

Evolutionary programming for mix design

Tatsumi Furuya ^{a,*}, Takaaki Satake ^b, Yoshlyuki Minami ^c

^a *Toho University, Department of Information Science, Miyama 2-2-1, Funabashi-shi, Chiba-ken 274, Japan*

^b *University of Tsukuba, Japan*

^c *Zen-noh, Japan*

Abstract

An evolution method for the solution of a mix design problem is proposed. It is an application of genetic algorithms in which the ratio of ingredients have evolved. In this paper, the evolution algorithm, new crossover methods and mutation schemes are introduced. The usual crossover operation was not used, instead one field of a superior chromosome was copied to an inferior chromosome. A mutation was generated by a combination of uniform random number and normal distribution random number and the mutation rate was very high (1/2). This approach can be applied to non linear optimization problems. The method has been successfully applied to the actual mix problem. © 1997 Elsevier Science B.V.

Keywords: Optimal design; Genetic algorithm; Mix design; Feed

1. Introduction

Mix design problem is a constrained optimization problem to minimize the cost while satisfying constraints. Linear programming methods can be used, if the constraints are given by a set of linear equations. In general, however, the constraints have non linear equations. Also, sometimes there is no solution which satisfies all constraints. In order to solve these problems, we propose an evolutionary programming method.

* Corresponding author. Tel.: + 81 474 728337; e-mail: furuya@toho-u.ac.jp

When chromosomes consist of real-valued genes, it is difficult to obtain an optimum solution. Though many ideas are proposed in evolutionary algorithms, the search space is too large to obtain a solution within a reasonable time period (Back et al., 1997). Fortunately, the mix design problem has a constraint that drastically shrinks the search space. We made use of this characteristic and devised breeding operations (crossover and mutation).

In this paper, first the mix design problem is introduced, then the evolutionary algorithm is shown and last evaluations results are reported.

2. Mix design problem and genetic algorithm

The mix design problem is a constraints optimization problem. The constraints optimization problem is given by free search space, an objective function to be optimized and some constraints to be satisfied.

In our mixed feed design problem, we search a vector for the ratios of the ingredients $\langle X_1, X_2, \dots, X_n \rangle$ which minimize the production cost. This vector is a chromosome. At the same time the solution must satisfy its constraints (nutritive ratio). The following equations show the objective function and constraints with n ingredients and m nutrients. Objective function to be minimized:

$$\sum C_i X_i$$

where C_i is the cost of the ingredient i and X_i is the ratio (percentage) of ingredient i . Constraints:

$$\sum B_{ji} X_i \leq A_j$$

$$\sum B_{mi} X_i \leq A_m$$

where B_{ji} is the content of the nutrient j in the ingredient i and A_j is the range of the nutrient

$$\min X_i \leq X_i \leq \max X_i$$

$$X_1 + X_2 + \dots + X_n = 100$$

The last constraint (100% constraint) plays the most important role to shrink the search space. Though the above constraints are linear constraints, no problem occurs if the constraints include non-linear functions.

2.1. Algorithm

The evolution cycle is represented as follows:

```
initialization;  
while termination do  
  Evaluate;  
  Recombination*;  
  Mutate;  
  Arrange 100%;
```

The devised points of the above basic operations are as follows:

(1) Evaluation: Fitness values for each individual are calculated by the use of cost function (objective function). Then whether each constraint is satisfied or not is checked. If the chromosome doesn't satisfy the constraint a penalty value is added to the cost value. The population is classified into three classes (good, average, poor) based on the fitness value. We also used the elite method.

(2) Recombination*: Our recombination* is a mixture of cross over and recombination*. Two chromosomes are picked at random. A random 'cut point' is chosen to divide both chromosomes. The cut part of the better parent is copied to the worse parent's chromosome. Namely, the chromosome of the better parent is not changed. One elite (best fitness valued) chromosome is kept. If both parents belong to the poor group, then the recombination* is done between the elite chromosome and poor chromosomes. This method accelerates the evolution.

(3) Mutation: A high mutation rate is required. This is shown in the following section.

(4) Arrange 100%: The summation of each mixture rate should be 100%. In the recombination* operation, one part of a chromosome is concatenated to one part of other chromosome. So the summation of mixture rate of the new chromosome will not be 100%. This 'arrange 100%' part forces the summation (changes values of some genes to satisfy the 100%) to 100%. The search space is drastically shrunk by this operation see Section 3.

3. Evaluation

In order to evaluate this evolutionary programming, a feed mixture for livestock is examined. The feed is produced by the use of 20 materials to satisfy the nutrients. Table 1 shows the nutrients in the ingredients and the constraints used in the evaluation. This table is almost the same as used in the real formula feeds problem, however, some parts has been changed to obtain an answer by the simplex method. The simplex method results are used in evaluation of the performance of the evolutionary programming. They are also compared with the results obtained from the genetic method.

Table 1
Nutriment table of ingredients

Material	Nutrition component											Min.	Max.	Cost (10 \$/t)		
	Protein	Fat	Fiber	Ash	DCP	TDN	Ca	P	Lysine	Methionine	+Cystine				Tryptoph	Ca/P
Corn	8.8	3.9	1.9	1.2	6.7	30.7	0.03	0.31	0.24	0.15	0.32	0.1	0.09	20	F	22
Grain sorghum	9	3.1	1.9	1.5	6.7	80.6	0.03	0.3	0.2	0.13	0.27	0.1	0.1	F	50	21
Barley	10.6	2.1	4.4	2.3	8.1	70.4	0.07	0.38	0.37	0.15	0.33	0.11	0.18	F	30	28
Wheat bran	15.4	4.1	9.1	5	11.7	58.9	0.13	1.1	0.61	0.18	0.43	0.2	0.11	F	5	22
Defatted rice bran	17.7	1.9	8.5	11.7	12.6	52.4	0.07	2.82	0.87	0.3	0.58	0.19	0.02	F	5	20
Soybean meal	46.1	1.3	5.6	5.9	40.6	71	0.33	0.7	2.91	0.52	1.22	0.64	0.47	F	20	40
Rapeseed meal	37.1	2.2	9.7	6.4	29.3	59.9	0.71	1.25	2	0.69	1.39	0.44	0.56	F	5	30
Alfalfa meal	15.9	2.3	24.3	10.2	7.6	35.9	1.3	0.24	0.66	0.17	0.31	0.25	5.41	F	5	27
Corn gluten feed	19.8	2.4	8.2	5.4	14.5	57.9	0.25	1	0.69	0.31	0.61	0.12	0.25	F	5	20
Meat and bone meal	50.4	10.6	1.6	30.7	41.8	63.1	11.36	5.74	2.54	0.71	1.12	0.27	1	F	5	60
Animal fat	0	99	0	0	0	213.8	0	0	0	0	0	0	0	0	F	362
Calcium carbonate	0	0	0	99.5	0	0	38.73	0.01	0	0	0	0	3873	F	10	9
Calcium phosphate	0	0	0	99.5	0	0	31.83	18.47	0	0	0	0	1.72	F	3	70
Salt	0	0	0	98	0	0	0.03	0	0	0	0	0	0	0.25	F	40
L-lysine	93	0	0	1	93	93	0	0	0	0	0	0	0	F	F	400
Methionine	57.5	0	0	1	57.5	57.5	0	0	0	0	0	0	0	F	F	500
Molasses	3.1	0.5	0	8.3	1.3	59	1.19	0.11	0.01	0.01	0.03	0.02	10.81	F	5	23
Fish meal	61.2	9.3	0.6	18.7	53.2	69.6	5.69	3.24	4.18	1.53	2.12	0.67	1.75	F	3	80
Cassava tubers	2.6	0.6	4.2	5.8	1.8	79.3	0.26	0.11	0.12	0.03	0.06	0.03	2.36	F	20	18
Vitamin mineral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	450
Max.	F	F	5.5	8	F	F	1	F	F	F	F	F	1.5			
Min.	16	2.5	F	F	13.5	76	0.55	0.5	0.8	0.2	0.5	0.15	1			

F, free.

Table 2
Generations to a solution

Population	500	1000	1500	2000
Generations	5406 (3/5)	5098 (7/10)	5519 (4/5)	5141 (7/10)

Values in parenthesis refer to the rate that can get solution within 10 000 generations.

Besides section Section 2 algorithm, various algorithms such as normal cross over, various mutation rates, etc. are examined. But other than the section Section 2 algorithm could not obtain the optimum solution. Table 2 shows the average number of generations to get an optimum solution and the ratios of solutions obtained within 10 000 generations. Fig. 1 shows the effect of the arrange 100% operation. The objective function $10\,000/xy$ is to be minimized. The constraints and the number of generations to reach the optimum solution are shown in the figure. The generation number in Fig. 1(a) is about ten times smaller than in Fig. 1(b). As shown in the figure, if there is a 100% constraint, all chromosomes are always on the 100% line. Namely, if there is a 100% constraint, search space is limited to only the 100% line.

Randomly selected genes are mutated by the random value generated by normal distribution or uniform distribution. Fig. 2 shows the mutation rate necessary to reach the optimum solution. The size of the circle shows the probability of getting the solution. The results show that the mutation rate (percentage of chromosomes which is mutated) should be very high (50% = 1/2), the normal distribution random number should use a high ratio and the variance of normal distribution should be small. Fig. 3 shows the convergence of two different objective functions (linear

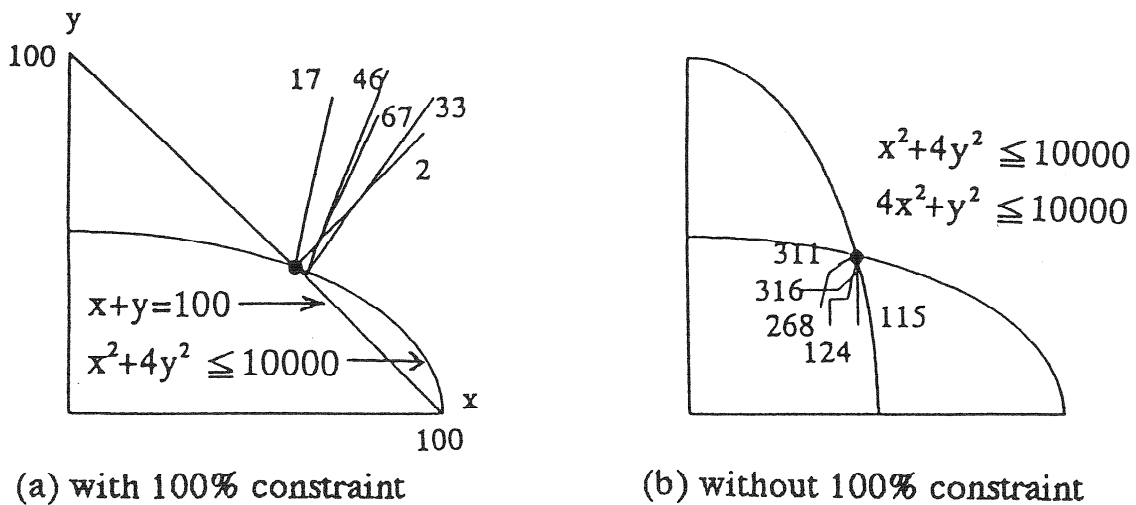


Fig. 1. Effect of 100% constraint.

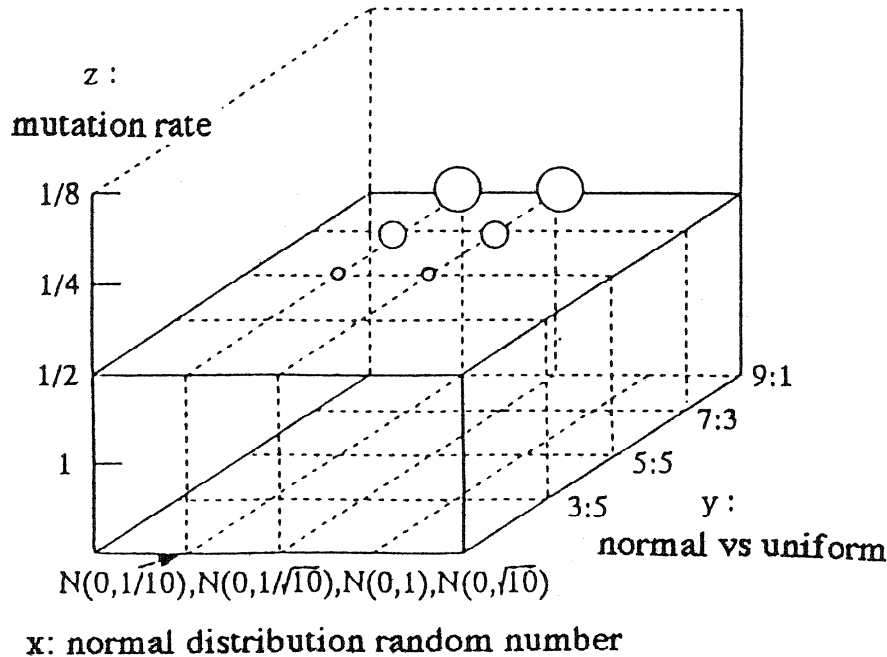


Fig. 2. Effect of mutation.

objective function and non-linear objective function). The non-linear cost is shown in Fig. 3(a). Fig. 3(b) shows that the convergence characteristics are almost the same in both functions. The time required for convergence was 2 or 3 h when using a workstation. On the other hand, if there is a solution which satisfies all constraints and if the constraints are all linear, it takes about 7 min by the simplex method with a personal computer.

4. Conclusion

An evolutionary program for formula feeds design is proposed. By the method, a near optimum solution can be obtained even in the following situations: (a) if the constraints have non-linear functions and (b) if there is no solution which satisfies all constraints.

It is said that the evolutionary program is not easy to apply to real world applications. However, by the use of 100% constraints the search space shrinks drastically and this enables the problem to be solved. Other proposed methods, such as recombination* and mutation parameters will be applicable to other applications.

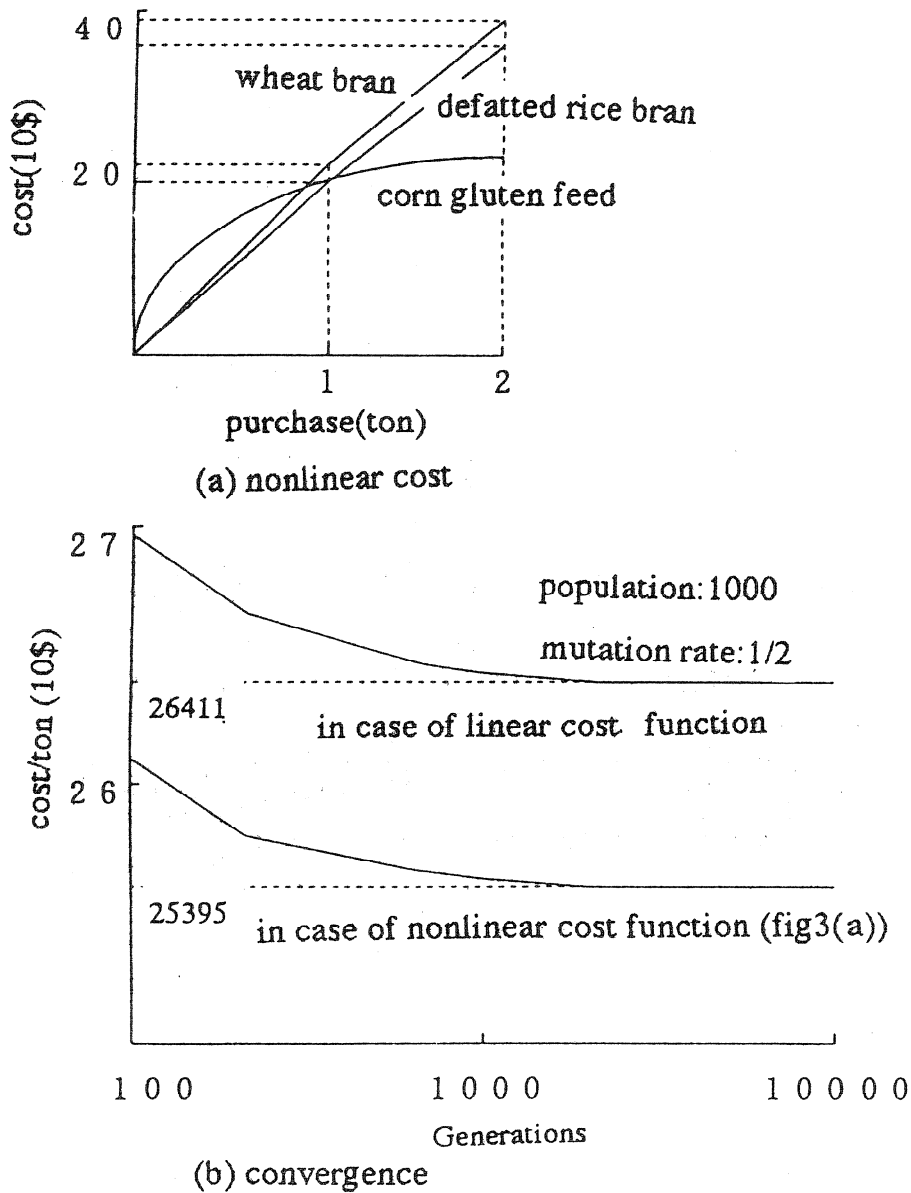


Fig. 3. Convergence to the solution.

References

Back, T., Rudolph, G. and Schwefel, H., 1993. Evolutionary programming and evolution strategies. Proc. 2nd Annual Conference on Evolutionary Programming, Evolutionary Programming Society, San Diego, CA, pp. 11–22.