHP (Analytic Hierarchy Process – see section 3.1.1) model was adopted to derive a global priority ranking.

The multiple-use forest planning problem considered in their study is a 29,000 acre forest tract located in the Shawnee National Forest. In managing the forest, three goals are considered simultaneously: maximize the economic return from the forest in terms of discounted net revenue, maximize the area suitable for wildlife habitat, and maximize the area suitable for non-motorized semiprimitive recreation. What they have found here is that they can create the payoff table representing the optimal values of each objective as well as the values of other objectives at the optimum values of a given objective. The payoff table provides a convenient framework to describe the maximum model in the context of a bargaining situation where alternatives must be negotiated.

Pikens and Hof (1989) used fuzzy goal programming to solve a forestry problem, harvesting scheduling plan, where the goals are maximization of Net Present Value (NPV) and a stable flow of wood and fiber. The traditional solution to this problem is to maximize NPV subject to a set of constraints which assure that planned harvests will never decline between any successive pair of harvests of the model. Thus, the problem is addressed by selecting one goal as the objective and the other as a crisp constraint. In this study, the problem was reformulated to treat the stable flow of wood and fiber as a "fuzzy" concept rather than as a crisp constraint set. They found that formulating harvest scheduling models with conflicting goals of profit maximization and stable harvests as a fuzzy goal programming problem has the potential to generate solutions which are superior to the traditional crisp formulation for both of the stated goals. That is, fuzzy goal programming gives more profit then crisp goal programming does and the harvest determined by fuzzy goal programming is much more even compared to that determined by crisp goal programming.

Motivated by criticism of using linear programming (LP) for determining timber harvest scheduling where all data are considered to be non-stochastic measurements that are known with certainty, Bare and Mendoza (1992) described how fuzzy mathematical programming can be used to cope with uncertainty in timber harvest scheduling models. They assume that uncertainties can be adequately modeled as fuzzy sets. Thus, timber yield coefficients are treated as deterministic, but strict satisfaction of constraint limit is relaxed and attainment of goal aspiration level is sought but not required. They assume that the fuzziness appears only in the objective function and the timber harvest flow constraints. Other constraints of the LP model are treated as a crisp constraints. The problem addressed represents a situation where the decision maker tolerates some degree of violation in the accomplishment of the timber harvest flow constraints.

A crisp linear programming problem can be written as

maximize $\mathbf{c}^{\mathbf{T}}\mathbf{x}$ (NPV)

subject to

$\mathbf{b}^{\mathbf{T}}\mathbf{x} \le b_1$ (harvest flow constraint)

$\mathbf{D}\mathbf{x} \le \mathbf{b}_2$ (other constraints).

To treat the objective function and harvest flow constraint as fuzzy, we want to find $\mathbf{x}$ such that

$$\mathbf{c}^T\mathbf{x} \gtrsim Z$$
$$\mathbf{B}^T\mathbf{x} \lesssim b_1$$
$$\mathbf{Dx} \leq \mathbf{b}_2$$

where Z is an aspiration (target) level. The crisp model equivalent to this fuzzy model is as follows.

maximize k

subject to

$$kt + \mathbf{c}^T\mathbf{x} \leq Z + t \Rightarrow \mathbf{c}^T\mathbf{x} \leq Z + t(1-k)$$
$$kt + \mathbf{B}^T\mathbf{x} \leq b_1 + t \Rightarrow \mathbf{B}^T\mathbf{x} \leq b_1 + t(1-k)$$
$$\mathbf{Dx} \leq \mathbf{b}_2$$

Bare and Mendoza compared fuzzy linear programming with the linear programming solution of a timber harvest scheduling problem. (The problem was selected from McQuillan (1986), and Pickens et. Al (1990). They found out that by relaxing the harvest flow constraint, the NPV slightly increases and the harvest flow remains reasonably nondeclining. Also by relaxing the objective function, they found out that the NPV can be increased to a certain amount depending on the tolerable deviation and the degree of satisfaction needed. Therefore, they conclude that fuzzy linear programming has potential as a tool to systematically explore alternative solutions.

Mendoza et al. (1993) used a fuzzy Multiple Objective Linear Programming approach to forest planning. They assumed that the decision maker can specify the coefficients in the objective function as intervals $[c_i^l, c_i^u]$ rather than exact values. The paper is organized as follows. First, a single objective function with interval-valued coefficients is formulated as a two-objective function problem. Then, in the presence of multiple objectives, some of which have exact coefficients while others have interval-valued coefficients, the problem is formulated as a multiple objective linear programming problem. Finally, a fuzzy multiple objective linear programming model is formulated with both interval-valued and exact coefficients.

They tested the method with a case study adopted from Johnson et al. (1986). There are four goals, namely, minimizing sediment (solid material, both mineral and organic, that is in suspension and being transported from its site of origin by the force of air, water, gravity or ice), maximizing timber, maximizing forage, and maximizing the net present value (NPV), in which the first three objectives have exact coefficients and the last mentioned objective has interval coefficients, subject to some constraints. The form in the mathematical formulations is:

$$\begin{array}{lll} \text{maximize} & z_1 = \mathbf{c}_1^T \mathbf{x} & \text{NPV} \\ \text{minimize} & z_2 = \mathbf{c}_2^T \mathbf{x} & \text{sediment} \\ \text{maximize} & z_3 = \mathbf{c}_3^T \mathbf{x} & \text{timber} \\ \text{maximize} & z_4 = \mathbf{c}_4^T \mathbf{x} & \text{forage use} \end{array}$$

subject to $\mathbf{Ax} \leq \mathbf{B}$

$$\mathbf{x} \geq \mathbf{0}.$$

Based on the yields, costs, and interest rates, the NPV's coefficients are computed using arithmetic operations in determining interval values. After determining the interval coefficients for PNV, the problem is formulated as:

$$\left.\begin{array}{l} \text{maximize}\, z_1^l = (\mathbf{c}_1^l)^{\mathrm{T}} \mathbf{x} \\ \text{maximize}\, z_1^u = (\mathbf{c}_1^u)^{\mathrm{T}} \mathbf{x} \end{array}\right] \quad \text{PNV}$$

minimize $z_2 = \mathbf{c}_2^T \mathbf{x}$ sediment

maximize $z_3 = \mathbf{c}_3^T \mathbf{x}$ timber

maximize $z_4 = \mathbf{c}_4^T \mathbf{x}$ forage use

subject to $\mathbf{Ax} \le \mathbf{B}$

$\mathbf{x} \ge \mathbf{0}$.

where z_1^l is the lower side of objective 1 (that is, the lower bound of the interval of the coefficients of objective 1), and z_1^u is the upper side of the objective 1.

Among the five objective functions, one (i.e. sediment) is to be minimized. Following Zimmermann (1978), the fuzzy multi-objective linear programming for this case study is formulated as a maximum problem described below:

maximize k

subject to

$$(\mathbf{c}_1^l)^{\mathrm{T}} \mathbf{x} \ge k f_{01}^l + (1-k) f_{11}^l$$

$$(\mathbf{c}_1^u)^{\mathrm{T}} \mathbf{x} \ge k f_{01}^u + (1-k) f_{11}^u$$

$$\mathbf{c}_2^T \mathbf{x} \le k f_{02} + (1-k) f_{12}$$

$$\mathbf{c}_3^T \mathbf{x} \ge k f_{03} + (1-k) f_{13}$$

$$\mathbf{c}_4^T \mathbf{x} \ge k f_{04} + (1-k) f_{14}$$

$$\mathbf{Ax} \le \mathbf{B}$$

$$\mathbf{x} \ge \mathbf{0}.$$

where f_{0i} is the optimal or most desirable value for objective i, and f_{1i} is the least desirable or tolerable value for objective i. To find a solution using this formulation, the f_{0i}'s and f_{1i}'s must be known. These values may be specified by the decision maker. Otherwise, these values can be computationally derived using a payoff table as explained in Mendoza et al. (1993).

3.2. Spatial Consideration

Mathematical analysis is usually included in a forest management project in order to ensure that the varying interests and concerns of the general public and industry are being addressed and taken to account. At the most detailed levels of planning, it is necessary to conduct analysis that incorporates high levels of spatial interaction. This means that management activity in one area impacts the kind of activity that is acceptable in neighboring or adjacent areas. Concerns for the size of open areas, habitat disruption, and fragmentation of a forest are examples of the management considerations where adjacency restrictions have been utilized.

People everywhere are saying it is high time to shift the focus of forest planning from economic production of goods and services to sustainable ecosystem management. In this new paradigm, three classes of forest outputs are recognized: economic commodities, human services, and the health state of the forest ecosystem itself. The weighting on these classes has shifted over the past 20 years from near total preoccupation with producing economic commodities to today's struggles to assign higher priority to the health and sustainability of the total forest ecosystem. It is rare to find an environmentally-related meeting or read a professional journal or even the daily newspaper without seeing this simple message over and over again. It is easy to say forest health should come first but hard to find the balance and also hard to understand the impact of management actions.

Integrated forest management planning is a rudimentary science owing to our poor understanding of the impacts that management actions (such as timber harvest) have on other aspects of the forest. As a result, management objectives, such as preservation of wildlife habitat or biodiversity, are often not explicitly included in the harvest schedule optimization process but instead, are incorporated in the planning process through the series of restrictions to harvest scheduling. Two commonly used restrictions to harvest scheduling are minimum exclusion periods between adjacent clear-cuts (Gross and Dykstra 1988) and the maximum clear-cut size restriction (Hokans 1983). While not directly addressing non-timber concerns, these two restrictions prevent some timber harvest schedules that are known to have poor characteristics.

However, when sustainable ecosystem management becomes paramount, then spatial considerations become crucial. The spatial considerations are usually resolved by using "adjacency constraints" which restrict the time that must elapse before contiguous forested areas of a given maximum size may be harvested. These contiguous areas are usually referred to as harvest blocks.

Mathematical models involving spatial considerations require integer variables resulting in potentially larger and more difficult mathematical formulations than large-scale linear programming forest models such as FORLAN. The more spatial restrictions included, the more integer variables are needed. Usually the integer form of linear programming is used to handle these spatial considerations. Due to the difficulty of solving the integer linear programming problems, some analysts use aggregation heuristics – ways of combining some similar characteristics into one criterion - in order to reduce the number of variables (Meneghin et al., 1988, Tores-Rejo et al., 1990). However, this heuristic method is not always successful especially in large problems.

Some analysts use other mathematical approaches to reduce the total number of necessary constraints in light of the spatial considerations. The most promising one so far is that proposed by Murray and Church (1996). They propose methods that could minimize the number of constraints so that the Mixed Integer Programming (MIP) method can be used successfully. The main purpose of management considerations is to impose adjacency constraints in harvesting forest so that there are no two adjacent units being harvested at the same time. They present a general formulation of an operational forest planning problem which is based in large part on the work of Nelson and Brodie (1990). The objective of this formulation is to maximize net present value while imposing adjacency constraints.

The problem formulation is to maximize $z = \sum_i \sum_t a_{it} x_{it}$ subject to

(1)Limit harvest of a unit to at most once in planning interval $t - p$ *to* $t + p$.

$$\sum_{l=t-p}^{l=t+p} x_{il} \leq 1 \quad \text{for all } i, \text{ and all } t \in [p+1, T-p]$$

(2)Adjacency restrictions to prevent simultaneous harvest of neighboring units.

$$x_{it} + x_{jt} \leq 1 \quad \text{for all } i,t \text{ and for all } j \in N_i$$

(3)Upper and lower bounds on harvest volume in each time period.

(a). $\sum_i v_{it} x_{it} \geq L_t$ for all t (b). $\sum_{i} v_{it} x_{it} \leq U_t$ for all t

(4)Undiscounted revenue bound requirement in each time period.

$$\sum \hat{a}_{it} x_{it} \geq R_t \text{ for all t}$$

(5)Integer requirements.

$$x_{it} = 0,1 \text{ for all } i,t.$$

Where

$i =$ index of harvest units $(i = 1,2,\ldots,I)$

$t =$ index of time periods $(t = 1,2,\ldots,T)$

$x_{it} = \begin{cases} 1 & \text{if unit } i \text{ is harvested in time } t \\ 0 & \text{otherwise} \end{cases}$ (integer decision variables)

$a_{it} =$ discounted revenue generated from harvesting unit i in period t

$\hat{a}_{it} =$ undiscounted revenue generated from harvesting unit i in period t

$U_t =$ upper bound on total volume harvested in period t

$L_t =$ lower bound on total volume harvested in period t

$R_t =$ lower bound on total undiscounted revenue generated in period t

$p =$ harvest exclusion period length

$N_i =$ set of indices of all harvest units adjacent to unit i

v_{it} = volume generated from harvesting unit i in period t

Many researchers have developed adjacency constraints based on a pairwise adjacency approach similar to the above, but this often results in an excessive number of necessary constraints. To deal with this problem, cliques have been used to create better constraint formulations. A clique is defined as a set of mutually adjacent units. That is, each member of a clique is adjacent to the other members of the clique. Figure 2.1 depicts some possible cliques (Murray and Church, 1996).

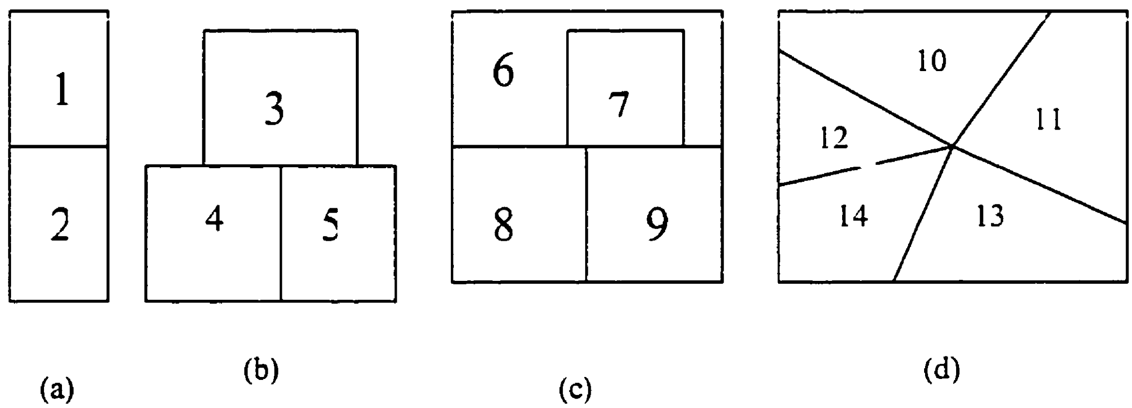


Figure 3.1. Possible adjacency patterns for cliques: (a) pair, (b) triplet, (c) quadruplet, (d) higher ordered

Simultaneous harvesting of two adjacent units (e.g., unit 1 and 2 given in figure 2.1.a) is prevented using the following constraint: $x_1 + x_2 \leq 1$. In the pairwise adjacency approach, constraints of this form are necessary for each pair of adjacent planning units. This has been the traditional approach used to prevent simultaneous activities in adjacent units (Nelson and Brodie, 1990). The unfortunate problem with the pairwise approach is that it typically requires a large number of constraints to impose all the necessary adjacency conditions. Fortunately, it is possible to identify higher ordered cliques which form stronger inequalities than the pairwise clique constraints. Such higher ordered clique constraints can represent the entire set of adjacency restrictions with a significantly fewer total number of needed inequalities, compared to the total number of unique pairwise constraints. For example, in order to prevent adjacent activity in figure 3.1.b, three pairwise constraints can be enforced: $x_3 + x_4 \leq 1$; $x_3 + x_5 \leq 1$; $x_4 + x_5 \leq 1$. Alternatively, adjacency restrictions for the three mutually adjacent units can be enforced by using one inequality of the form: $x_3 + x_4 + x_5 \leq 1$.

The formal definition of a clique is as follows. In order for a set C to be a clique, it is required that all potential pairs of harvest units $i, j \in C$ be adjacent to each other. For each clique C, there is a constraint $\sum_{j \in C} x_j \leq 1$ which imposes the condition that at most one unit in the clique can be harvested.

Clique Constraints and Forest Planning

Type I Approach

Meneghin et. al (1988) develop an approach for identifying quadruplet, triplet, and pairwise cliques in order to maintain all of the required adjacency or pairwise conditions, while also attempting to keep the total number of required constraints to a minimum. They call this set of clique conditions type I constraints. The type I constraint identification approach is very fast and resembles an enumerative approach found in the clique literature (Bron and Kerbosch, 1973). It represents the first practical approach for utilizing clique conditions within an optimization problem.

Type I approach is as follows. First identify all quadruplet cliques such that at most two of the compartments in a potential clique are represented in a previously identified quadruplet clique. After all quadruplets are identified, triplet cliques are identified by selecting those triplets that do not have more than two of their harvest units (compartments) in any previously identified quadruplet or triplet clique. The remainder of unrepresented adjacency conditions are imposed through pairwise cliques.

The Type I approach represents an improvement, because the number of cliques generated is significantly less than the total number of pairwise restrictions so that it provides a substantial benefit to the LP model.

Maximal Cliques

Although the type I approach allows a maximum clique size of four, in many forest planning problems larger cliques exist. For example, it is possible to have common boundary that is defined by a point like the five units given in figure 3.1.d. Also adjacent units need not be defined as those sharing a common edge or point, but may be defined as those units within a specified distance of each other. Because using type I is beneficial, a natural approach to deal with larger cliques would be an extension of this approach (Daust and Nelson, 1994.) To facilitate this, a maximal clique is defined to be the largest subset of mutually adjacent units. For each harvest unit with index i, we can determine the maximal clique that includes it and a subset of N_i (the set of units adjacent to unit i.) Following the identification of cliques larger than quadruplets, the type I approach could then be utilized to identify the remaining necessary clique constraints to form a complete adjacency constraint set.

The process described above identifies cliques that do not have a specified amount of overlap with selected cliques. Murray and Church (1996) propose an alternative selection approach. Rather than using a rule to narrow the selection/generation of the prospective cliques, they are interested in all alternative cliques and how they compare to other previously identified cliques. As such, a prospective clique may be redundant or dominated by one of the already identified cliques.

This can be demonstrated using the following example of inequalities:

(a) $x_i + x_j + x_l \leq 1$, (b) $x_i + x_j + x_l + x_t \leq 1$.

Constraint (a) is a triplet clique and (b) is a quadruplet clique condition. Inequality (a) is said to be dominated by inequality (b) since (b) logically implies (a). Thus inequality (b) need only be used. This rule can be used to modify the type I approach by selecting only nondominated cliques. The nondominated approach for clique selection is to begin by selecting nondominated maximal cliques that are larger than quadruplet, next, non-dominated quadruplets are selected which impose adjacency conditions that are not yet represented (nondominated by selected maximal cliques), then triplets that are not dominated are identified, finally, nondominated pairwise cliques are used to impose the remaining adjacency conditions.

Methods that provide a reduction in the number of necessary constraints typically produce constraint forms that are of poor structure. Poor constraint structure generally results in the inability to solve for the optimal integer solution. Murray and Church (1996) propose an approach which uses a minimal subset of clique constraints approach to develop a process that can identify a minimal set of adjacency constraints, without sacrificing the constraint structure that aids in the solution process.

It is important to recognize that there could be a certain amount of representational redundancy in a given selected clique set. Even though each clique is nondominated, there still exists the likelihood that some clique may not be necessary in order to maintain all of the pairwise adjacency restrictions. Example, consider the following clique constraints;

$$x_1 + x_2 + x_3 + x_4 \leq 1$$
$$x_3 + x_4 + x_5 \leq 1$$

Neither constraint dominates the other, but the adjacency relationship of unit 3 and 4 is imposed in both constraints and can be considered redundantly imposed to a certain extent. That is, other clique constraints may exist that impose the restrictions between units 3 and 5 and 4 and 5, as an example, so that the second constraint given above could be eliminated.

A set-covering formulation can be utilized to express the optimization problem of identifying a minimal subset of nondominated cliques.

Let k = index of potential clique constraints

i = index of pairwise adjacency conditions

S_i = the set of clique constraint k that imposes pairwise condition i.

Choose the decision variables:

$$y_k = \begin{cases} 1 \text{ if clique constraint } k \text{ is used to impose adjacency conditions} \\ 0 \text{ otherwise} \end{cases}$$

The set covering problem for clique constraint selection:

$$Minimize\ z = \sum_k y_k$$

subject to:

(1) ensure that all pairwise adjacency conditions are represented in the selected set of clique constraints:

$$\sum_{k \in S_i} y_k \geq 1 \qquad \text{for all } i$$

(2) integer requirements $y_k = 0$ or 1 for all k

The objective is to minimize the number of clique constraints used. Constraint (1) requires that each pairwise adjacency condition is imposed at least once in the selected set of clique constraints. Constraint (2) imposes integer restrictions on the decision variables. Given a clique set, for example a nondominated clique set, we can identify a minimal subset of these cliques which maintain all pairwise restrictions by using this set-covering approach.

Nonlinear techniques

Solving integer linear programming problems using the simplex algorithm is quite difficult. Therefore, some analysts use other algorithms to solve these problems. Clements and Jamnick (1990) use Monte Carlo integer programming (MCIP) to generate short-term (25-year), spatially feasible timber harvest plans for a New Brunswick Crown license. A typical MCIP algorithm begins by generating random solutions to a mixed-integer programming problem. These solutions are tested against a set of spatial and temporal constraints, and solutions meeting all of the constraints are designated feasible. Each feasible solution is evaluated relative to an objective function. After a large number of feasible solutions have been identified, the solutions best satisfying the objective function are selected for further analysis. This procedure does not guarantee finding the optimal solution, but it does quickly generate several near-optimal ones. It also has the advantage of being able to handle large, complex problems that are too large or complex to solve, in reasonable amounts of time.

Jammick, and Walter (1991) used MCIP to determine timber harvest volumes in the presence of adjacency constraints. Their study presumed that a particular harvest blocking pattern has been established. Given this pattern, the objectives is to determine a near optimal integer solution. They present an analysis of twelve alternative harvest blocking patterns for a small New Brunswick forest. For this particular forest, the difference between blocking patterns are relatively small and forest managers have considerable flexibility in choosing a blocking pattern which best meets operational criteria without sacrificing timber harvest volume.

Lockwood and Moore (1993) used the nonlinear optimization method called simulated annealing (SA) to generate harvest scheduling solutions of a model with many spatial constraints, especially the requirement to comply with exclusion periods and maximum clear-cut size restrictions. SA is a stochastic optimization technique, which has been used successfully to solve large combinatorial optimization problems. An attractive feature of the SA procedure is that it allows nonlinear and discontinuous constraints and objectives in an optimization framework.

Linear programming has become one of the most common analysis techniques in renewable natural resource management and planning. The intrinsic linearity of the approach is clearly a limitation, but the exact nature of this limitation is rather subtle. Linear programming can be used to piecewise approximate highly nonlinear relationship between inputs and outputs. For example, if different management prescriptions are included that involve different levels of intensity of some input's utilization, the different A-matrix coefficients under these different management prescriptions can reflect highly nonlinear response to changing input intensity. This linearity assumption of the LP do cause some problems, however. The most important of these problems is that of accounting for the impact of the spatial configuration of a management action on outputs of interest. If management prescriptions are based on a per acre basis, the LP determines the number of acres to which each management prescription applies. The problem is, the (nonlinear) response to different sizes and shapes of the management action is lost in a fixed per-acre production coefficient. Considering these concerns, many authors have tried to resolve them.

Clement et al. (1990) and Nelson and Finn (1991) have defined the management variables in terms of timber stand that are treated discretely and are preserved as discrete units in solution. The spatial considerations are then typically viewed in terms of nonadjacency constraints over time constraints that limit the size of contiguous cutover areas at any given time. Considerable progress has been made in solving the problem viewed this way, but the approach is limited by accepting and preserving the initial stand definitions. Also, this approach avoids "spatial anomalies" but it does not account for the nonlinear response of many forest outputs (such as wildlife and fish, water, esthetics, etc.) to different sizes, shapes, and arrangement actions. It would be difficult to argue that it finds "spatial optimal" solutions for all outputs of concern (as opposed to just timber). Thinking that this approach does not give "spatial optimal" solution, Hof and Joyce have proposed several nonlinear approaches to land allocation modeling that optimize spatial layout, per se, for a single time period and that have the property that the number of choice variables increases linearly with the level of spatial resolution. Their paper focuses mainly on wildlife habitat as the primary non-timber spatial concern. Wildlife habitat requirements include factors related to food, shelter, and shelter or cover needed. They address a subset of these needs that are related to spatial configuration and assume that the following criteria are important in the spatial layout of wildlife habitat and vary according to the species considered: the amount of edge, the juxtaposition of different habitat types for cover versus feeding needs, the distance between favorable habitats, and the minimum size of a patch of habitat. They propose two nonlinear models: one that accounts for spatial patterns with a cellular grid, and an alternative that uses geometric shapes.

What they concluded from their study is that this nonlinear problem is not easy to solve even for the simple case considered and is not realistic. Consequently, this approach is not recommended as a method to optimize spatial layout but they propose instead a different way of looking at the problem of spatially specified forest management.

3.3. Road construction

It takes more than two centuries for a forest to recover naturally from the damage caused by harvesting and revert to a useful softwood forest again (Minamikata, 1984). In contrast the regeneration periods if planned artificial methods are used can be less than seventy –five years. However, in comparison with natural regeneration, artificial regeneration requires lots of labor for planting, weeding, pruning, and thinning. Generally speaking, the greater the labor required in stand management and the higher the labor cost, the more economically significant will be the road network in the forest. In these cases, the forest agency therefore tends to use the roads in the forest area as much as possible so as to give minimum cost of operation. On the other hand, the influence of forest road construction on the ecosystem or natural environment of the forest may be very important. For example, opening up forest roads occasionally causes landslides, sometimes on a large scale.

To accommodate the economic affects and various impacts of road construction, Minamikata (1984) proposed a road planning system based on a mesh analysis method.

Using this method, the forest road is extended section by section. Extensions consist of road along the sides of the grid or along a diagonal, taking into account the direction along which the route has already been laid, and extension directions with the highest economic effect. The procedure is then repeated from the new starting point.

Road construction in forest management is also very important because by constructing a road network in the forest properly, we can minimize the cost of harvesting or other activities and also minimize the influence of forest road construction on the forest environment. Carson et al. (1978) showed that a transportation system of forest roads may be described as a network, a collection of interconnected segments or links. Each link describes a unique path between two adjacent nodes. A node may be as departure or destination of some path through network such a landing or mill. Nodes may also be road intersections, viewpoints, scaling stations, and bridges. The unit of measure selected to judge a path's length can be anything, such as hauling cost, distance or time, construction cost, maintenance cost, or even a measure of scenic or esthetic value along the link. Carson et al. produced a program that can find the shortest path through a network from a specified point of departure utilizing the "Moore algorithm" . However this program is limited only to 60 nodes and 255 links with no more than 8 links meeting at a single node.

Planning forest road networks in steep mountain terrain is very difficult to achieve using analytic methods. Kouchi (1966) proposed a forest road planning technique using topological considerations in conjunction with analytic methods.

Typical objectives of planning a forest road network are to minimize the length of the roads in the network or to minimize the cost. The objective considered by Kouchi is planning a road network with shortest possible length. The method of Kouchi starts by picking out all places where the road must pass through without actually drawing any road lines. If these so-called passing points can be picked successfully, then the job is to complete the network by connecting them - aiming at the shortest total road length at minimum total cost.

A typical system of logging and transportation is to gather the cut trees with or without branches in some open area – called a landing- where log making and sorting take place. The timber is transported directly from this landing to markets or factories. This method is agreed to be ideal in many logging areas from the stand point of cost management. Using the Ashu Forest in Japan as a study case. Kouchi decided that the ideal system is to have one step to the landing and one step to market, and that the total road length should be minimized.

To realize the idea above, the collection areas selected must have enough width and flatness for log making operation and must not result in delays in the delivery of the timbers. Therefore, at first, a large number of such places were chosen all over the forest on a map of 1/10000 scale. Then the criterion applied was that the gradient in a circle of 1 cm diameter on the map was less than 3/10. The number of the places selected was about 100.

The next job was to select a minimal number of these locations sufficient to carry out successfully the one step logging to landing stage. For this purpose, we can utilize the concept of minimum external stable set from graph theory. It is assumed that some logging level optimal for Ashu Forest has been decided and that some passing points have already been decided. Here, "logging level" is a phrase Kouchi used to express what kind of logging method is used and to what extent, for example, tractor skidding within 30 minutes cycle time, cable way of one span within 1000m, and so on. Then if any timber at any location in the forest can be gathered to some of the selected passing points, using the decided logging level he call such set of collection points an "external stable set". An external stable set with a minimum number of points is called a "minimum external stable set ". In the study of Kouchi, a cable way of one span within 1000m was taken uniformly for all points, as the logging level. And he drew the sphere of logging from each preliminary collection point on the same map.

The algorithm for extracting a minimum external stable set out of the above preliminary selected points is illustrated by the following.

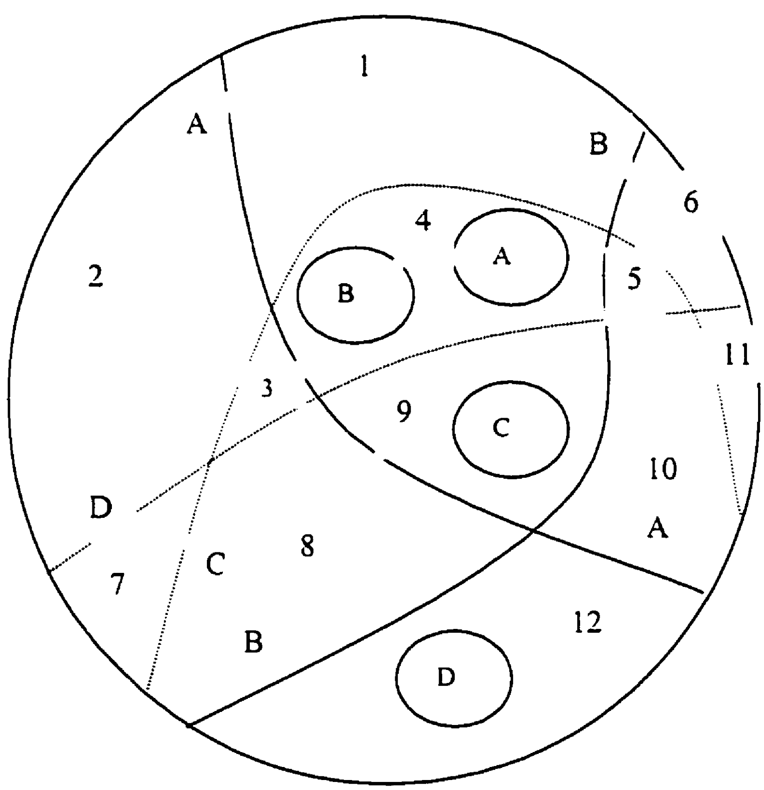


Fig. 3.3. Logging area of each landing

Assume A, B, C, D on figure 2.3 are the points initially selected. Lines on the map representing the sphere of logging from each point divide the forest into many small domains, In figure 1 the total area is divided into 12 domains. Timber in the domain 1 can be gathered to A or B, and so it is represented as (A+B). The domain 2 belongs only to B, so it is denoted by B, on so on. Any domain needs to belong to some of the points for our purpose, so we represent that as ().

1 2 3 4 5 6 7 8 9 10 11 12

(A+B)B(B+C)(A+B+C)(A+C)A(B+D)(B+C+D)(A+B+C+D)(A+C+D)(A+D)(C+D) ..(1)

Logical meaning of this is (A+B) AND B AND (B+C) AND ...... AND (C+D).

In the next step shrink the number of points by using the law of absorption which is as follows; AX $\rightarrow$ A if A is entirely included in X, for example, A(A+B+C) $\rightarrow$ A. This reduced (1) to (2),

AB(C+D) (2)

In the next step expand (2) to get (3) which is the solution

ABC+ABD (3)

For this to be true, either ABC has to be true or ABD. Hence, the minimal number is 3 and we have two choices ABC or ABD, and the choice might be the one which could be connected by the shorter road line.

Decision of the forest road network

There will be a lot of discussion about what kind of network should be chosen, but here our only aim is that the total road length should be minimal. The algorithm for connecting all the points by the roads with shortest total length makes use of the concept of "tree" from the graph theory. It is assumed that connecting roads never cross each other except at the passing points and the algorithm is as follows.

a) Choose the shortest one out of the road-lines connecting any two of the points

b) Delete the road-lines already chosen and the ones that will make cycles with the formers. And after that, choose the shortest just like in a).

c) Iterate b)'s step until all the points will be connected by chosen lines.

3.4. Forest Products

Foresters and other executives engaged in the forest products industries are constantly faced with the problem of how to allocate their resources in a manner that will maximize some utility, usually profit. In making these decisions, they usually think in terms of improved manufacturing methods. One of the most difficult and costly components in the process of converting the forest crop into useful products is the production of logs. Once the tree is on the ground, a bucking crew begins to make some fundamental decisions that are as important as any that are made in the total process of tree conservation. The loggers, with their axes and chainsaws, determine what portions of the tree to allocate to lumber production, veneer, and pulp. Further, they influence greatly the length of lumber and veneer or plywood to be produced, and ultimately profits. The log makers are constantly looking for better methods to assist them in their effort to maximize returns.

Pnevmaticos and Mann (1972) use dynamic programming to produce a pattern of tree bucking (sawing felled trees into shorter lengths). The problem is formally defined as follows: Given a stem (fig. 2.4) of length L, larger diameter D, and smaller diameter d, it is desired to cut it into logs in such a way so that the total return from the stem is maximized. The objective is to maximize the returns from the tree by finding the number of logs to be cut, their length and diameter, and the location of the cuts. The constraints are: the total length of the logs must be equal to or less than the initial stem length, the diameters of any log must be within the limits of the diameters of the remaining stem, and both log length and diameter must be within the limits specified by management.

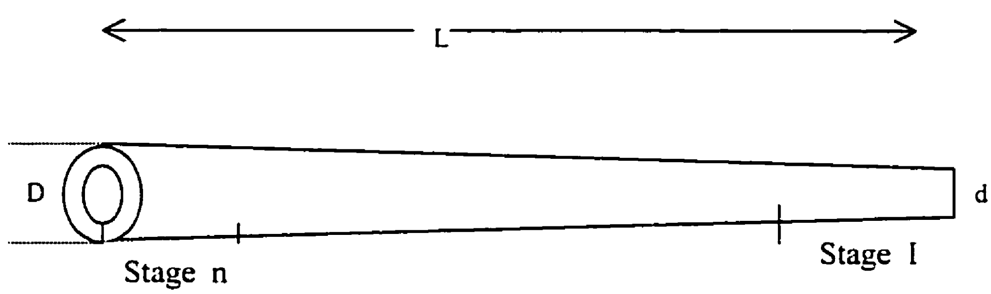


Figure 2.4. Relation of dynamic programming terminology to tree bucking

For this type of decision process, dynamic programming (DP) is an appropriate technique for finding an optimal cutting policy. Some necessary parameters are as follows:

L = stem length in feet

D = large diameter of stem inside bark in feet

d = small diameter of stem inside bark in feet

k = minimum length of log, in feet, acceptable to management

m = maximum length of log, in feet, acceptable to management

Costs and revenue in this operation are $c(r,u)$, the cost of making a cut of diameter u for a log of length r, and $v_g(r,s,t)$, the value of a log of length r, large diameter s, small diameter t, and grade g.

Let $p_g(r,s,t)$ be the probability that a log of length r, large diameter s, and small diameter t, is of grade g. Define $f_i(L,D,d)$ as the maximum expected value of a stem of length L, large diameter D, and small diameter d with i stages remaining in the decision process. We assume $f_o(L,D,d) = 0$ for all values of L, D, and d. Then, assuming that cutting begins at the large end of the stem and proceeds toward the end with the smaller diameter, we have

$$f_n(L,D,d) = \max_j \left\{ \begin{array}{l} \sum_g v_g\left[jk, D, D - jk\left(\frac{D-d}{L}\right), d\right] p_g\left[jk, d + jk\left(\frac{D-d}{L}\right), d\right] \\ - c\left[jk, d + jk\left(\frac{D-d}{L}\right)\right] + f_{n-1}\left[L - jk, D - jk\left(\frac{D-d}{L}\right), d\right] \end{array} \right\}$$

where j is the decision variable indicating the number of minimum log lengths to remove from the stem with a single cut. The decision variable j is constrained as follows: $1 \le j \le \min\left(n, \left[\frac{m}{k}\right]\right)$ where $[m/k]$ indicates the greatest integer contained in m/k. This constraint on j is imposed to insure that at least a single log of minimum length is removed from the stem, but no more is removed than the maximum allowable log length.

Bucking decisions should include consideration of stem taper, stem length, and the log quality , as well as capability and capacity of manufacturing machinery and market demands for various end-use products. If the stem can be bucked so that it meets the above considerations, optimal revenue can be produced. An approach that takes into account simultaneously the limitations of the forest resource in terms of quality and quantities and the market requirement for end-use products seems superior. Eng, (1986) tries to accommodate those considerations in order to prescribe appropriate bucking patterns.

Mathematical Model of Eng,

The DP sub-problem, bucking the stem, is used to generate activities for the Linear Programming (LP), so that the bucking strategies can reflect properly the opportunity cost resulting from critical constraints on demands and (or) resources.

Assume that the forest resource has been classified into J tree classes, each defined by size and quality of the stems found in one or more stands. Let X_{ij} denote the number of true stems of class j bucked by pattern i.

Define r_{ij} as the return from bucking a stem of class j by pattern i and a_{ijk} as the associated volume of log type k, $k=1,2,\ldots,K$.

Let S_j be the number of stems of class j available in the wood resource, and b_k the required demand in the given time period for log type k. The objective is to determine the optimal X_{ij} value so as to

maximize $\sum_i \sum_j r_{ij} X_{ij}$ subject to the constraints [1]

$$\sum \sum a_{ijk} X_{ij} \ (\leq, =, \geq) \ b_k, \ k = 1,2,\ldots,K \qquad [2]$$

And the tree supply constraints

$$\sum X_{ij} \leq S_j, \ j = 1,2,\ldots,J, \qquad [3]$$

$$X_{ij} \geq 0 \qquad [4]$$

Note that X_{ij} is an integer but X_{ij} values are large for most practical application so that the integer constraint can be dropped and the problem reverts to a regular LP.

Note that each column or activity in this LP problem represents a possible bucking pattern for a given tree class. To determine the return r_{ij} and parameters a_{ijk} , $k = 1,2,...,K$ column ij, we use a streamlined dynamic programming formulation for finding the optimal bucking pattern. Consider a stem of class j. For large scale practical applications, the dimensions of a stem are approximated by a taper equation as a function of tree species, forest locality, age, etc. , and either height of stem or its diameter at breast height over bark(dbhob). Using height as the index for site, consider a section of a stem of length L, measured from the base of the stem, with $0 \leq L \leq L_{\max j}$, where $L_{\max j}$ is the total usable length of the stem. We wish to buck that section optimally into shorter logs so as to maximize a return function that reflects the marketable values of those logs.

Let y_k denote the length of a short log of type k cut a distance $L - y_k$ from the base of the stem and $r(y_k, L)$ represent its associated end-use product value then the following recursive relation results.

$$f(L) = \underset{\substack{k \\ y_k \in Y(L)}}{\text{maximum}} \left(r(y_k, L) + f(L - y_k) \right) \qquad [5]$$

for $0 \le L \le L_{\max j}$ and with *f(0)=0* and where *Y(L)* is the set of feasible short logs at L for all K end-use products. This set depends on factors such as minimum and maximum length, minimum small-end diameter, and permissible defects, as dictated by marketing considerations. The optimization is over all log types K and all short log lengths y_k feasible at L. The value of a short log from a stem of tree class j can be made depending on both location and length. For instance, sawlogs cut from large diameter sections of the stem have a higher value per unit volume than those coming from smaller diameter sections. Similarly , the unit volume value of long poles is higher than that of short poles of the same large end diameter.

Making $r(y_k, L)$ location dependent also allows the formulation to accommodate any stem defects of tree class *j*. This is achieved by repricing any potentially defective short logs. Assume a stem contains a defect from location $\underline{l}_k$ to $\bar{l}_k$, which render that portion unsuitable for inclusion in short log of type *k*. Then any short log of type *k* cut from the stem from location $L \; to \; L - y_k$, where position $[\underline{l}_k, l_k]$ is contained in $[L, L - y_k]$, is assigned a negative $r(y_k, L)$ value. As a result, the optimal DP solution will never include such short logs.

The output of the DP bucking problem for each tree class *j* consists of the vector of log type volumes. $\{a_{ij1}, a_{ij2}, \ldots, a_{ijk}\}$, and the associated return per stem $r_{ij} = F(L_{\max j})$.

Solution Method

The output of the DP recursion becomes the input to the LP problem. The Dantzig-Wolfe decomposition algorithm allows us to approach the global optimum successively. This algorithm can be viewed as a notional dialogue between a forest resource planner and buckers. The planner's job is to coordinate bucking patterns applied to the forest resource so as to meet end-use product demands, while maximizing the total value of the forest resource in terms of log type production. The buckers are responsible for bucking all stems in a given tree class. Each party is viewed as a profit-maximizing unit. Starting with an initial set of bucking patterns, the planner finds a provisional operating plan in the form of a subset of bucking patterns to match demand and supply constraints. The planner then assesses the internal penalties and premiums associated with the demand and supply constraints for that solution. The use of any stems from a tree class in tight supply is penalized to discourage their use. Similarly, any log types with upper demand limits fully met are penalized to discourage their production, while those with lower demand limits just met will be given a premium to encourage their production. Given the original log type prices and these penalties and premiums, the buckers, in turn, attempt to generate new bucking patterns that maximize their return. They do so without regard for the feasibility of the planner's overall allocation problem. These new bucking patterns are reported back to the planner, who adds them to the previous set of alternative bucking patterns to find a new operating plan and a new set of internal penalties and premiums. This iterative process continues until the buckers are unable to generate new bucking patterns that are profitable at the prevailing set of penalties and premiums. At this point the process terminates. The globally optimal solution has been found.

We now outline how the above process translates into an algorithm. The algorithm is initiated by finding at least one bucking pattern for each tree class using recursion [5]. It would also be useful, though not necessary, to generate additional patterns for some tree classes. This will facilitate finding an initial feasible solution to the LP on the first iteration. Assume now that we have just solved the LP at iteration $n \geq 1$.

Denote by π_k the shadow price of the demand constraint for log type k. This is the negative of the premium or penalty per unit volume referred to above. The $r(y_k, \mathrm{L})$ values to be used in [5] are now adjusted to reflect the shadow prices. Say, for simplicity p_k, is the unit market value of log type k. Then, the new adjusted unit price becomes $p_k - \pi_k$. Let $\hat{\pi}$ be the shadow price for the supply constraint of tree class j. A new bucking pattern for tree class j is profitable if

$$F(LMAX_j) \geq \hat{\pi}_j \qquad [6]$$

Hence, at each iteration, we apply recursion [5] to find a new bucking pattern for each tree class using the newly adjusted log type prices. Those that satisfy condition [6] are added as new activities to the LP. The LP is then resolved, resulting in new shadow prices and the process is repeated. It stops when no tree class can generate a new bucking pattern that satisfies [6]. At this point, as shown in End.(1982), the optimal solution has been found.

Chapter 4

Genetic Algorithms

4.1. Introduction

Any abstract task to be accomplished can be thought of as solving a problem, which, in turn, can be perceived as a search through a space of potential solutions. Since we want the best solution, we can view this task as an optimization process. For continuous and simple solution spaces, classical exhaustive methods usually suffice; for more complicated spaces special techniques must be employed. Genetic Algorithms (GAs) are among these special techniques. They are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian natural selection or survival of the fittest.

In the biological world, the process of natural selection is thought to be a major control over evolution. Organisms most suited for their environment tend to live long enough to reproduce and are more successful in their reproduction, whereas less-suited organisms often die before producing young or produce fewer and/or weaker young. A GA is an artificial life simulation method that mimics the process of evolution by creating an artificial world, populated with pseudo-organisms governed by some measures of survival and reproduction. The given measures of survival and reproductive success can be chosen to ensure that this very crude form of evolution encourages the pseudo-organisms to evolve to a specific goal.

GAs have been successfully applied to optimization problems such as wire routing, scheduling, adaptive control, game playing, cognitive modeling, transportation problems, traveling salesman problems, and optimal control problems. However, De

Jong (1985) warned against perceiving GAs as a completely reliable optimization tool: "because of the historical focus and emphasis on function optimization applications, it is easy to fall into the trap of perceiving GAs themselves as optimization algorithms and then being surprised and/or disappointed when they fail to find an 'obvious' optimum in a particular search space." He suggests that a way to avoid this perceptual trap is to think of GAs as a simulation of a natural process. As such they embody the goals and purposes of that natural process. On the other hand, optimization is a major field of GA's applicability.

4.2. General Structure of Genetic Algorithms

GAs were formally introduced in the United States in the 1970s by John Holland at the University of Michigan (Holland, 1975). The continuing price/performance improvements of computational systems have made them attractive for some types of optimization. In particular, genetic algorithms work very well on mixed (continuous *and* discrete), combinatorial problems. They are less susceptible to getting 'stuck' at local optima than gradient search methods. But they tend to be computationally expensive.

GAs belong to the class of stochastic search methods (other stochastic search methods include simulated annealing, and some forms of branch and bound (Goldberg, 1989)). Whereas most stochastic search methods operate on a single solution to the problem at hand, genetic algorithms operate on a population of solutions. This population evolves, from generation to generation, into a population of better solutions to the problem.

The general structure of a typical GA is as follows. First, solutions to a problem must be encoded in a structure that can be stored in the computer. Each encoded solution is called a chromosome. An initial population of chromosomes is created. These initial chromosomes can be chosen at random or by using information that is at hand. Genetic operators, called recombination (or crossover) and mutation are applied to the individuals in the population to generate new individuals. Some selection criteria is used to choose fitter individuals for the next generation. Fitness is usually determined by an objective function value.

Encoding of solutions (chromosomes) can be done in many ways. Traditionally, GAs use strings of bits to represent solutions. Holland worked primarily with strings of bits, but arrays, trees, lists, or any other object can be used. The key thing to keep in mind is that the genetic machinery will manipulate a finite representation of solutions, not the solutions themselves. Of course, mutation, crossover, and selection will be defined differently depending on the representation chosen.

Selection is some means or procedure for discriminating good solutions from bad solutions. This can be as simple as having a human intuitively choose better solutions over worse solutions, or it can be an elaborate computer simulation or model that helps determine what "good" means. But the idea is that something must determine a solution's relative fitness. This will be used by the genetic algorithm to guide the evolution of future generations. Simply stated, selection allocates a greater likelihood of survival to better individuals – this is the survival-of-the-fittest mechanism we impose on our solution.

Selection can be used in two different ways. On the one hand, it can determine how individuals are chosen for mating by recombination and mutation. On the other, it can be used to choose, among the parents and children, those individuals that will survive into the next generation. Either way, if we use a selection method that picks only the best individual, then the population will quickly converge to that individual. So the selector should be biased toward better individuals, but should also pick some that aren't quite as good (but hopefully have some good genetic material in them).

Some of the more common methods of selection include the following. In roulette wheel selection, the likelihood of picking an individual is proportional to the individual's fitness. Thus a new population is selected with respect to the probability distribution based on fitness value. See section 4.3 for a detailed example. In a tournament selection a number of individuals are picked using roulette wheel selection, then the best of these are chosen for mating. In rank selection the best individuals are picked every time. Recombination (crossover) is a genetic operator that combines bits and pieces of parental solutions to form, new, possibly better offspring. There are many ways of accomplishing this, but the primary idea to keep in mind is that the offspring under recombination will not be identical to any particular parent and will instead combine parental traits in a novel manner. Typically crossover is defined so that two individuals (the parents) combine to produce two more individuals (the children). But you can define asexual crossover or single-child crossover as well. The primary purpose of the crossover operator is to get genetic material from the previous generation to the subsequent generation. By itself, recombination is not all that interesting an operator, because a population of individuals processed under repeated recombination alone will undergo what amounts to a shuffling of extant traits.

The mutation operator introduces a certain amount of randomness to the search. It can help the search find solutions that crossover alone might not encounter. Mutation acts by simply modifying a single individual. There are many variations of mutation, but the main idea is that the offspring be identical to the parental individual except that one or more changes is made to an individual's trait or traits by the mutation operator. By itself mutation represents a random walk in the neighborhood of a particular solution. If done repeatedly over a population of individuals, we might expect the resulting population to be indistinguishable from one created at random.

Two of the most common genetic algorithm implementations are 'simple' and 'steady state'. The simple genetic algorithm is a *generational* algorithm in which the entire population is replaced at each generation. In the steady state genetic algorithm, only a few individuals are replaced at each 'generation'. This type of replacement is often referred to as overlapping populations.

In recent years, genetic algorithms have taken many forms, and in some cases bear little resemblance to Holland's original formulation. Researchers have experimented with different types of representations, different crossover and mutation operators, and different approaches to reproduction and selection. However, all these methods have a family resemblance in that they take some inspiration from biological evolution and from Holland's original GA.

4.3. Example

In this section, one implementation of a GA (Michalewicz, 1992) is discussed by way of a simple example. It is a steady state GA, using strings of bits to encode real numbers, and a roulette wheel selection method.

Suppose the optimum is required of a simple function of one variable, defined as $f(x) = x\sin(10\pi x) + 1$. The problem is to find x from the range [-1, 2] which maximizes the function f, i.e., to find x_0 such that $f(x_0) \geq f(x)$, for all $x \in [-1,2]$.

The approximate solution can be found analytically, for comparison purposes, as follows.

$$f'(x) = \sin(10\pi x) + 10\pi x\cos(10\pi x) = 0 \text{ when } \tan(10\pi .x) = -10\pi x.$$

This has solutions of the form.

$$x_i = \frac{2i-1}{20} + \varepsilon_i, \text{for } i = 1,2,\ldots.$$

$$x_0 = 0$$

$$x_i = \frac{2i+1}{20} - \varepsilon_i, \text{for } i = -1,-2,\ldots,$$

where terms $\varepsilon_i s$ represent decreasing sequences of real numbers (for $i=1,2,\ldots$ and $i = -1,-2,\ldots$) approaching zero.

Note also that the function f reaches its local maxima for x_i if i is an odd integer, and its local minima for x_i if i is an even integer.

Since the domain of the problem is $x \in [-1,2]$, the function reaches its maximum for $x_{19} = \frac{37}{20} + \varepsilon_{19} = 1.85 + \varepsilon_{19}$ where $f(x_{19})$ is slightly larger than $f(1.85) = 1.85.\sin(18\pi + \frac{\pi}{2}) + 1.0 = 2.85$

Now a genetic algorithm is used to solve the above problem, i.e., to maximize the function f.

Representation

We use a binary vector as a chromosome to represent real values of the variable x. The length of the vector depends on the required precision. Suppose that a solution is required to be accurate to six decimal places. Since the domain of the variable x has length 3, the precision requirement implies that the range [-1,2] should be divided into at least $(3x10^6) = 3000000$ equal size ranges. The required number of bits (denoted by m) must satisfy m = smallest integer larger than $log_2(3000000)$,

$$2^{m-1} < 3000000 < 2^m - 1$$

$$2097152 = 2^{21} < 3000000 \leq 2^{22} = 4194304.$$

This means that 22 bits are required in the binary vector (chromosome):

Initial Population

The initialization process is to create a population of chromosomes, where each chromosome is a binary vector of 22 randomly chosen bits.

Evaluation function

Since the evaluation function $f(x)$ is real valued, the binary vector v must first be decoded. The mapping from a binary string $\langle b_{21}b_{20}...b_3b_2b_1b_0\rangle$ into a real number x from the range [-1,2] is straightforward and is completed in two steps:

- convert the binary string $\langle b_{21}b_{20}...b_3b_2b_1b_0\rangle$ from the base 2 to base 10:

$$(\langle b_{21}b_{20}...b_3b_2b_1b_0\rangle)_2 = \left(\sum_{i=0}^{21} b_i.2^i\right)_{10} = x'$$

- find a corresponding real number x in the range [-1,2]

$$x = -1.0 + x'.\frac{3}{2^{22}-1}$$

The evaluation function *eval* for binary vectors v is equivalent to the function f:

$$eval(v) = f(x),$$

where the chromosome v represents the real value x.

As noted earlier, the evaluation function plays the role of the environment, rating potential solutions in terms of their fitness. For example, three chromosomes:

$v_1 = (1000101110110101000111)$,

$v_2 = (0000001110000000010000)$,

$v_3 = (1110000000111111000101)$,

Correspond to values $x_1 = 0.637197$, $x_2 = -0.958973$, and $x_3 = 1.627888$, respectively. Consequently, the evaluation function would rate them as follows:

$$eval(v_1) = f(x_1) = 1.586345$$

$$eval(v_2) = f(x_2) = 0.078878$$

$$eval(v_3) = f(x_3) = 2.250650$$

Clearly, the chromosome v_3 is the best of the three chromosomes, since its evaluation returns the highest value. On the other hand, v_2 has a very low fitness.

Selection

A roulette wheel approach is adopted as the selection procedure. The roulette wheel can be constructed as follows:

1. Calculate the fitness value $eval(v_k)$ for each v_k:

$$eval(v_k) = f(x), \qquad k = 1, 2, \ldots, pop\_size$$

2. Calculate the total fitness for the population:

$$F = \sum_{k=1}^{pop\_size} eval(v_k)$$

3. Calculate selection probability p_k for each chromosome v_k:

$$p_k = \frac{eval(v_k)}{F}, \qquad k = 1, 2, \ldots, pop\_size$$

4. Calculate cumulative probability q_k for each chromosome v_k:

$$q_k = \sum_{j=1}^{k} p_j, \qquad k = 1, 2, \ldots, pop\_size.$$

The selection process begins by spinning the roulette wheel pop_size times; each time, a single chromosome is selected for a new population in the following way:

Step 1. Generate a random number r from the range $[0,1]$.

Step 2. If $r \le q_1$, then select the first chromosome v_1; otherwise, select the kth chromosome $v_k (2 \le k \le pop\_size)$ such that $q_{k-1} < r < q_k$.

Genetic operators

After pop_size vectors (not all different) are chosen, some of them will be altered. This alteration phase uses two classical genetic operators: mutation and crossover.

Mutation is an operation that alters a gene (i.e. a position in the chromosome). In binary representation, a mutation is simply a bit flip. As an example, if the fifth gene from chromosome v_3 is selected for a mutation, the current value 0 is flipped into a 1 yielding

$v_3' = (1110100000111111000101)$.

This chromosome represents the value $x_3' = 1.721638$ and $f(x_3') = -0.082257$which is a significant decrease of the fitness value of the chromosome v_3. On the other hand, if the 10^{th} gene of chromosome v_3 was selected for mutation, then v_3''=(1110000001111111000101). The corresponding real value is x_3''= 1.630818, and $f(x_3'')$= 2.343555 represent an improvement in fitness over the original value of $f(x_3)$ =2.250650.

Mutation occurs with a probability equal to the mutation rate, p_m. If, for example, p_m =0.01, it is probable that 1% of the total number of genes in the population would undergo mutation. For each gene in a given chromosome, a random number $r \in [0,1]$ is chosen. If $r < p_m$, that gene is mutated (i.e. the bit is flipped). Otherwise it is not.

After mutation is performed on the selected chromosomes, crossover is performed. This combines the features of two parent chromosomes to form two similar offspring by swapping corresponding segments of the parents. For each pair of chromosomes a random integer number *pos* is generated from the range [1..*m*-1], where *m* is the total length (number of bits) in a chromosome. The number *pos* indicates the position of the crossing points. The chromosomes $(b_1b_2...b_{pos}b_{pos+1}...b_m)$ and $(c_1c_2...c_{pos}c_{pos+1}...c_m)$ are replaced by their offspring $(b_1b_2...b_{pos}c_{pos+1}...c_m)$ and $(c_1c_2...c_{pos}b_{pos+1}...b_m)$.

For example, consider applying the crossover operator to chromosomes v_2 and v_3. First randomly select the crossover point. Assume it is selected after the 5th gene.

v_2 = (00000|01110000000010000),

v_3 = (11100|00000111111000101),

The two resulting offspring are

v_2'= (00000|**00000111111000101**),

v_3'= (11100|**01110000000010000**).

These resulting offspring fitness values are:

$f(v_2') = f(-0.998113) = 0.940865$,

$f(v_3') = f(1.666028) = 2.459245$., which are better than both of each parent.

In each case, the new vector is fitter than the old one.

Crossover occurs with a probability of p_c. If, for example, p_c=0.25, it is probable that 25% of chromosomes will undergo the crossover operation. For each chromosome, a random number $r \in [0,1]$ is generated. If $r < p_c$, that chromosome is selected for crossover. From this list of selected chromosomes, pairs are chosen at random for crossover, as described above.

Parameters

For this particular problem the following parameters have been used: population size *pop_size* = 50, probability of crossover p_c = 0.25, probability of mutation p_m =0.01. The following section presents some experimental results for such a genetic system.

Experimental results

In Table 4.1 the generation number and function value are provided for which an improvement in the evaluation function was noted. The best chromosome after 150 generations was in generation 145 where v_{max} = (1111001101000100000101), corresponding to x_{max} = 1.850773 and $f(x_{max}) \approx 2.85$. This is very close to the approximation found previously by analysis.

Table 4.1 . Results of 150 generations

Generation number	Evaluation function
1	1.441942
6	2.250003
8	2.250283
9	2.250284
10	2.250363
12	2.328077
39	2.344251
40	2.345087
51	2.738930
99	2.849246
137	2.850217
145	2.850227

4.4. Genetic Algorithms for Multiobjective Optimization

The GAs discussed in the previous section are designed to optimize single-objective functions. In the real world, however, typical optimization problems have multiple objectives, as discussed in section 3.1.1. GAs can be easily modified to solve multiobjective problems.

Multiobjective optimization seeks to optimize the components of a vector-valued function. Unlike single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto-optimal set as discussed in Chapter 2 section 2.3. Each point in this surface is optimal in the sense that no improvement can be achieved in one cost vector component that does not lead to degradation in at least one of the remaining components. That is, the solution set represents the best compromises among all the objectives.

The main difference between GAs for single objective optimization and for multiobjective optimization is that in single objective optimization, fitness can easily be defined by the value of the objective function. In multiobjective optimization, however, fitness must be based on all objective functions. Since GAs maintain a population of solutions they can search for many non-dominated solutions in parallel. Thus the concept of Pareto-optimality can be used to define fitness. The idea then is that an initial population evolves into a population that is representative of the Pareto-optimal set. Two such approaches will be discussed below – Vector Evaluated Genetic Algorithm (VEGA) (Schafter, 1985) and Multi-Objective Genetic Algorithm (MOGA) (Fonceca and Fleming, 1993).

Being aware of the potential GAs could have in multiobjective optimization, Schafter (1985) proposed an extension of simple GAs to accommodate vector-value fitness measures, which he called a VEGA. A simple vector version of the survival of the fittest process was implemented. The selection step was modified so that, at each generation, a number of sub-populations was generated by performing proportional selection according to each objective function in turn. Thus, for a problem with q objectives and population of size N, selection is used to generate q sub-populations of size N/q. These would then be shuffled together to obtain a new population of size

N, in order for the algorithm to proceed with the application of crossover and mutation in the usual way.

However, as noted by Richardson et al. (1989), shuffling all the individuals in the sub-populations together to obtain the new population is equivalent to linearly combining the fitness vector components to obtain a single-valued fitness function. The weighting coefficients, however, depend on the current population. This means that, in the general case, not only will two non-dominated individuals be sampled at different rates, but also, in the case of a concave trade-off surface, the population will tend to split into different species, each of them particularly strong in one of the objectives. Schaffer anticipated this property of VEGA and called it speciation. Speciation is undesirable in that it is opposed to the aim of finding a compromise solution.

To avoid combining objectives in any way requires a different approach to selection. Fonseca and Fleming (1993) proposed a technique which they called a MOGA in which fitness is based on ranking.

Multiobjective Pareto Ranking

Ranking of solutions according to Pareto-optimality can be done as follows. Consider an individual x_i at generation t which is dominated by $p_i^{(t)}$ individuals in the current population. Its current position in the individuals' rank can be given by

$$\text{rank}\left(x_i, t\right) = 1 + p_i^{(t)}.$$

Thus, all non-dominated individuals are assigned the best ranking, 1. The fitness of an individual can then be assigned, for example, as the reciprocal of the rank.

To clarify the concept of Pareto-ranking, consider the following example:

Maximize: $f_1 = x, \qquad f_2 = y$

Subject to: $x^2 + y^2 \le 1$ and $0 \le x,\ y \le 1$.

The Pareto front is then a quarter arc of the circle $x^2 + y^2 = 1$ at $0 \le x,\ y \le 1$. In Fig.4.1 the ranking of several points is shown.

Recently, Fonseca et al. (1995) published a survey of evolutionary algorithms for multiobjective optimization. They identified several open research issues, and provided an overview of two categories of techniques - (those which combine many criteria into one objective function and return a single value, and those, which are based on Pareto-optimality and return a set of values).

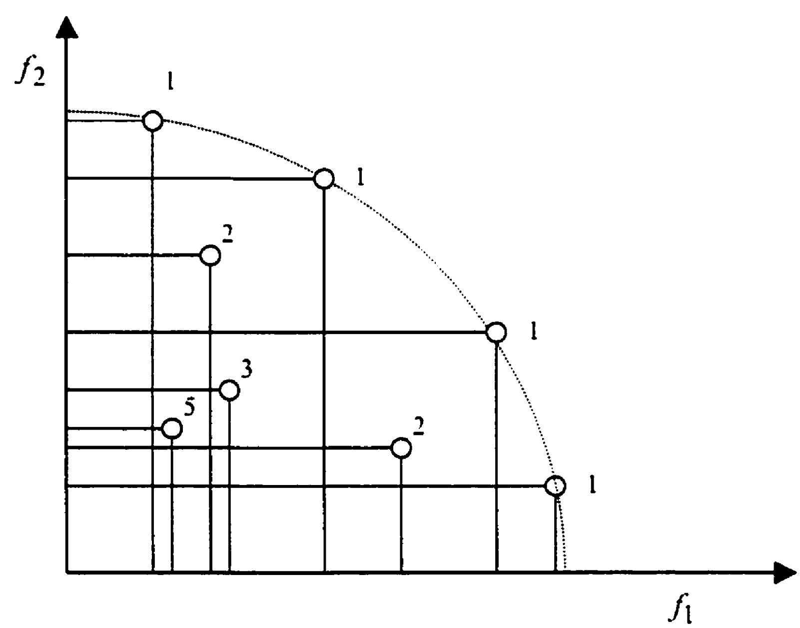


Fig. 4.1. Pareto ranking method

Chapter 5

Some examples of Multiobjective Operations Research Solutions to Forestry Problem

In chapter 3 various techniques used to solved multiobjective problems in forestry management have been discussed. The most popular one is Goal Programming (GP). After a thorough search of the literature, it appears that a Multiobjective Genetic Algorithm (MOGA) has not yet been used to solve forestry problems. As discussed in chapter 4, MOGAs can search for the Pareto-optimal set, making them potentially useful for resource allocation in forestry management where there are often multiple conflicting objectives. A MOGA would enable decision makers to choose one solution, suitable to the current situation, out of many alternatives in the Pareto-optimal set.

In this chapter two forestry problems with many conflicting objectives are solved using a MOGA and compared with the GP solution.

In the first problem, decision makers have decided in advance the target level of each objective that they want to achieve and they are satisfied if their target levels are met. In the second problem, the decision makers don't specify any target level for the goals. Instead they want to obtain the best possible solution.

5.1. A Multiobjective Forestry Problem with specified target levels of goals.

The first type of problem solved is a multiobjective forestry problem where the decision makers specify the target levels of their goals. The decision makers must also give their preferences (priorities) for the cases where not all goals are satisfied. The problem will be solved using two approaches, GP and MOGA.

The specific example solved is given by Field (1973). Josiah Freeman, of South Haven, Conn., purchased a 600-acre track of woodland in west-central Maine. His motives for acquiring this land were twofold: to provide a recreational retreat for his family, and to develop a supplementary source of income. A local consulting forester provided Mr. Freeman with a timber management plan that specifies a sustained-yield allowable cut of about 21 MBF (thousand board feet) per year. The only other major potential income-producing feature of the property, as well as an important leisure-time asset, is a cabin in a grove of pine trees near the center of the property.

At the time of purchase, Mr. Freeman had definite ideas about the management of his new property:

1) From his preliminary calculation he thought that a realistic goal would be to net about $2100 a year from timber sales and rental of the cabin.
2) He wished to ensure the availability of the property for his family's 30-day summer vacation,
3) He wanted the use of the property for his own annual 7-day fall hunting trip.
4) In addition, he preferred not to endanger the long-term production potential of the forest by exceeding the allowable cut.
5) For reasons of safety and esthetics, he felt that no timber harvesting should be going on while either he or his tenants were using the property.

These goals were constrained by the following facts:

1) The number of summer and fall days available are estimated to be 90 and 60, respectively.

2) For every four days that the cabin was rented, about one day of Mr. Freeman's own time had to be spent in maintaining the cabin and access road. Assume that half of the maintenance time is spent in fall, and half in summer.

3) The part-time timber harvesting crew he wanted to employ was available during summer day only and could turn out, on the average, about 3 (three) MBF per day.

The net returns on harvest and rental activities were estimated to be:

a) summer rental $20/ day;

b) fall rental $15/day;

c) timber $15/MBF.

5.1.1. Using Goal Programming Procedure.

Mathematical Formulation of the Problem.

First of all we formulate the goal equations.

Let x_1 = number of days of summer rental

x_2 = number of days of fall rental

x_3 = number of days of timber harvesting

x_4 = number of days of summer vacation plan

x_5 = number of days of fall hunting plan

x_6 = number of days of work days for maintenance

As discussed in section 2.3, the goal programming approach is to introduce variables d_{iu} and d_{io}, where d_{iu} represents the amount by which goal i is underachieved, and d_{io} that by which it is overachieved. The overall objective is thus to minimize these deviations. For example, the underachievement variable, d_{1u}, has value zero if goal 1 is satisfied or overachieved. Similarly the overachievement variable, d_{1o}, has value zero if goal 1 is exactly satisfied or is underachieved.

Goal 1. Net income, coming from rental and sale of timer, is about $2100.

$$20x_1+15x_2+45x_3+d_{1u}-d_{1o}=2100$$

where d_{1u} is the shortfall in achieving this income (in dollars), and d_{1o} = excess of income above the goal. The goal is to minimize d_{1u}.

Goal 2. Availability of 30 days for summer vacation.

$$x_4+d_{2u}-d_{2o}=30$$

where d_{2u} = number of days less than 30 available for vacation, and d_{2o} = number of extra days above 30 available for vacation.

The goal is to minimize d_{2u}.

Goal 3, Availability of 7 days for fall hunting

$$x_5+d_{3u}-d_{3o}=7$$

where d_{3u} = number of days less than 7 available for hunting, and d_{3o} = number of extra days above 7 available for hunting. The goal is to minimize d_{3u}.

Goal 4. Not to endanger the long term production. The goal here is to not exceed the allowable cut of 21MBF (7 days of harvesting).

$$x_3+d_{4u}-d_{4o}=7$$

where d_{4u} = number of days less than 7 available for harvesting, and d_{4o} = number of extra days above 7 available for harvesting. The goal is to minimize $d_{4u} + d_{4o}$.

Goal 5. The goal is that every 4 days rental needs one work day for maintenance. That is, $-x_1 - x_2 + 4x_6 = 0$ or $-\frac{1}{4}x_1 - \frac{1}{4}x_2 + x_6 = 0$. Hence, the formulation of goal 5 is as follows.

$$-\tfrac{1}{4}x_1 - \tfrac{1}{4}x_2 + x_6 + d_{5u} - d_{5o} = 0$$

where d_{5u} = the number of days less than the goal allocated to maintenance and d_{5o} = the number of days more than the goal available for maintenance. The goal is to minimize d_{5u}.

In summary, the five goals are:

Goal 1: Minimize d_{1u}

Goal 2: Minimize d_{2u}

Goal 3: Minimize d_{34}

Goal 4: Minimize $d_{4u} + d_{4o}$

Goal 5: Minimize d_{5u}

The constraints :

1. There are only 90 days available for summer. These days are used for rental, harvest, family's vacation, and repair. Half of the work days will occur in the summer.

   $x_1 + x_3 + x_4 + 0.5x_6 \leq 90$.

2) There are only 60 days in the fall. They are used as fall rental, fall hunting, and work days. Four days rental needs 1 day repair and 4 day fall hunting needs 1 day repair. So we can formulate the constraint as follows:

$$x_2 + x_5 + 0.5x_6 \leq 60.$$

The formulation of the problem

Initially, Mr. Freeman did not rank his goals. That is, each goal is given weight one. The problem formulation is thus, after putting zero values on variables which he did expect to be positive (d_{2o}, d_{3o}, d_{5o})

Minimize $d_{1u} + d_{2u} + d_{3u} + d_{4u} + d_{4o} + d_{5u}$

Subject to

$$20x_1 + 15x_2 + 45x_3 + d_{1u} \geq 2100$$

$$x_4 + d_{2u} = 30$$

$$x_5 + d_{3u} = 7$$

$$x_3 + d_{4u} - d_{4o} = 7$$

$$-\tfrac{1}{4}x_1 - \tfrac{1}{4}x_2 + x_6 + d_{5u} = 0$$

$$x_1 + x_3 + x_4 + 0.5x_6 \leq 90$$

$$x_2 + x_5 + 0.5x_6 \leq 60$$

All variables are non-negative.

The formulation problem is solved using LINGO (LINDO System Inc., 1999). The output is summarized in Table 5.1 below. The formulation of the model for LINGO and the output can be found in Appendix 1 and Appendix 2.

Table 5.1. Unranked goals of Mr.Freeman

Activity	Allocation (no. of days)		Goal	Deviation
Summer rental	33		Income	0
Fall rental	43		Harvest	+11
Harvest	18		Summer vacation	0
Summer vacation	30		Fall hunting	0
Fall hunting	7		Work days	0
Work	19			

We can see from this result that we cannot achieve all goals. The variable d_{4o} =11 which means that there is an overcut of 33 MBF (11 days of harvesting). This is an undesirable result of this simple problem formulation. In order to improve the result the individual goals must be given specific priorities based on other considerations.

For the purposes of this problem it is supposed that Mr. Freeman makes the following decisions. He decided that he would, most of all, like to make the $2100 per year. More would be acceptable. Meeting the allowable cut ranked second in his scheme of things, but he was twice as concerned over the consequences of exceeding the limit as he was about to undercutting. Thus d_{4o} is multiplied by 2 in the priority 2 part, as shown below. Summer and fall leisure time and working days all ranked in the third level, but assuring vacation and hunting time (equally valued) seemed three times as important as getting the work done. Thus d_{2u} and d_{3u} are multiplied by 3 in priority 3. Let P_1 refer to first priority, P_2 refer to second priority, and P_3 refer to third priority. The problem we want to solve now is

$$\text{Minimize } P_1 d_{1u} + P_2(d_{4u} + 2d_{4o}) + P_3(3d_{2u} + 3d_{3u} + d_{5u})$$

Subject to

$$20x_1 + 15x_2 + 45x_3 + d_{1u} \geq 2100$$

$$x_4 + d_{2u} = 30$$

$$x_5 + d_{3u} = 7$$

$$x_3 + d_{4u} - d_{4o} = 7$$

$$-\tfrac{1}{4}x_1 - \tfrac{1}{4}x_2 + x_6 + d_{5u} = 0$$

$$x_1 + x_3 + x_4 + 0.5x_6 \leq 90$$

$$x_2 + x_5 + 0.5x_6 \leq 60$$

All variables are non-negative.

This formulation solved using LINGO and the results are summarized in the Table 5.2. The source code of this LINGO formulation and the output can be found in Appendix 3 and Appendix 4.

Table 5.2. Ranked goals of Mr.Freeman

Activity	Allocation (no. of days)		Goal	Deviation
Summer rental	51		Income	0
Fall rental	51		Harvest	0
Harvest	7		Summer vacation	0
Summer vacation	30		Fall hunting	0
Fall hunting	7		Work days	-21
Work	4			

It is shown in Table 9 that in the first run the work days goal could be met but after this goal is ranked as ranking number 3 (third priority) this goal becomes underachieved by 21 days. On the other hand, the harvest goal that is not satisfied in the first run becomes completely satisfied as this goal is ranked as ranking number 2 (second priority). It shows that goal ranked 3, work days, is sacrificed in order to satisfy goal ranked 2, harvesting.

5.1.2. Solutions using Multiobjective Genetic Algorithm (MOGA).

The above stated problem of 5.1.1 is now solved using the MOGA algorithm (Binh, 1996) implemented in MATLAB. This algorithm is used to search for the Pareto-optimal set of a given vector-valued objective function. The source code for this function can be found in Appendix 5. A population size of 100 is used. The output can be found in Appendix 6 and rounded to integer values in Appendix7. A summary of the solution is given in Table 5.3. The table gives some extreme pareto optimal solutions for which optimal results for the goal are given regardless of the other goals. For example, to achieve goal 1 as closely as possible (i.e. satisfy the target level exactly without considering the other goals then solution number 1 might be the best. In table 5.4 some reasonably good trade-off solutions are displayed. For example, solution number 1 is quite good trade-off solution, i.e., we achieve net income of $1930 (target of $2100), 20 day summer vacation (target of a 30 days), and –2 work days (target of 0, short of only a half working day). There are other trade-off solutions in this table or we might prefer another solution from table 5.5.

Table 5.3. Some of the pareto optimal solutions of extreme values (MOGAs solution)

							Goal 1:	Goal 2	Goal 3	Goal 4	Goal 5
	Summer Rental	Fall Rental	Harvest Days	Summer Vacation	Fall Hunting	Work Days	Net Income	Summer Vacation	Hunting	Harvesting	Work Days
No.							$2,100.00	30 days	7 days	7 days	0 days
1	47	39	13	19	7	21	2110	19	7	13	-2
2	43	43	7	30	5	6	1820	30	5	7	-62
3	48	38	13	19	7	21	2115	19	7	13	-2
4	42	44	7	29	9	13	1815	29	9	7	-34
5	45	40	13	21	7	21	2085	21	7	13	-1

Table 5.4. Some of good trade- off solutions (MOGAs solution)

							Goal 1:	Goal 2	Goal 3	Goal 4	Goal 5
	Summer Rental	Fall Rental	Harvest Days	Summer Vacation	Fall Hunting	Work Days	Net Income	Summer Vacation	hunting	Harvesting	Work Days
No.							$2,100.00	30 days	7 days	7 days	0 days
1	47	39	9	20	7	21	1930	20	7	9	-2
2	48	48	7	29	7	10	1995	29	7	7	-56
3	51	38	11	22	7	10	2085	22	7	11	-49
4	43	42	7	28	6	20	1805	28	6	7	-5

Table 5.5. Selected solutions considering Mr.Freeman's preferences

								Goal 1:	Goal 2	Goal 3	Goal 4	Goal 5
	Summer Rental	Fall Rental	Harvest Days	Summer Vacation	Fall Hunting	Work Days		Net Income	Summer vacation	Hunting	Harvesting	Work days
No.								$21,000	30 days	7 days	7 days	0 days
1	48	38	13	19	7	21		2115	19	7	13	-2
2	47	39	13	20	7	20		2110	20	7	13	-6
3	47	39	13	19	6	21		2110	19	6	13	-2
4	43	39	12	21	9	21		1985	21	9	12	2
5	48	42	12	20	6	20		2130	20	6	12	-10
6	47	41	12	20	7	21		2095	20	7	12	-4
7	51	38	11	22	7	10		2085	22	7	11	-49
8	45	41	11	23	5	21		2010	23	5	11	-2
9	45	41	11	23	6	16		2010	23	6	11	-22
10	45	42	10	24	6	17		1980	24	6	10	-19
11	47	39	10	20	7	18		1975	20	7	10	-14
12	47	42	10	20	7	16		2020	20	7	10	-25
13	49	39	9	18	7	17		1970	18	7	9	-20
14	47	39	9	20	7	21		1930	20	7	9	-2
15	44	40	9	27	5	20		1885	27	5	9	-4
16	51	38	11	22	7	10		2085	22	7	11	-49
17	42	40	8	29	6	21		1800	29	6	8	2
18	43	46	8	30	7	13		1910	30	7	8	-37
19	43	41	8	30	5	18		1835	30	5	8	-12
20	48	48	7	29	7	10		1995	29	7	7	-56
21	47	45	7	29	7	12		1930	29	7	7	-44
22	43	43	7	30	5	6		1820	30	5	7	-62
23	43	46	7	30	7	6		1865	30	7	7	-65
24	47	45	7	29	7	6		1930	29	7	7	-68
25	43	42	7	28	6	20		1805	28	6	7	-5

What we can see from this result is that the MOGA's solution gives many alternative schedules with many different trade-offs. This allows us to choose a solution that is acceptable based on other considerations which are not part of the mathematical problem. In contrast the GP solution only gives one solution without displaying any other potentially good trade-offs. Even for this simple problem, the genetic algorithm generates solutions no worse (if not better) than that generated by GP approach. On the other hand, MOGAs are capable of handling complex problems where the GP approach might not be very efficient.

5.2. A Multiobjective Forestry Problem without specified goal target levels.

Here we will solve a multiobjective forestry problem where the decision makers don't specify targets for the goals they want to achieve. In this case the decision makers might have no idea about the target levels they can achieve. The problem is a modification of a reforestation budgeting and planting stock allocation study of MacLean (1980). MacLean used linear programming to solve the original problem. However, the GP procedure and MOGAs techniques have been used to solve the modified problem. The specific goal programming used here was proposed by Walker, H. D. (1984).

The general background of the problem is as follows. The silviculture staff in the district office of a public forest management agency is planning reforestation activities for the coming season. An area of 5000 ha of unstocked land is available for this purpose. The land has been classified into three types: 1000 ha of site type X, 2200 ha of site type Y, and 1800 ha of site type Z. The staff are confident that areas within any of these site types which are not treated will not regenerate naturally to commercial stands. Three species (A,B, and C) are being considered. For each of the nine combinations of species and site type, three alternative reforestation treatments (1,2, and 3) have been defined, with high, medium, and low costs and yields, respectively. Treatments 1 and 2 involve planting, while treatment 3 involves seeding in combination with one or more other operations.

Specific data for the problem are as follows: A limited supply of planting stock is available. Based upon prescribed planting densities, up to 700 ha of species A, 400 ha of species B, and 400 ha of species C could be planted. Seed for all three species is available in abundance. To meet expected future wood requirements, minimum yields have been established. These yields are expressed as average yields per year over the chosen rotation ages. Species A and B have a combined minimum yield of 3300 $m^3/year$ and species C has a minimum yield of 1700 $m^3/year$. An overall minimum yield for the area is set at 5500 $m^3/year$. A budget of $800,000 is available. No equipment or labor shortages are foreseen. Table 6 shows the expected establishment costs and yields in cubic meter per hectare per year for each of the 27 activities.

Table 5.6. Establishment costs and expected yields for sample problem

Species	Site type	Silvicultural treatment								
		1			2			3		
		Cost ($/ha)	Yield		Cost ($/ha)	Yield		Cost ($/ha)	Yield	
A	X	350	2.7	x_1	310	1.6	x_2	140	1.0	x_3
A	Y	350	2.3	x_4	270	1.4	x_5	140	0.6	x_6
A	Z	270	1.4	x_7	170	1.3	x_8	100	0.6	x_9
B	X	350	2.9	x_{10}	310	1.9	x_{11}	140	1.1	x_{12}
B	Y	350	2.3	x_{13}	270	1.4	x_{14}	140	0.6	x_{15}
B	Z	270	1.1	x_{16}	170	1.0	x_{17}	100	0.5	x_{18}
C	X	310	3.4	x_{19}	170	2.6	x_{20}	90	1.1	x_{21}
C	Y	310	2.9	x_{22}	170	2.1	x_{23}	90	0.8	x_{24}
C	Z	170	1.4	x_{25}	130	1.0	x_{26}	90	0.5	x_{27}

The silviculture staff wishes to incorporate three different goals into the analysis. These include maximum expected annual volume yields, maximum area replanted, and minimum cost to achieve required yields. The staff realizes that these goals are competitive, but wishes to attempt to meet all of them concurrently.

Problem formulation

A total of 27 variables, x_i, $i = 1,2,...,27$, each corresponding to the area planted under a specific regeneration system (combination of species, site type, and silvicultural treatment), is needed, as shown in Table 5.6.

Five types of constraints are needed. The budget constraint is formulated as:

$$\sum_{i=1}^{27} C_i x_i \le \$800{,}000 \tag{1}$$

where C_i is the cost (dollars per hectare) of regeneration system i, and x_i is the area (hectares) assigned to regeneration system i.

The planting stock area constraints are:

$$\begin{aligned} &x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \le 700 \\ &x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} \le 400 \\ &x_{19} + x_{20} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} \le 400 \end{aligned} \tag{2}$$

The minimum volume yield constraints needed are:

$$\sum_{i=1}^{18} Y_i x_i \ge 3300 \tag{3}$$

$$\sum_{i=19}^{27} Y_i x_i \ge 1700 \tag{4}$$

$$\sum_{i=1}^{27} Y_i x_i \ge 5500 \tag{5}$$

where Y_i is the yield (cubic meters per hectare per year) of species A or B under regeneration system i.

Three maximum area constraints are needed:

$$\begin{aligned} &\sum_{i=1}^{3} x_i + \sum_{i=10}^{12} x_i + \sum_{i=19}^{21} x_i \le 1000 \\ &\sum_{i=4}^{6} x_i + \sum_{i=13}^{15} x_i + \sum_{i=22}^{24} x_i \le 2200 \\ &\sum_{i=7}^{9} x_i + \sum_{i=16}^{18} x_i + \sum_{i=25}^{27} x_i \le 1800 \end{aligned} \tag{6}$$

Additional nonnegativity constraints ensure that all variables are assigned nonnegative values.

The three goals are:

$$\text{Maximize } Z_1 = \sum_{i=1}^{27} Y_i x_i \qquad [7]$$

$$\text{Maximize } Z_2 = \sum_{i=1}^{27} x_i \qquad [8]$$

$$\text{Minimize } Z_3 = \sum_{i=1}^{27} C_i x_i \qquad [9]$$

where Z_1 is the total expected yield (cubic meters per year), Z_2 is the total area replanted (hectares), and Z_3 is the total expected replanting cost (dollars).

5.2.1. Goal Programming Approach

Constraints [3] – [6] define a feasible set of solutions. Within this set the feasible policy space for each goal is determined by formulating and solving a pair of linear programming (LP) problems, including all constraints but including only the one goal. One of these problems maximizes the goal level, while the other minimizes the goal level. The best goal level is called the simple optimal level and the worst goal level is called the worst feasible level. The interval between the simple optimal and worst feasible levels is the feasible policy space. It indicates to decision makers the range of possible attainment levels for that goal. Deletion or modification of any other goals, or the addition of new goals, may or may not affect this feasible policy space. The results are summarized in Table 5.7.

The multiobjective programming problem is now solved using ordinal ranking GP. The problem is formulated as a GP problem, with goal levels set to the best values in the feasible policy space. In ordinal ranking, each of the three goals is ranked with priority one, two, or three. Table 5.8 shows the goal attainment levels and solutions associated with each of the six possible ordinal rankings. Having these alternative solutions, the decision makers can choose one of those alternatives according to their preference. If none of the alternatives are satisfactory the decision makers must then specify their preferred target level of each goal, and new solutions would be found.

Table 5.7. Feasible and optimal goal attainment levels and policy space

Goal	**Goal attainment levels and policy spaces**		
	Simple Optimal level	Worst feasible level	Feasible policy space
Maximum volume (m^3 / year)	6473	5500	973
Maximum area (ha)	5000	3209	1791
Minimum cost ($)	652082	800000	147918

Table 5.8. Solution for the problem with ordinal goal ranking

	Goal ranks					
	Solution 1:	Solution 2:	Solution 3:	Solution 4:	Solution 5:	Solution 6
Activities	1_2_3	1_3_2	2_1_3	2_3_1	3_1_2	3_2_1
X1	200.0	200.0	200.0	0.0	285.0	285.0
X2	0.0	0.0	0.0	0.0	0.0	0.0
X3	0.0	0.0	0.0	0.0	0.0	0.0
X4	278.8	278.8	278.8	0.0	0.0	0.0
X5	0.0	0.0	0.0	0.0	0.0	0.0
X6	0.0	0.0	0.0	0.0	0.0	0.0
X7	0.0	0.0	0.0	0.0	0.0	0.0
X8	221.3	221.3	221.3	700.0	415.0	415.0
X9	1578.8	1578.8	1578.8	1100.0	1385.0	1385.0
X10	400.0	400.0	400.0	400.0	400.0	400.0
X11	0.0	0.0	0.0	0.0	0.0	0.0
X12	0.0	0.0	0.0	518.2	0.1	0.0
X13	0.0	0.0	0.0	0.0	0.0	0.0
X14	0.0	0.0	0.0	0.0	0.0	0.0
X15	0.0	0.0	0.0	0.0	0.0	0.0
X16	0.0	0.0	0.0	0.0	0.0	0.0
X17	0.0	0.0	0.0	0.0	0.0	0.0
X18	0.0	0.0	0.0	0.0	0.0	0.0
X19	400.0	400.0	400.0	0.0	0.0	0.0
X20	0.0	0.0	0.0	81.8	315.0	315.0
X21	0.0	0.0	0.0	0.0	0.0	0.0
X22	0.0	0.0	0.0	0.0	0.0	0.0
X23	0.0	0.0	0.0	174.8	85.0	85.0
X24	1921.3	1921.3	1921.3	2025.2	1503.1	1503.1
X25	0.0	0.0	0.0	0.0	0.0	0.0
X26	0.0	0.0	0.0	0.0	0.0	0.0
X27	0.0	0.0	0.0	0.0	0.0	0.0
Volume	**6473.0**	**6473.0**	**6473.0**	**5500.0**	**5500.0**	**5500.0**
Area	**5000.0**	**5000.0**	**5000.0**	**5000.0**	**4388.2**	**4388.2**
Cost	**800000.0**	**800000.0**	**800000.0**	**667440.6**	**652082.0**	**652082.0**

The results in Table 5.7 gives decision makers ideas about possible maximum and minimum values of each goal, and Table 5.8 gives several solutions that they can choose from. However, these solutions are very extreme in the sense that one goal is satisfied completely but some other goals may be very far from their optimal values. In real problems, a solution representing more compromise is usually preferred. In this example the decision makers might prefer results (volume and cost) that lie between those of solutions 3 and 4 in Table 5.8. If that were the case, the decision makers would have to specify new target levels and the problem would be reformulated and resolved. Much interaction with forestry users is necessary for this process, which can be very time consuming and costly.

5.2.2. Multiobjective Genetic Algorithm Solution

The problem of 5.2 is now solved using the MOGA previously described in 5.1.2, run in MATLAB. Optimization in this case is set up as a minimization. The source code of the function for this second problem can be found in Appendix 8. A population size of 100 is used. Seven of the better solutions are shown in Table 5.9. If the user does not find the initially selected solutions to be satisfactory, it might be beneficial to show other alternative solutions generated. In Table 5.10 are 40 different alternative solutions.

Table 5.9. Some alternatives of MOGA solution for reforestation budget allocation

No.	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15
1	198.6	5.4	6.5	281.9	3.7	3.4	7.3	223.7	157.3	400.3	1.8	6.7	0.2	5.7	8.7
2	197.3	0.5	1.6	276.8	0.3	2.9	1.8	216.5	1576.2	396.7	3.2	2.9	0.2	0.9	5.2
3	204.5	7.4	9.2	98.8	12.2	11.7	6.0	303.6	1304.1	405.0	11.4	11.9	13.0	12.0	7.3
4	203.7	6.4	14.2	99.6	6.1	10.1	14.9	300.6	1297.4	398.7	12.5	14.7	12.3	6.7	7.5
5	197.7	6.2	9.6	96.3	7.2	5.3	11.9	303.6	1300.0	397.7	10.3	9.3	14.1	5.3	13.6
6	195.9	10.0	10.9	95.0	7.2	10.6	8.2	299.3	1300.9	398.7	9.5	8.1	7.2	5.8	8.1
7	127.3	235.0	216.0	205.5	17.8	192.5	119.3	50.7	290.3	30.1	167.5	64.0	88.0	3.2	4.8

No.	x16	x17	x18	x19	x20	x21	x22	x23	x24	x25	x26	x27	Objective 1 Volume	Objective 2 Area	Objective 3 Cost
1	8.7	3.9	1.8	403.3	6.6	6.4	5.9	6.0	1919.2	1.9	5.5	1.1	6620	5100	820820
2	1.3	3.2	4.6	404.8	1.3	7.9	4.6	1.8	1922.0	4.0	3.9	1.3	6530	5040	806760
3	9.8	12.4	8.6	202.1	199.9	14.8	22.7	11.3	1999.0	13.5	12.2	12.6	6250	4940	751430
4	10.7	14.7	12.4	200.7	202.9	10.4	20.8	8.4	1997.2	17.9	11.4	14.8	6220	4930	748420
5	10.0	12.1	7.9	195.3	200.6	11.9	21.9	12.6	1995.4	19.5	11.6	12.9	6170	4900	741380
6	7.7	11.4	5.2	204.6	19.9	8.1	18.9	11.9	1997.8	10.3	14.3	14.0	6140	4880	737030
7	14.1	66.9	294.1	232.5	214.8	291.3	292.5	55.4	60.1	252.5	187.3	249.5	6220	4020	799620

Table 5,10. MOGAs' solution for reforestration budget allocation

No.	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20
1	198.6	5.4	6.5	281.9	3.7	3.4	7.3	223.7	157.3	400.3	1.8	6.7	0.2	5.7	8.7	8.7	3.9	1.8	403.3	6.6
2	203.1	1.7	5.7	281.7	5.5	1.3	2.7	225.4	1581.3	398.0	7.3	3.8	6.5	1.9	2.3	4.4	4.8	5.2	401.7	6.2
3	203.1	3.2	5.7	281.5	0.4	6.6	7.3	225.8	1583.3	402.9	2.6	6.9	2.9	5.6	8.4	2.1	3.3	4.3	398.0	1.6
4	197.8	0.5	5.4	277.8	4.0	6.8	2.2	224.5	1581.7	403.9	6.4	3.7	4.9	5.0	5.7	3.1	5.8	3.1	401.3	4.1
5	201.6	1.6	7.4	277.8	6.0	2.7	0.0	222.6	1580.8	402.6	4.8	3.3	6.1	3.1	6.2	4.6	4.7	4.2	400.0	0.1
6	201.1	7.2	4.2	274.3	0.1	3.4	0.8	221.6	1580.9	398.6	7.4	1.6	2.6	3.9	5.4	6.7	4.9	6.6	403.7	0.7
7	196.5	0.1	2.7	277.7	0.5	2.2	7.3	220.1	1582.0	396.1	4.5	6.1	3.8	1.2	2.4	6.1	4.3	4.6	403.1	1.9
8	201.6	3.5	5.7	276.0	7.5	4.3	1.3	222.9	1582.7	396.8	5.5	3.4	0.6	0.5	0.3	5.5	1.9	0.1	396.9	5.5
9	195.6	8.0	2.5	282.2	6.1	5.0	0.8	216.3	1579.5	398.5	4.7	4.1	1.7	0.5	1.2	2.8	1.9	0.8	396.3	6.8
10	198.4	2.5	7.6	278.8	5.3	5.1	1.8	222.9	1575.9	399.8	0.3	1.6	0.9	2.8	3.4	3.8	2.4	3.9	395.8	6.1
11	202.2	2.9	4.4	276.1	0.9	2.1	2.7	224.1	1574.1	400.1	1.8	0.1	5.5	4.3	0.5	1.8	5.7	0.1	396.3	1.7
12	195.4	0.5	2.0	276.2	3.4	2.3	2.2	219.9	1580.4	397.1	2.3	4.9	5.1	2.0	4.4	1.4	4.7	5.9	298.8	1.0
13	195.4	0.5	2.0	276.2	3.4	2.3	2.2	219.9	1580.4	397.1	2.3	4.9	5.1	2.0	4.4	1.4	4.7	5.9	398.8	1.0
14	203.1	1.0	0.4	274.1	5.3	1.1	5.4	219.6	1574.3	396.2	1.4	5.1	0.8	3.5	8.5	5.7	2.8	2.5	395.0	5.7
15	199.7	3.0	4.4	275.3	1.4	5.5	1.8	221.0	1574.8	396.6	0.5	1.8	6.9	0.5	2.0	3.3	2.9	5.2	395.2	6.8
16	197.3	0.5	1.6	276.8	0.3	2.9	1.8	216.5	1576.2	396.7	3.2	2.9	0.2	0.9	5.2	1.3	3.2	4.6	404.8	1.3
17	197.3	0.5	1.6	276.8	0.3	2.9	1.8	216.5	1576.2	396.7	3.2	2.9	0.2	0.9	5.2	1.3	3.2	4.6	404.8	1.3
18	127.3	235.0	216.0	205.5	17.8	192.5	119.3	50.7	290.3	30.1	167.5	64.0	88.0	3.2	4.8	14.1	66.9	294.1	232.5	214.8
19	109.3	254.7	129.6	201.4	19.3	29.2	251.3	119.9	176.0	143.5	43.9	11.6	97.2	28.9	18.1	260.5	250.9	221.9	81.9	80.1
20	170.9	49.0	94.6	22.9	46.1	101.2	144.5	149.7	228.0	53.0	255.0	168.5	29.1	153.4	138.7	142.3	140.9	278.1	260.3	259.0
21	132.4	209.8	91.3	71.5	164.5	163.4	276.0	171.9	257.4	89.0	12.8	219.7	49.2	34.4	146.6	159.8	149.5	23.3	163.3	284.3
22	120.2	2.4	17.7	35.1	180.1	122.1	37.5	295.7	176.0	36.7	271.8	156.8	285.0	96.9	25.1	241.3	162.1	144.2	110.6	199.7
23	52.4	236.0	162.1	273.8	36.9	97.1	142.7	159.7	49.7	22.5	145.7	222.8	150.2	53.3	117.8	28.2	234.3	257.2	111.9	239.3
24	184.0	71.4	217.9	128.1	295.9	219.6	157.1	90.9	127.0	293.0	79.3	191.0	26.7	68.3	130.3	16.5	45.3	78.3	37.0	174.0
25	203.9	12.8	13.0	103.0	13.8	11.0	12.6	297.7	1304.1	399.2	14.0	8.7	12.8	13.7	12.3	9.6	14.2	12.4	204.8	198.7
26	203.0	11.5	13.4	101.8	5.2	6.0	11.9	304.7	1301.1	404.2	9.8	10.6	14.0	10.8	12.2	11.4	9.9	12.8	202.4	203.8
27	204.5	7.4	9.2	98.8	12.2	11.7	6.0	303.6	1304.1	405.0	11.4	11.9	13.0	12.0	7.3	9.8	12.4	8.6	202.1	199.9
28	202.6	5.2	6.5	100.5	13.2	6.6	10.0	304.2	1304.4	401.2	11.5	14.9	13.8	5.9	13.8	11.4	11.4	13.2	202.6	202.6
29	204.5	11.9	13.6	103.1	10.7	10.4	7.4	304.3	1303.1	401.0	7.5	12.5	9.1	12.7	11.5	9.8	10.5	6.5	201.0	202.2
30	203.7	6.4	14.2	99.6	6.1	10.1	14.9	300.6	1297.4	398.7	12.5	14.7	12.3	6.7	7.5	10.7	14.7	12.4	200.7	202.9
31	198.3	12.2	8.7	95.5	11.7	6.9	11.5	296.4	1303.2	395.9	10.4	5.1	14.4	13.2	10.9	7.7	13.6	8.4	203.8	201.8
32	99.5	16.3	245.1	36.2	199.6	225.6	151.8	94.4	173.7	134.1	51.1	43.9	99.4	69.6	0.7	245.4	288.4	170.1	104.8	255.0
33	198.2	7.6	14.8	103.9	7.4	10.1	7.2	304.1	1297.6	398.2	9.8	13.9	12.8	7.7	11.7	12.7	5.0	8.9	196.8	201.2
34	197.7	6.2	9.6	96.3	7.2	5.3	11.9	303.6	1300.0	397.7	10.3	9.3	14.1	5.3	13.6	10.0	12.1	7.9	195.3	200.6
35	204.1	8.8	13.5	97.7	12.4	7.4	11.4	296.2	1299.0	397.9	6.0	13.7	5.6	14.0	8.8	10.6	11.2	6.1	196.8	198.9
36	198.0	9.4	9.2	96.3	9.6	6.6	13.4	298.4	1297.0	398.3	9.4	7.9	10.0	5.8	9.3	10.5	5.7	11.3	202.0	204.4
37	197.0	8.8	5.0	100.6	14.2	6.6	5.7	302.9	1302.1	398.6	6.2	6.9	11.7	5.3	8.1	7.5	5.2	11.2	196.8	204.4
38	195.0	8.4	10.2	96.3	8.5	14.2	10.7	301.7	1300.3	400.3	9.4	12.0	9.3	5.0	13.0	5.3	6.6	5.3	196.0	198.7
39	195.9	10.0	10.9	95.0	7.2	10.6	8.2	299.3	1300.9	398.7	9.5	8.1	7.2	5.8	8.1	7.7	11.4	5.2	204.6	199.2
40	195.9	10.0	10.9	95.0	7.2	10.6	8.2	299.3	1300.9	398.7	9.5	8.1	7.2	5.8	8.1	7.7	11.4	5.2	204.6	19.9

No	x21	x22	x23	x24	x25	x26	x27	Objective 1 Volume	Objective 2 Area	Objective 3 Cost
1	6 4	5 9	6 0	1919 2	1 9	5 5	1 1	6620	5100	820820
2	7.7	5 3	3 7	1924 0	2 9	0 8	3 6	6620	5100	820230
3	3 7	3 2	6 4	1923 1	3 9	1 4	5 8	6610	5100	819670
4	2 0	5 5	6 5	1923 3	2 3	6 0	6 7	6610	5100	819180
5	7 6	6 4	5 8	1921 5	0 9	3 9	7 4	6600	5090	818340
6	7 7	3 8	4 9	1917 9	0 6	3 6	0 8	6580	5080	815360
7	4 5	3 3	4 4	1919 9	3 3	5 1	2 5	6560	5070	811840
8	2 7	0 5	0 2	1923 5	5 2	6 1	7 7	6540	5070	811370
9	4 2	2 6	2 5	1917 7	2 2	2 0	0 8	6540	5050	809830
10	2 9	3 2	3 2	1917 9	2 3	2 3	2 0	6540	5050	809570
11	7.5	0 9	0 3	1919 7	0 2	6 9	0 5	6530	5040	808860
12	1 5	6 3	0 3	1925 8	5 8	2 4	3 8	6530	5060	808780
13	1.5	6 3	0 3	1925 8	5 8	2 4	3 8	6530	5060	808780
14	1 7	3 3	5 2	1916 7	0 8	1 0	2 0	6520	5040	808140
15	3 7	2 7	1 6	1917 4	2 6	6 9	5 4	6530	5050	807930
16	7 9	4 6	1 8	1922 0	4 0	3 9	1 3	6530	5040	806760
17	7 9	4 6	1 8	1922 0	4 0	3 9	1 3	6530	5040	806760
18	291 3	292 5	55 4	60 1	252 5	187 3	249 5	6220	4020	799620
19	258 7	125 0	293 7	61 6	76 0	242 8	291 8	5680	3880	793230
20	172 5	228 0	172 0	190 1	71 5	30 6	231 1	6140	3980	786790
21	77 4	133 6	17 8	294 1	178 7	157 4	281 2	5740	4010	782730
22	134 1	94 0	251 3	77 8	249 6	78 1	116 3	5790	3720	779110
23	226 1	161 8	29 9	206 3	153 2	16 0	237 4	5630	3820	764130
24	286 4	130 7	227 8	246 4	23 2	94 8	133 1	5770	3770	759520
25	14 6	17 8	9 6	1995 6	13 7	16 5	13 9	6250	4950	755370
26	11 4	21 9	5 2	2000 5	17 1	19 2	5 8	6260	4940	753130
27	14 8	22 7	11 3	1999 0	13 5	12 2	12 6	6250	4940	751430
28	14.9	18 0	10 3	1998 3	17 5	18 8	9 8	6240	4940	750790
29	8 4	20 2	7 7	2002 3	16 7	14 2	13 2	6230	4940	750410
30	10.4	20 8	8 4	1997 2	17 9	11 4	14 8	6220	4930	748420
31	13 8	24 8	11 1	1999 6	18 0	14 8	14 6	6220	4930	748180
32	297 1	54 2	163 5	212 9	240 7	106 2	130 5	5550	3950	745210
33	9 0	16 4	8 3	2004 0	16 5	17 9	13 4	6180	4920	743960
34	11 9	21 9	12 6	1995 4	19 5	11 6	12 9	6170	4900	741380
35	10 4	18 0	5 5	1999 8	10 7	15 6	7 6	6150	4890	740110
36	8 3	19 8	7 6	1997 6	13 6	12 8	13 0	6160	4890	739980
37	12 0	18 4	8 9	2003 6	14 2	14 0	7 9	6150	4880	737910
38	8 9	21 4	6 3	1999 8	19 2	13 2	13 4	6140	4890	737880
39	8 1	18 9	11 9	1997 8	10 3	14 3	14 0	6140	4880	737030
40	8 1	18 9	11 9	1997 8	10 3	14 3	14 0	6140	4880	737030

Some of the MOGA results are very similar to GP solutions. For example, GP solution 3 is (6473, 5000, 8000) for goal 1, goal 2, and goal 3 respectively and MOGA solution 2 (Table 5.9) is (6530, 5040, 806760). While all GP solutions in Table 5.8 are extreme non-dominated solutions, the MOGA solutions mostly represent tradeoffs between extreme solutions, along the Pareto-front. One good tradeoff solution that decision makers might consider is the one between GP solutions 3 and 4 in Table 5.8, which are (6473, 500, 8000) and (5500, 5000, 667440) for volume, area, and cost respectively. Where GP failed to find a tradeoff between these two solutions, MOGA succeeded. For example, MOGA solution 3 in Table 5.9 is (6250, 4900, 751430) for volume, area, and cost respectively. Solution number 1 in Table 5.9 (6620, 5100, 820830) is another possible good tradeoff from MOGA if the decision makers are willing to increase their budget. That is, by adding only $20820 to their cost, they receive significant increase in volume (147 m^3 per ha per year) and area (100 ha).

The ability of MOGA to search for many Pareto-optimal solutions, where representing tradeoffs between the extremes, makes it possible for the forestry analysts to minimize interaction with the forest users (decision makers). Multiple meetings between the analysts and the forest users would not be necessary resulting in time and cost savings.

Chapter 6

Conclusion

For the last few decades, operation research (OR) has been intensively used in forestry management. Initially forest analysts used one of the OR methods, linear programming (LP), for decision making. Because of the incapability of LP to accommodate many forestry problems and because of the advances of OR technique, forestry analysts began to use many other different OR techniques to solve their problems. The most common of these are goal programming (GP), fuzzy programming, shortest path algorithms, and dynamic programming.

Due to the increasing demands from forests, forestry problems are better formulated as multiobjective problems. Some of these objectives might be in conflict with each other. For example, forest industries want to maximize the profit, people surrounding the forests want to use forest for their benefits, wildlife need forest for forage and shelter, and environmental impacts of the actions should be minimized. These demands could not possibly be met simultaneously. Therefore, forest analysts must find a good tradeoff between these conflicting demands. Conventional OR techniques, usually GP, have been used with limited success to accomplish this. GP can be easily used to find some Pareto-optimal solutions but these solutions are usually extreme, in that they optimize one goal at the expense of the others. Usually, solutions representing better tradeoffs between all goals are preferable. Therefore, it would be beneficial if an OR technique were able to find many Pareto-optimal solutions, both extreme and non-extreme, in a single run.

Recently developed techniques called genetic algorithms have a different approach from that of conventional OR techniques. They operate on a population of solutions. This population evolves from generation to generation into one containing better solutions to the problem. An extension of these GAs, called a multiobjective genetic algorithm (MOGA), is designed to search for many Pareto-optimal solutions where most of those are not extreme. The ability of this MOGA to search for the Pareto-optimal set makes it useful for solving conflicting multiobjective forestry problems.

In this thesis, two multiobjective forestry problems are solved using MOGA. In the first problem the decision makers specify their target level of each objective (goal) and in the second problem they do not specify the target levels or priorities of the objectives. In both problems, MOGA generates some extreme solutions, similar to those generated by GP, but it also produce some very good tradeoff solutions. Thus MOGA is demonstrated to be able to generate many potential solutions in a single run. This minimizes interaction between forest analysts and forestry users, improving the efficiency of the overall process. Since these are both real problems, it is shown that MOGA is a valuable tool that could be used by foresters.

The set of MOGA solutions shows that the algorithm does not search for Pareto-optima uniformly. In a large population there are only a few solutions that are truly unique. That is, some of the individuals in the population converge to exactly the same solution. It would be ideal if the individuals were spread more uniformly throughout the Pareto-optimal set, giving the decision makers a greater variety of potential strategies. Further work is needed for this purpose.

Bibliography

Allen, J.A. (1986), "Multiobjective regional forest planning using the noninferior set estimation (NISE) method in Tanzania and the United States", For. Sci. 32(2): 517-533.

Amidon, E. and Akin, G. (1968), "Dynamic programming to determine optimum levels of growing stock", Forest Science 14, 287-291.

Anderson, F.J. (1976), "Control theory and the optimum timber rotation", Forest Science 22, 242-246.

Bare, B.B. and Field, R.C. (1987), "An evaluation of FORPLAN from an operations research perspective". In FORPLAN: An evaluation of a forest planning tool. Symposium Proceedings. U.S. For. Serv. Rocky Mt. For. Range Exp. Stn. Gen. Tech. Rep. RM-140. pp. 133-144.

Bare, B.B. and Mendoza, G.A. (1988), "Multiple objective forest land management planning: An illustration," European Journal of Operational Research, Vol. 34, No.1, pp. 44-55.

Bellman, R., and Zadeh, L. (1970), "Decision making in fuzzy environments", Manage. Sci. 17:141-156.

Binh, T.T. (1996), "Evolution Strategy Toolbox", http://ifatwww.et.uni-magdeburg.de/~binh/evolution/esdoc.html

Bron, C., and Kerbosch, J. (1983), "Finding all cliques of an undirected graph", Communications of the ACM 16, 575-577.

Byrne, J.J., Nelson, R.J., and Googins, P.H. (1960), "Logging road handbook: the effect of road design on hauling costs", Agric. Handb. No. 183, 65 p. USDA For. Serv., Washington, DC.

Carson, W. and Dykstra, D.P. (1978), "Programs for road network planning", USDA Forest Service, general technical report PNW-67, Portland, Oregon.

Charnes, A., and Cooper, W.W. (1961), "Management Models and Industrial Applications of Linear Programming", Wiley, New York.

Clements, S.E., Dallain, P.L., and Jamnick, M.S. (1990), "An operational, spatially constrained harvest scheduling model", Can. J. For. Res. 20:1438-1447.

Daust, D.K., and Nelson, J.D. (1993), "Spatial reduction factors for strata-based harvest schedules", Forest Science, Vol. 39, No. 1, pp. 152-165.

DeJong, K.A., (1988), "Learning with genetic algorithm: An overview". Machine Learning, Vol. 3, pp.121-138.

De Kluyver, K.A. (1979), "On the importance of goal-norming in non-preemptive goal programming", Opsearch, 16: 88-97.

Dyer, A.A, Hof, J.G., Kelly, J.W., Grim, S.A., and Alward, G.S., (1979), "Implications of goal programming in forest resource allocation". For. Sci. 25(4), 535-543.

Eng, G. (1982), "A methodology for forest outturn assessment and optimal tree bucking", M.Sc. thesis, University of Canterbury, Christchurch, New Zealand.

Eng, G., Daellenbach, H.G., and Whyte, A.G.D. (1986), "Bucking tree-length stems optimally", Canadian Journal of Forest Research, Vol. 16, No.5, pp. 1030-1035.

Faustman, M. (1849), "Calculation of the value which forest land and immature stands possess for forestry", translated by W. Linnard (1968), institute paper 42, Commonwealth Forest Institute, University of Oxford, England.

Field, D.B. (1973), "Goal programming for forest management", Forest Science 19, 125-135.

Fonseca, C.M., and Fleming, P.J. (1993), "Genetic algorithms for multiobjective optimization: Formulation, Discussion, and Generalization", in Genetic Algorithms: Proc. 5th Int. Conf. On Genetic Algorithms (San Mateo, CA), Forrest, S., ed. Morgan Kaufmann.

Fonseca, C.M., and Fleming, P.J. (1995), "An overview of evolutionary algorithms in multiobjective optimization", Evolutionary Computation, Vol. 3, No. 1, pp. 165-180.

Grevatt, J.G. and Wardle, P.A. (1967), "Two mathematical models to aid in nursery planning", XIV World Congress of IUFRO, Section 25, Munich, Germany, Fed. Rep., 362-370.

Gross, T.E., and Dykstra, D.P. (1988), "Harvest scheduling with nonadjacency constraints" In Proceedings, Soc. Am. For. Nat. Conv., 16-19 Oct. 1988, Washington, DC. pp. 310-315.

Grossman, I. S., (19994), "Applications supplement to elementary linear algebra", Saunders College Publishing, United Stated of America, p. 168.

Hannan, E. (1980), "Nondominance in goal programming", INFOR, 18: 300-309.

Haimes, Y.Y., Hall, Y.Y., and Freedman, H.T. (1975), "Multiobjective optimization in water resource systems", Elsevier Science Publishing Co., Inc., New York.

Holland, J. (1975), "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor.

Hokans, R.H. (1983), "Evaluating spatial feasibility of harvest schedules with simulated stand-selection decisions", J. For. 81(7): 601-603, 613.

Hyberg, B.T. (1987), "Multi-attibute decision theory and forest management: A discussion and application", Forest Science, Vol. 33, No. 4, pp. 835-845.

Hyberg, B.T., and Holthausen, D.M. (1989), "The behaviour of nonindustrial private forest landowners", Canadian Journal of Forest Research 19, pp. 1014-1023.

Ignizio, J.P. (1976), "Goal programming and extensions", D.C. Heath and Co., Lexington, MA.

Ignizio, J.P. (1981), "The determination of a subset of efficient solutions via goal programming", Computers and Operations Research 8(1), 9-16.

Ignizio, J.P. (1983), "Generalized goal programming", Comput. & Oper. Res. 10(4): 277-289.

Jamnick, M., Walters, K. (1991), "Harvest blocking, adjacency constraints, and timber harvest volumes", Proceedings of the Planning and Implementing Future Forest Operations, International Mountain Logging and 8th Pacific Northwest Skyline Symposium, 14-16 December 1992, Bellevue, Washington, USA, Published by University of Washington, Forestry Continuing Education Department, pp. 241-247.

Johnson, K.N., Stuart, T., and Crimm, S.A. (1986), "FORPLAN, Version 2: An overview," USDA Forest Service, Land Management Planning Systems Section, Washington, DC.

Kao, C. and Brodie, J.D. (1979), "Determination of optimal thinning interval using dynamic programming", Forest Science 25(4), 672-674.

Kangas, J., and Pukkala, T. (1992), "A decision theoretic approach applied to goal programming of forest management", Silva Fennica 1992, Vol. 26, No. 3, pp. 169-176.

Keeney, R.L., and Faiffa, H. (1976), "Decision analysis with multiple conflicting objectives", John Wiley & Sons, Inc., New York. 736 p.

Korhonen, P., and Laakso, J. (1986a), "A visual interactive method for solving the multiple criteria problem", European Journal of Operational Research 26: 335-363.

Kosko, B. (1992), "Neural Networks and Fuzzy Systems", Englewood Cliffs, New Jersey: Prentice Hall.

Kouichi, K. (1966), "A planning of the forest road network by the theory of graphs", Faculty of Agriculture, Kyoto University, Kyoto. Journal of the Japanese Forestry Society 48(10)'66:365-371.

Lee, S.M. (1972), "Goal programming for decision analysis", Auerbach Publishers, Inc., Philadelphia.

Lindo Systems, Inc. (1999), "Lingo", Premier Modeling Tools.

Lockwood, C., and Moore, T. (1993), "Harvest scheduling with spatial constraints: A simulated annealing approach", Canadian Journal of Forest Research, Vol. 23, No. 3, pp. 468-478.

MacLean, H.R. (1980), "Operations research using reproduction inventory data and linear programming to help select optimum forest regeneration systems for Nipigon District", B.Sc.F. thesis, Lakehead University, Thunder Bay, Ontario.

Mandl, C., (1979), "Applied network optimization", Academic Press, London. p. 175.

Michalewicz, Z. (1996), "Genetic Algorithms + Data Structures = Evolution Programs", Springer-Verlag, New York. p.387.

Minamikata, Y. (1984), "Effective forest road planning for forest operations and the environment", In COFE/IUFRO (15-18 Aug. 1984, Fredericton, NB, Canada), pp. 219-224.

Murray, A.T., Church, R.L. (1996a), "Constructing and selecting adjacency constraints", INFOR, Vol. 34, pp. 232-248.

Murray, A.T., Church, R.L. (1996b), "Analysing cliques for imposing adjacency restrictions in forest models", For. Sci. 42(2), 166-175.

Navon, D.I. (1971), "Timber RAM... A long-range planning method for commercial timber lands under multiple-use management", USDA Forest Service, Research Paper PSW-70, Berkeley, California, 22pp.

Nelson, J., Brodie, D. (1990), "Comparison of a random search algorithm and mixed integer programming for solving area-based forest plans", Canadian Journal of Forest Research 20, 934-942.

Nelson, J.D., and Finn, S.T. (1991), "The influence of cut block size and adjacency rules on harvest levels and road networks", Can. J. For. Res. 21:595-600.

Pnevmaticos, S.M., and Mann, S.H. (1972), "Dynamic programming in tree bucking", Forest Products Journal 22(2), 26-30.

Raiffa, H. (1968), "Decision analysis", Addison-Wesley, Reading, MA. 569 p.

Richardson, J.T., Palmer, M.R., Liepins, G., and Hilliard, M. (1989), "Some guidelines for genetic algorithms with penalty functions. In Schaffer, J.D., editor, Proc. 3rd Int. Conf. On Genetic Algorithms, pp. 191-197. Morgan Kaufmann.

Romero, C. (1985), "Naïve weighting in non-preemptive goal programming. J. Oper. Res. Soc. Am. 7:647-649.

Rustagi, K.P., and Bare, B.B. (1987), "Resolving multiple goal conflicts with interactive goal programming", Canadian Journal of Forest Research, Vol. 17, No. 11, pp. 1401-1407.

Saaty, T.L. (1977), "A scaling method for priorities in hierarchical structures", Journal of Mathematical Psychology 15, 234-281.

Saaty, T.L. (1980), "The analytic hierarchy process. Planning, priority setting, resource allocation", McGraw-Hill.

Saaty, T.L., and Kearns, K.P. (1985), "Analytic planning. The organization of systems", Pergamon Press, Oxford.

Sakawa, M. (1993), "Fuzzy sets and interactive multiobjective optimization", Plenum Press, New York.

Schaffer, J.D. (1985), "Multiple objective optimization with vector evaluated genetic algorithms", Proc. 1st Int. Conf. On Genetic Algorithms (Pittsburgh, PA, July 1985) ed. Grefenstette, J.J. (Hillsdale, NJ:
Erlabaum) pp. 93-100.

Schmidt, J.W. (1981), "Foundations of analysis in operations research", Academic Press, Inc. Ltd., London. p. 383.

Schmidt, J.W., and Tedder, P.L. (1981), "A comprehensive examination of economic harvest optimization simulation methods", Forest Science 27, 523-536.

Schuler, A. and Meadows, J. (1975), "Planning resource use on national forests to achieve multiple objectives", Journal of Environmental Management 3, 351-366.

Srinivas, N., and Deb, K. (1994), "Multiobjective optimization using nondominated sorting in genetic algorithms", Evolutionary Computation 2(3), 221-248.

Torres-Rojo, J.M., and Brodie, J.D. (1990), "Adjacency constraints in harvest scheduling: an aggregation heuristic". Can. J. For. Res. 20: 978-986.

Walker, H.D. (1984), "An alternative approach to goal programming", Can. J. For. Res., Vol. 15, pp. 319-325.

Winston, W.L. (1987), "Operations research: applications and algorithms", 1st ed., PWS-KENT Publishing Company, Boston. p. 1025.

Zeleny, M. (1982), "Multiple criterion decision making", McGraw-Hill, New York.

Zimmermann, H.J. (1976), "Description and optimization of fuzzy systems", Internat. J. General Systems 2, 209-215.

Appendix 1
LINGO Formulation for Mr. Freeman's Problem (unranked goals)

```
! This is Mr. Freeman's model ;

MIN = d1M + d2M + d3M + d4M + d4P + d5M;

! constraints;

!Net income of about $2100 a year :
20* x1 + 15 *x2 + 45* x3 + d1M = 2100;

! availability of the property for his family's 30-day summer
vacation;
x4 + d2M = 30;

! His 7- day fall hunting trip;
x5 + d3M = 7;

! Stability of the harvest (allowable harvest);
x3 + d4M - d4P = 7;

! For safety and aesthetics (working days);
x6 - 1/4* x1 - 1/4* x2 + d5M = 0;

! Summer activities;
x1 + x3 + x4 + 0.5* x6 <= 90;

! Fall activities;
x2 + x5 + 0.5* x6 <= 60;
! End;
```

Appendix 2
Output of LINGO Model of Mr. Freeman's (unranked goals)

" The output of the linggo solution of Mr. Freeman's forestry problem without priority goals"

```
 Rows=       8 Vars=      16 No. integer vars=       0  ( all are linear)
 Nonzeros=     38 Constraint nonz=    26(     20 are +- 1)
Density=0.279
 Smallest and largest elements in abs value=   0.500000        2100.00
 No. < :    2 No. =:    5 No. > :    0, Obj=MIN, GUBs <=    4
 Single cols=    4

 Global optimal solution found at step:            14
 Objective value:                            10.56140

         Variable           Value        Reduced Cost
              D1M       0.0000000           0.9649123
              D2M       0.0000000           0.4210526
              D3M       0.0000000           0.5964912
              D4M       0.0000000            2.000000
              D4P        10.56140           0.0000000
              D5M       0.0000000           0.8771930
               X1        32.89474           0.0000000
               X2        43.45614           0.0000000
               X3        17.56140           0.0000000
              D1P       0.0000000         0.3508772E-01
               X4        30.00000           0.0000000
              D2P       0.0000000           0.5789474
               X5        7.000000           0.0000000
              D3P       0.0000000           0.4035088
               X6        19.08772           0.0000000
              D5P       0.0000000           0.1228070

              Row    Slack or Surplus      Dual Price
                1        10.56140            1.000000
                2       0.0000000        -0.3508772E-01
                3       0.0000000          -0.5789474
                4       0.0000000          -0.4035088
                5       0.0000000            1.000000
                6       0.0000000          -0.1228070
                7       0.0000000           0.5789474
                8       0.0000000           0.4035088
```

Appendix 3
LINGO Formulation for Mr. Freeman's Problem (ranked goals)

```
! This is Mr. Freeman's model with third priority goal of
  ;

MIN = 3*d2M + 3*d3M + d5M;
! constraints;

!Net income of about $2100 a year ;
20* x1 + 15 *x2 + 45* x3 + d1M = 2100;

! availability of the property for his family's 30-day summer
vacation;
x4 + d2M = 30;

! His 7- day fall hunting trip;
x5 + d3M = 7;

! Stability of the harvest (allowable harvest);
x3 + d4M - d4P = 7;

! For safety and aesthetics (working days);
x6 - 1/4* x1 - 1/4* x2 + d5M = 0;

! Summer activities;
x1 + x3 + x4 + 0.5* x6 <= 90;

! Fall activities;
x2 + x5 + 0.5* x6 <= 60;

! the result of first and second priority goals as a constraints;
d1M=0;
d4M=0;
d4P=0;
! End;
```

Appendix 4

Output of LINGO Model of Mr. Freeman's (ranked goals)

Global optimal solution found at step: 6

Objective value: 21.50000

variable	value	reduced cost
d1m	0.0000000	0.9649123
d2m	0.0000000	0.4210526
d3m	0.0000000	0.5964912
d4m	0.0000000	2.000000
d4p	10.56140	0.0000000
d5m	0.0000000	0.8771930
x1	32.89474	0.0000000
x2	43.45614	0.0000000
x3	17.56140	0.0000000
d1p	0.0000000	0.3508772e-01
x4	30.00000	0.0000000
d2p	0.0000000	0.5789474
x5	7.000000	0.0000000
d3p	0.0000000	0.4035088
x6	19.08772	0.0000000
d5p	0.0000000	0.1228070

row	slack or surplus	dual price
1	10.56140	1.000000
2	0.0000000	-0.3508772e-01
3	0.0000000	-0.5789474
4	0.0000000	-0.4035088
5	0.0000000	1.000000
6	0.0000000	-0.1228070
7	0.0000000	0.5789474
8	0.0000000	0.4035088

Appendix 5
Matlab Source code for Mr. Freeman's Problem

```
%This is the Mr.Freeman's forestry problem
%File name is Mr_Freeman
function g = forest1(x)
%
if all(x>=0),
    g=zeros(6,1);
    % Compute the objective function
    g(2)=abs(2100-20*x(1)-15*x(2)-45*x(3));
    g(3)=abs(30-x(4));
    g(4)=abs(7-x(5));
    g(5)=abs(7-x(3));
    g(6)=abs(x(1)+x(2)-4*x(6));
    % Compute the Constraints, in the form gconstmp>=0
    gconstmp=90-x(1)-x(3)-x(4)-0.5*x(6);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    gconstmp=60-x(2)-x(5)-0.5*x(6);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end


 else,
    g=inf;
end
```

Appendix 6
Matlab output of Mr. Freeman's problem

Summer Rental	Fall Rental	Harvest Days	Summer Vacation	Fall Hunting	Work Days	Goal 1: Net Income $ 2,100.00	Goal 2 Summer vac 30 days	Goal 3 hunting 7 days	Goal 4 Harvesting 7 days	Goal 5 Rest days 0 days	Const. 1 Goal<=90	Const 2 Goal<=60
LINDO solution:												
59.877	40.458	7	10.58	7	25.083	2119.41	10.58	7	7	-0.003	89.9985	59.9995
MOGA solutions:												
47.3916	38.9254	12.6732	19.3362	6.983	21.0565	2102.007	19.3362	6.983	12.6732	-2.091	89.92925	56.43665
47.6944	42.278	11.5981	19.985	5.7502	20.4278	2109.9725	19.985	5.7502	11.5981	-8.2612	89.4914	58.2421
44.5772	40.1891	12.7847	20.5862	6.7792	20.6313	2069.692	20.5862	6.7792	12.7847	-2.2411	88.26375	57.28395
43.4818	40.7507	12.6274	29.3699	6.8336	6.6939	2049.1295	29.3699	6.8336	12.6274	-57.4569	88.82605	50.93125
47.2215	42.2058	10.0102	19.749	6.9761	16.1846	2027.976	19.749	6.9761	10.0102	-24.6889	85.073	57.2742
44.5538	41.649	10.4484	24.2056	5.5445	17.2229	1985.989	24.2056	5.5445	10.4484	-17.3112	87.81925	55.80495
47.3909	41.4073	8.6138	28.1213	0.8057	4.5734	1956.5485	28.1213	0.8057	8.6138	-70.5046	86.4127	44.4997
42.8597	43.9398	9.0789	26.8974	2.9415	15.668	1924.8415	26.8974	2.9415	9.0789	-24.1275	86.67	54.7153
42.6712	46.9194	7.4701	27.45	7	8.8538	1893.3695	27.45	7	7.4701	-54.1754	82.0182	58.3463
43.2032	44.4848	7.1777	29.7897	6.14	5.8311	1854.3325	29.7897	6.14	7.1777	-64.3636	83.08615	53.54035
42.9519	39.2328	7.0276	29.9175	6.999	9.666	1763.772	29.9175	6.999	7.0276	-43.5207	84.73	51.0648
36.2416	42.3632	7.7234	29.9527	5.187	6.5147	1707.833	29.9527	5.187	7.7234	-52.546	77.17505	50.80755
43.2586	40.844	7.4226	29.9419	7.3364	5.8559	1811.849	29.9419	7.3364	7.4226	-60.679	83.55105	51.10835
43.2032	40.9645	7.1777	29.7897	6.14	5.8311	1801.528	29.7897	6.14	7.1777	-60.8433	83.08615	50.02005
47.2141	41.4073	8.4501	30.4701	6.8163	4.5509	1945.646	30.4701	6.8163	8.4501	-70.4178	88.40975	50.49905
40.1994	40.6295	8.8986	29.2543	5.9404	21.003	1813.8675	29.2543	5.9404	8.8986	3.1831	88.8538	57.0714
43.1798	40.439	7.2876	31.1934	11.0295	12.3927	1798.123	31.1934	11.0295	7.2876	-34.048	87.85715	57.66485
43.2956	44.5595	7.381	28.3176	2.5873	5.8359	1866.4495	28.3176	2.5873	7.381	-64.5115	81.91215	50.06475
42.5713	40.4718	8.8823	27.4983	5.8051	20.6485	1858.2065	27.4983	5.8051	8.8823	-0.4491	89.27615	56.60115
44.0046	40.1891	9.3456	26.7752	5.4001	19.5367	1903.4805	26.7752	5.4001	9.3456	-6.0469	89.89375	55.35755
42.6712	42.2773	9.7122	25.7566	3.4043	20.5194	1924.6325	25.7566	3.4043	9.7122	-2.8709	88.3997	55.9413
43.8668	42.0446	9.8333	25.3422	5.7486	16.1927	1950.5035	25.3422	5.7486	9.8333	-21.1406	87.13865	55.88955
45.3477	41.1919	11.1592	22.8923	5.3086	18.4134	2026.9965	22.8923	5.3086	11.1592	-12.886	88.6059	55.7072
46.0614	39.5169	11.7719	22.026	6.8194	18.6049	2043.717	22.026	6.8194	11.7719	-11.1587	89.16175	55.63875
44.5772	40.1891	9.3456	20.5862	6.7792	20.5405	1914.9325	20.5862	6.7792	9.3456	-2.6043	84.77925	57.23855
45.424	39.0116	12.6539	19.5191	6.7968	20.8173	2063.0795	19.5191	6.7968	12.6539	-1.1664	88.00565	56.21705
45.8053	42.6164	12.8498	20.9715	6.9555	18.1376	2133.593	20.9715	6.9555	12.8498	-15.8713	88.6954	58.6407

41.6524	45.5606	7.3771	19.8274	6.892	5.8256	1848.4265	19.8274	6.892	7.3771	-63.9106	71.7697	55.3654
44.3872	39.0571	12.6437	19.6155	6.6988	20.6912	2042.567	19.6155	6.6988	12.6437	-0.6795	86.992	56.1015
47.2898	38.2831	12.8514	19.4236	6.563	20.8633	2098.3555	19.4236	6.563	12.8514	-2.1197	89.99645	55.27775
43.2303	40.5156	7.3771	29.8533	6.3634	5.8256	1804.3095	29.8533	6.3634	7.3771	-60.4435	83.3735	49.7918
43.2919	40.6444	6.8927	29.3854	6.1237	5.3737	1785.6755	29.3854	6.1237	6.8927	-62.4415	82.25685	49.45495
48.7825	40.0785	12.0095	19.749	5.8115	13.9886	2117.255	19.749	5.8115	12.0095	-32.9066	87.5353	52.8843
42.0278	43.7718	9.3442	29.259	8.6164	4.5689	1917.622	29.259	8.6164	9.3442	-67.524	82.91545	54.67265
43.2366	43.5695	7.5885	30.0639	4.9873	5.9551	1859.757	30.0639	4.9873	7.5885	-62.9857	83.86655	51.53435
43.0693	40.9645	7.6307	30.0948	4.5628	5.9794	1819.235	30.0948	4.5628	7.6307	-60.1162	83.7845	48.517
43.237	43.7192	9.3452	27.8397	3.9224	7.8864	1941.062	27.8397	3.9224	9.3452	-55.4106	84.3651	51.5848
41.8878	40.6369	7.1023	19.8274	3.5115	12.7671	1766.913	19.8274	3.5115	7.1023	-31.4563	75.20105	50.53195
41.3844	40.844	7.4226	29.2725	1.5378	14.04	1774.365	29.2725	1.5378	7.4226	-26.0684	85.0995	49.4018
42.9519	39.12	7.0276	24.9046	6.563	20.5194	1762.08	24.9046	6.563	7.0276	0.0057	85.1438	55.9427
41.6977	44.2902	7.0999	29.2641	9.1816	12.7905	1817.8025	29.2641	9.1816	7.0999	-34.8259	84.45695	59.86705
47.2141	45.1816	7.2378	29.1761	7.0006	5.7701	1947.707	29.1761	7.0006	7.2378	-69.3153	86.51305	55.06725
43.2956	42.278	7.381	28.3176	5.7502	20.4278	1832.227	28.3176	5.7502	7.381	-3.8624	89.2081	58.2421
43.0693	44.4848	7.6307	30.0948	4.5628	5.9794	1872.0395	30.0948	4.5628	7.6307	-63.6365	83.7845	52.0373
43.2586	45.5694	7.8452	29.9419	7.3364	5.8559	1901.747	29.9419	7.3364	7.8452	-65.4044	83.97365	55.83375
41.8639	40.432	8.1528	28.8799	5.7709	20.559	1810.634	28.8799	5.7709	8.1528	-0.0599	89.1761	56.4824
42.9519	39.12	8.5105	19.749	7	20.5168	1828.8105	19.749	7	8.5105	-0.0047	81.4698	56.3784
46.0614	39.5169	9.3442	22.026	6.8194	18.6049	1934.4705	22.026	6.8194	9.3442	-11.1587	86.73405	55.63875
42.7365	42.0215	10.0001	25.4604	3.3723	21.1693	1935.057	25.4604	3.3723	10.0001	-0.0808	88.78165	55.97845
47.1041	39.2544	10.4056	20.0581	6.999	17.6513	1999.15	20.0581	6.999	10.4056	-15.7533	86.39345	55.07905
45.3732	40.6024	11.1358	22.7415	5.8819	16.2881	2017.611	22.7415	5.8819	11.1358	-20.8232	87.39455	54.62835
43.1688	39.2328	11.7321	21.1568	8.8946	20.5194	1979.8125	21.1568	8.8946	11.7321	-0.324	86.3174	58.3871
47.2141	41.4073	12.2753	19.6518	6.8163	20.6565	2117.78	19.6518	6.8163	12.2753	-5.9954	89.46945	58.55185
47.5312	37.7306	12.9805	18.8211	7	21.0083	2100.7055	18.8211	7	12.9805	-1.2286	89.83695	55.23475
47.579	38.6486	12.6457	19.3168	5.9501	20.9167	2100.3655	19.3168	5.9501	12.6457	-2.5608	89.99985	55.05705
47.579	38.6486	12.6457	19.3168	5.9501	20.9167	2100.3655	19.3168	5.9501	12.6457	-2.5608	89.99985	55.05705
47.1111	39.12	12.6607	19.985	7.0006	20.4278	2098.7535	19.985	7.0006	12.6607	-4.5199	89.9707	56.3345
34.4736	40.4025	8.1953	27.9518	7.0006	20.5194	1664.298	27.9518	7.0006	8.1953	7.2015	80.8804	57.6628
43.237	38.5572	7.5643	18.0944	6.6445	16.9402	1783.4915	18.0944	6.6445	7.5643	-14.0334	77.3658	53.6718
48.9727	38.5572	8.7221	18.0944	6.6445	16.9402	1950.3065	18.0944	6.6445	8.7221	-19.7691	84.2593	53.6718
42.1623	44.2715	7.4675	28.0657	3.6537	15.452	1843.356	28.0657	3.6537	7.4675	-24.6258	85.4215	55.6512
43.2784	40.4025	7.5793	27.9518	6.9761	13.9769	1812.674	27.9518	6.9761	7.5793	-27.7733	85.79795	54.36705
41.4971	43.4092	7.7116	29.2627	6.4526	12.6393	1828.102	29.2627	6.4526	7.7116	-34.3491	84.79105	56.18145
47.3423	44.2994	7.1023	19.8274	5.4827	11.986	1930.9405	19.8274	5.4827	7.1023	-43.6977	80.265	55.7751
43.2794	46.9194	12.6296	27.45	6.563	8.8538	2137.711	27.45	6.563	12.6296	-54.7836	87.7859	57.9093
43.2586	45.5694	7.4226	29.9419	7.3364	5.8559	1882.73	29.9419	7.3364	7.4226	-65.4044	83.55105	55.83375

47.2141	39.12	12.6457	19.749	7	20.6313	2100.1385	19.749	7	12.6457	-3.8089	89.92445	56.43565
47.2716	40.0785	12.0095	19.6057	6.1902	20.5981	2087.037	19.6057	6.1902	12.0095	-4.9577	89.18585	56.56775
43.237	41.4073	8.6138	28.1213	6.6445	16.9402	1873.4705	28.1213	6.6445	8.6138	-16.8835	88.4422	56.5219
47.2141	39.12	12.6274	19.6518	7	20.6565	2099.315	19.6518	7	12.6274	-3.7081	89.82155	56.44825
47.1041	39.2544	12.6375	20.0581	6.999	20.3579	2099.5855	20.0581	6.999	12.6375	-4.9269	89.97865	56.43235
43.2794	46.9194	8.5105	27.45	7	8.8538	1952.3515	27.45	7	8.5105	-54.7836	83.6668	58.3463
47.2952	38.5572	9.3456	19.5468	6.9993	20.5405	1944.814	19.5468	6.9993	9.3456	-3.6904	86.45785	55.82675
47.5311	38.5584	12.6891	19.3094	7.0642	20.8763	2100.0075	19.3094	7.0642	12.6891	-2.5843	89.96775	56.06075
42.0397	43.9975	9.4027	29.2543	8.7168	4.2831	1923.878	29.2543	8.7168	9.4027	-68.9048	82.83825	54.85585
47.2141	39.12	12.6355	19.749	6.9993	20.6313	2099.6795	19.749	6.9993	12.6355	-3.8089	89.91425	56.43495
47.2141	39.12	12.6274	19.749	7	20.6313	2099.315	19.749	7	12.6274	-3.8089	89.90615	56.43565
47.3322	38.585	12.7706	19.4886	6.5139	20.7329	2100.096	19.4886	6.5139	12.7706	-2.9856	89.95785	55.46535
47.2201	39.0533	12.6473	19.7953	6.563	20.5468	2099.33	19.7953	6.563	12.6473	-4.0862	89.9361	55.8897
47.2893	38.289	12.8498	19.4265	6.563	20.8609	2098.362	19.4265	6.563	12.8498	-2.1347	89.99605	55.28245
47.278	38.5507	12.7829	19.5405	6.3259	20.6937	2099.051	19.5405	6.3259	12.7829	-3.0539	89.94825	55.22345
47.2893	38.5405	12.8063	19.4265	5.9588	20.7897	2100.177	19.4265	5.9588	12.8063	-2.671	89.91695	54.89415
47.2141	39.12	12.6296	19.8274	6.563	20.5194	2099.414	19.8274	6.563	12.6296	-4.2565	89.9308	55.9427
47.2952	38.5572	12.7847	19.5468	6.9993	20.6313	2099.5735	19.5468	6.9993	12.7847	-3.3272	89.94235	55.87215
47.2261	39.12	12.6274	19.8366	6.8336	20.5014	2099.555	19.8366	6.8336	12.6274	-4.3405	89.9408	56.2043
47.3909	38.289	12.8498	19.296	6.337	20.8609	2100.394	19.296	6.337	12.8498	-2.2363	89.96715	55.05645
47.2141	39.12	12.6274	19.9149	7.0006	20.4464	2099.315	19.9149	7.0006	12.6274	-4.5485	89.9796	56.3438
47.2138	39.12	12.6274	19.9167	7.0043	20.4452	2099.309	19.9167	7.0043	12.6274	-4.553	89.9805	56.3469
47.2555	38.8328	12.7077	19.6458	6.4171	20.5754	2099.4485	19.6458	6.4171	12.7077	-3.7867	89.8967	55.5376
47.2716	38.7005	12.7447	19.6057	6.1902	20.5981	2099.451	19.6057	6.1902	12.7447	-3.5797	89.92105	55.18975
47.1253	39.2285	12.6457	20.0305	7	20.3749	2099.99	20.0305	7	12.6457	-4.8542	89.98895	56.41595
47.1253	39.2285	12.6355	20.0305	6.9993	20.3749	2099.531	20.0305	6.9993	12.6355	-4.8542	89.97875	56.41525
47.4062	38.2118	12.8705	19.2458	6.2909	20.8943	2100.4735	19.2458	6.2909	12.8705	-2.0408	89.96965	54.94985
47.0796	39.12	12.6397	20.09	6.9987	20.3382	2097.1785	20.09	6.9987	12.6397	-4.8468	89.9784	56.2878
46.4017	39.2843	12.6397	20.09	3.894	20.3382	2086.085	20.09	3.894	12.6397	-4.3332	89.3005	53.3474

Appendix 7
MOGAs Solution of Mr.Freeman's Forestry Problem (Rounded to Integer)

No.	Summer Rental	Fall Rental	Harvest Days	Summer Vacation	Fall Hunting	Work Days	Goal 1: Net Income $2,100.00	Goal 2 Summer vac 30 days	Goal 3 hunting 7 days	Goal 4 Harvesting 7 days	Goal 5 Rest days 0 days	Const. 1 Goal<=90	Const 2 Goal<=60
1	51	38	11	22	7	10	2085	22	7	11	-49	89.0	50.0
2	43	43	7	30	5	6	1820	30	5	7	-62	83.0	51.0
3	48	38	13	19	7	21	2115	19	7	13	-2	90.5	55.5
4	42	44	7	29	9	13	1815	29	9	7	-34	84.5	59.5
5	43	39	13	20	7	21	2030	20	7	13	2	86.5	56.5
6	47	39	13	19	7	21	2110	19	7	13	-2	89.5	56.5
7	48	42	12	20	6	20	2130	20	6	12	-10	90.0	58.0
8	45	40	13	21	7	21	2085	21	7	13	-1	89.5	57.5
9	43	41	13	29	7	7	2060	29	7	13	-56	88.5	51.5
10	47	42	10	20	7	16	2020	20	7	10	-25	85.0	57.0
11	45	42	10	24	6	17	1980	24	6	10	-19	87.5	56.5
12	47	41	9	28	1	5	1960	28	1	9	-68	86.5	44.5
13	43	44	9	27	3	16	1925	27	3	9	-23	87.0	55.0
14	43	47	7	27	7	9	1880	27	7	7	-54	81.5	58.5
15	43	44	7	30	6	6	1835	30	6	7	-63	83.0	53.0
16	43	39	7	30	7	10	1760	30	7	7	-42	85.0	51.0
17	36	42	8	30	5	7	1710	30	5	8	-50	77.5	50.5
18	43	41	7	30	7	6	1790	30	7	7	-60	83.0	51.0
19	43	41	7	30	6	6	1790	30	6	7	-60	83.0	50.0
20	47	41	8	30	7	5	1915	30	7	8	-68	87.5	50.5
21	40	41	9	29	6	21	1820	29	6	9	3	88.5	57.5
22	43	40	7	31	11	12	1775	31	11	7	-35	87.0	57.0
23	43	45	7	28	3	6	1850	28	3	7	-64	81.0	51.0
24	43	40	9	27	6	21	1865	27	6	9	1	89.5	56.5
25	44	40	9	27	5	20	1885	27	5	9	-4	90.0	55.0
26	43	42	10	26	3	21	1940	26	3	10	-1	89.5	55.5
27	44	42	10	25	6	16	1960	25	6	10	-22	87.0	56.0
28	45	41	11	23	5	18	2010	23	5	11	-14	88.0	55.0
29	46	40	12	22	7	19	2060	22	7	12	-10	89.5	56.5
30	45	40	9	21	7	21	1905	21	7	9	-1	85.5	57.5
31	45	39	13	20	7	21	2070	20	7	13	0	88.5	56.5
32	46	43	13	21	7	18	2150	21	7	13	-17	89.0	59.0

33	42	46	7	20	7	6	1845	20	7	7	-64	72.0	56.0
34	44	39	13	20	7	21	2050	20	7	13	1	87.5	56.5
35	47	38	13	19	7	21	2095	19	7	13	-1	89.5	55.5
36	43	41	7	30	6	6	1790	30	6	7	-60	83.0	50.0
37	43	41	7	29	6	5	1790	29	6	7	-64	81.5	49.5
38	49	40	12	20	6	14	2120	20	6	12	-33	88.0	53.0
39	42	44	9	29	9	5	1905	29	9	9	-66	82.5	55.5
40	43	44	8	30	5	6	1880	30	5	8	-63	84.0	52.0
41	43	41	8	30	5	6	1835	30	5	8	-60	84.0	49.0
42	43	44	9	28	4	8	1925	28	4	9	-55	84.0	52.0
43	48	48	7	29	7	10	1995	29	7	7	-56	89.0	60.0
44	41	41	7	29	2	14	1750	29	2	7	-26	84.0	50.0
45	43	39	7	25	7	21	1760	25	7	7	2	85.5	56.5
46	42	44	7	29	9	13	1815	29	9	7	-34	84.5	59.5
47	47	45	7	29	7	6	1930	29	7	7	-68	86.0	55.0
48	43	42	7	28	6	20	1805	28	6	7	-5	88.0	58.0
49	43	44	8	30	5	6	1880	30	5	8	-63	84.0	52.0
50	43	46	8	30	7	6	1910	30	7	8	-65	84.0	56.0
51	42	40	8	29	6	21	1800	29	6	8	2	89.5	56.5
52	43	39	9	20	7	21	1850	20	7	9	2	82.5	56.5
53	46	40	9	22	7	19	1925	22	7	9	-10	86.5	56.5
54	43	42	10	25	3	21	1940	25	3	10	-1	88.5	55.5
55	47	39	10	20	7	18	1975	20	7	10	-14	86.0	55.0
56	45	41	11	23	6	16	2010	23	6	11	-22	87.0	55.0
57	43	39	12	21	9	21	1985	21	9	12	2	86.5	58.5
58	47	41	12	20	7	21	2095	20	7	12	-4	89.5	58.5
59	48	38	13	19	7	21	2115	19	7	13	-2	90.5	55.5
60	48	39	13	19	6	21	2130	19	6	13	-3	90.5	55.5
61	48	39	13	19	6	21	2130	19	6	13	-3	90.5	55.5
62	47	39	13	20	7	20	2110	20	7	13	-6	90.0	56.0
63	43	41	8	30	5	18	1835	30	5	8	-12	90.0	55.0
64	43	39	8	18	7	17	1805	18	7	8	-14	77.5	54.5
65	49	39	9	18	7	17	1970	18	7	9	-20	84.5	54.5
66	42	44	7	28	4	15	1815	28	4	7	-26	84.5	55.5
67	43	40	8	28	7	14	1820	28	7	8	-27	86.0	54.0
68	41	43	8	29	6	13	1825	29	6	8	-32	84.5	55.5
69	47	44	7	20	5	12	1915	20	5	7	-43	80.0	55.0
70	43	47	13	27	7	9	2150	27	7	13	-54	87.5	58.5
71	43	46	7	30	7	6	1865	30	7	7	-65	83.0	56.0

72	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
73	47	40	12	20	6	21	2080	20	6	12	-3	89.5	56.5
74	43	41	9	28	7	17	1880	28	7	9	-16	88.5	56.5
75	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
76	47	39	13	20	7	20	2110	20	7	13	-6	90.0	56.0
77	43	47	9	27	7	9	1970	27	7	9	-54	83.5	58.5
78	47	39	9	20	7	21	1930	20	7	9	-2	86.5	56.5
79	48	39	13	19	7	21	2130	19	7	13	-3	90.5	56.5
80	42	44	9	29	9	4	1905	29	9	9	-70	82.0	55.0
81	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
82	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
83	47	39	13	19	7	21	2110	19	7	13	-2	89.5	56.5
84	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
85	47	38	13	19	7	21	2095	19	7	13	-1	89.5	55.5
86	47	39	13	20	6	21	2110	20	6	13	-2	90.5	55.5
87	47	39	13	19	6	21	2110	19	6	13	-2	89.5	55.5
88	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
89	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
90	47	39	13	20	7	21	2110	20	7	13	-2	90.5	56.5
91	47	38	13	19	6	21	2095	19	6	13	-1	89.5	54.5
92	47	39	13	20	7	20	2110	20	7	13	-6	90.0	56.0
93	47	39	13	20	7	20	2110	20	7	13	-6	90.0	56.0
94	47	39	13	20	6	21	2110	20	6	13	-2	90.5	55.5
95	47	39	13	20	6	21	2110	20	6	13	-2	90.5	55.5
96	43	47	13	27	7	9	2150	27	7	13	-54	87.5	58.5
97	43	46	7	30	7	6	1865	30	7	7	-65	83.0	56.0
98	48	39	13	19	7	21	2130	19	7	13	-3	90.5	56.5
99	45	41	11	23	5	21	2010	23	5	11	-2	89.5	56.5
100	42	44	9	29	9	4	1905	29	9	9	-70	82.0	55.0

Appendix 8

Matlab Source code Walker's Problem

```
% Function source code for second problem
function g = walker(x)
%
if all(x>=0),
    g=zeros(4,1);
    % Compute the objective function
    g(2)=-2.7*x(1)-1.6*x(2)-1.0*x(3)-2.3*x(4)-1.4*x(5)-0.6*x(6)-
1.4*x(7)-1.3*x(8)-0.6*x(9)-2.9*x(10)-1.9*x(11)-1.1*x(12)-2.3*x(13)-
1.4*x(14)-0.6*x(15)-1.1*x(16)-1.0*x(17)-0.5*x(18)-3.4*x(19)-2.6*x(20)-
1.1*x(21)-2.9*x(22)-2.1*x(23)-0.8*x(24)-1.4*x(25)-1.0*x(26)-0.5*x(27);
    %v=ones(27,1);
    g(3)=-x(1)-x(2)-x(3)-x(4)-x(5)-x(6)-x(7)-x(8)-x(9)-x(10)-x(11)-
x(12)-x(13)-x(14)-x(15)-x(16)-x(17)-x(18,-x(19)-x(20)-x(21)-x(22)-
x(23)-x(24)-x(25)-x(26)-x(27);

g(4)=350*x(1)+310*x(2)+140*x(3)+350*x(4)+270*x(5)+140*x(6)+270*x(7)+17
0*x(8)+100*x(9)+350*x(10)+310*x(11)+140*x(12)+350*x(13)+270*x(14)+140*
x(15)+270*x(16)+170*x(17)+100*x(18)+310*x(19)+170*x(20)+90*x(21)+310*x
(22)+170*x(23)+90*x(24)+170*x(25)+130*x(26)+90*x(27);
    % Compute the Constraints, in the form gconstmp>=0
    % constraint 1
    gconstmp=800000-350*x(1)-310*x(2)-140*x(3)-350*x(4)-270*x(5)-
140*x(6)-270*x(7)-170*x(8)-100*x(9)-350*x(10)-310*x(11)-140*x(12)-
350*x(13)-270*x(14)-140*x(15)-270*x(16)-170*x(17)-100*x(18)-310*x(19)-
170*x(20)-90*x(21)-310*x(22)-170*x(23)-90*x(24)-170*x(25)-130*x(26)-
90*x(27);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 2
    gconstmp=700-x(1)-x(2)-x(3)-x(4)-x(5)-x(6)-x(7)-x(8)-x(9);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 3
      gconstmp=400-x(10)-x(11)-x(12)-x(13)-x(14)-x(15)-x(16)-x(17)-
x(18);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 4
      gconstmp=400-x(19)-x(20)-x(21)-x(22)-x(23)-x(24)-x(25)-x(26)-
x(27);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 5

gconstmp=2.7*x(1)+1.6*x(2)+1.0*x(3)+2.3*x(4)+1.4*x(5)+0.6*x(6)+1.4*x(7
)+1.3*x(8)+0.6*x(9)+2.9*x(10)+1.9*x(11)+1.1*x(12)+2.3*x(13)+1.4*x(14)+
0.6*x(15)+1.1*x(16)+1.0*x(17)+0.5*x(18)-3300;
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
```

```
    end
    % constraint 6

gconstmp=3.4*x(19)+2.6*x(20)+1.1*x(21)+2.9*x(22)+2.1*x(23)+0.8*x(24)+1
.4*x(25)+1.0*x(26)+0.5*x(27)-1700;
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 7

gconstmp=2.7*x(1)+1.6*x(2)+1.0*x(3)+2.3*x(4)+1.4*x(5)+0.6*x(6)+1.4*x(7
)+1.3*x(8)+0.6*x(9)+2.9*x(10)+1.9*x(11)+1.1*x(12)+2.3*x(13)+1.4*x(14)+
0.6*x(15)+1.1*x(16)+1.0*x(17)+0.5*x(18)+3.4*x(19)+2.6*x(20)+1.1*x(21)+
2.9*x(22)+2.1*x(23)+0.8*x(24)+1.4*x(25)+1.0*x(26)+0.5*x(27)-5500;
     if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 8
    gconstmp=1000-x(1)-x(2)-x(3)-x(10)-x(11)-x(12)-x(19)-x(20)-x(21);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 9
    gconstmp=2200-x(4)-x(5)-x(6)-x(13)-x(14)-x(15)-x(22)-x(23)-x(24);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end
    % constraint 10
    gconstmp=1800-x(7)-x(8)-x(9)-x(16)-x(17)-x(18)-x(25)-x(26)-x(27);
    if gconstmp<0,
       g(1)=g(1)+gconstmp^4;
    end

 else,
    g=inf;
end
```