Journal of the Operations Research Society of Japan
Vol. 43, No. 2, June 2000

GENETIC ALGORITHM IN UNCERTAIN ENVIRONMENTS FOR SOLVING STOCHASTIC PROGRAMMING PROBLEM

Yasunari Yoshitomi
Miyazaki University

Hiroko Ikenoue
Kumamoto National College of Technology

Toshifumi Takeba
Meitec Corp.

Shigeyuki Tomita
Miyazaki University

(Received June 4, 1998; Final August 2, 1999)

Abstract Many real problems with uncertainties may often be formulated as Stochastic Programming Problem. In this study, Genetic Algorithm (GA) which has been recently used for solving mathematical programming problem is expanded for use in uncertain environments. The modified GA is referred as GA in uncertain environments (GAUCE). In the method, the objective function and/or the constraint are fluctuated according to the distribution functions of their stochastic variables. Firstly, the individual with highest frequency through all generations is nominated as the individual associated with the solution presenting the best expected value of objective function. The individual with highest frequency is associated with the solution by GAUCE. The proposed method is applied to Stochastic Optimal Assignment Problem, Stochastic Knapsack Problem and newly formulated Stochastic Image Compression Problem. Then, it has been proved that the solution by GAUCE has excellent agreement with the solution presenting the best expected value of objective function, in cases of both Stochastic Optimal Assignment Problem and Stochastic Knapsack Problem. GAUCE is also successfully applied to Stochastic Image Compression Problem where the coefficients of discrete cosine transformation are treated as stochastic variables.

1. Introduction

Many real problems with some uncertainties may often be formulated as Stochastic Programming Problem [2, 3, 4, 5, 7, 19, 23, 27, 35]. From the points of support for effective decision making in real situations, the stochastic programming problem is very important. However, the problem is generally difficult to solve analytically. We have been longing for some flexible method for solving the problem.

On the other hand, Genetic Algorithm (GA)[12, 16] has been recently applied for solving mathematical programming problem [9, 10, 12, 13, 17, 24, 25, 28, 29, 30]. GA is a kind of solution generator, in addition to solution selector. General GA uses the stationary environment as one fundamental element for selection support, while GA for the nonstationary environment has been recently investigated [6, 8, 14, 18, 34].

Thus, in the present study, for solving Stochastic Programming Problem, the environment in GA is fluctuated through all generations, according to the stochastic distribution functions. For that purpose, Stochastic Optimal Assignment Problem and Stochastic Knapsack Problem [15, 31, 32] are studied first. We can easily generate the solutions for these problems. However, we must check whether our solution for each problem is optimum or not. So, we choose these examples which can be exactly solved with existing method. Then, we apply our method to Stochastic Image Compression Problem as newly formulated problem, because image compression has been recently receiving big attention in the field of computer and information science.

2. GA in Uncertain Environments

We consider the following generalized stochastic programming problem.

$$P_0: \quad Maximize \quad E\{f_0(\boldsymbol{x},\boldsymbol{\xi})\} \tag{2.1}$$

$$Subject\ to \quad P\{f_i(\boldsymbol{x},\boldsymbol{\xi}) \leq 0\} \geq 1-\varepsilon \quad (i=1,2,\cdots,s)$$

$$P\{f_i(\boldsymbol{x},\boldsymbol{\xi}) = 0\} \geq 1-\varepsilon \quad (i=s+1,\cdots,m)$$

$$\boldsymbol{x} \in \boldsymbol{X} \subset \boldsymbol{R}^n$$

where

$\boldsymbol{\xi}$: random variable vector defined on $\Xi \subset \boldsymbol{R}^n$
f_i : $\boldsymbol{R}^n \times \Xi \rightarrow \boldsymbol{R} \quad (i=0,1,\cdots,m)$
$\boldsymbol{X}$: closed

The expected value should be maximized under some stochastic constraints. We can convert the above formulation to the standard minimization type. However, in relation to GA, we use the above formulation in this study.

On the other hand, GA approach generally assumes the stationary environment for solving a mathematical programming problem. GA generates the random individuals associated with the solutions at first. Then, a survival game is generally to be played to get a final winner, or an acceptable solution for the fixed objective function and the fixed constraint. GA is also a solution generator. The winner is the acceptable solution, while the losers also have the information, namely, the individual objective function values. The environment in GA consists of the fitness function made from the objective function and the constraint.

Then, we treat the stochastic fluctuation of the objective function and/or the constraint in Stochastic Programming Problem as the stochastic fluctuation of fitness function in GA. Since the fitness function literally expresses the fitness of the individual to the environment, the fitness function in GA is fluctuated, according to their stochastic distribution-functions for the stochastic variables. Then, we get frequencies of solutions appearing through all generations.

Figure 1 shows the schematic comparison among Stochastic Programming Problem, ordinary GA and GA in Uncertain Environment (GAUCE). In the stochastic programming problem, each solution can have the distribution of objective function value, because of the stochastic fluctuation of variables in the objective function and/or the constraint. On the other hand, GA may be able to present the optimum value, for example, the maximum in terms of the fitness function value at the final generation. In GA for ordinary mathematical programming problem, each solution has individually the deterministic fitness function value made from the objective function value and the constraint. In comparison between Stochastic Programming Problem and ordinary GA, it is natural that GA can be extended for Stochastic Programming Problem, through fluctuating the fitness function or the environment, according to the stochastic distribution-functions for the variables in the fitness function. In each generation of GAUCE, the fitness function or environment is determined by random number generated according to the stochastic distribution-functions. Eventually, in the way of GAUCE, the frequencies of individuals associated with solutions are investigated through all generations.

Figure 2 shows the flowchart of GAUCE. Firstly, the initial population is randomly generated. Then, at each generation, the fitness function including the constraint is determined by random number generated according to the stochastic distribution-functions for their stochastic variables. Next, the calculation of fitness function value, selection, crossover

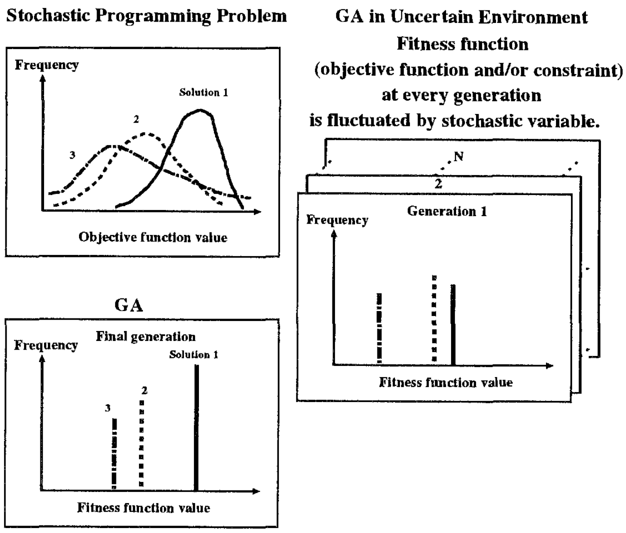


Figure 1: Schematic comparison among Stochastic Programming Problem, ordinary GA and GA in uncertain environment.

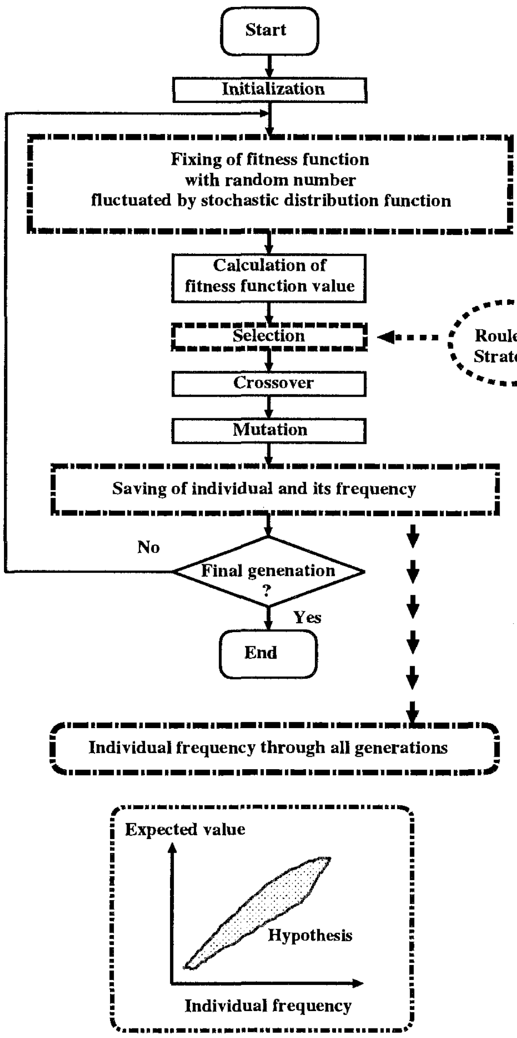


Figure 2: Flowchart of GA in uncertain environment

and mutation are performed as in the usual way in GA. Then, all individuals and their numbers are stored in a computer. These procedures from computing the fitness function to storing the information are performed through all generations. The method provides us the individuals associated with solutions and their frequencies through all generations.

However, for specific stochastic programming problem, the design of GA operator is indispensable. In this study, the objective is to maximize the expected value. From this reason, Roulette Strategy expressed by the following equation is adopted as selection procedure.

$$P(I_i) = \frac{f(I_i)}{\sum_{i=1}^{N} f(I_i)} \tag{2.2}$$

where $f(I_i)$ is the fitness function value for individual i, $P(I_i)$ is the selection probability for individual i.

With Roulette Strategy, the suitable individual for the environment at the generation is selected in proportion to its fitness function value. Moreover, since Roulette Strategy allows sampling with replacement, the selection pressure is relatively high. Therefore, by using Roulette Strategy, it is expected that the higher the expected value is, the higher the individual frequency through all generations is. The hypothesis that the individual with highest frequency through all generations presents the good solution in terms of expected value should be tested by the numerical experiments. If the hypothesis is proved to be valid, we can get the good solution as the individual with highest frequency through all generations.

This method is applied to Stochastic Optimal Assignment Problem, Stochastic Knapsack Problem and newly formulated Stochastic Image Compression Problem.

3. Numerical Experiment and Discussion

3.1. Stochastic optimal assignment problem

The Stochastic Optimal Assignment Problem is formulated as follows.

$$P_1: \quad Maximize \quad E\left(\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}\right) \tag{3.1}$$

$$Subject\ to \quad \sum_{j=1}^{m} x_{ij} = 1 \quad (i = 1, \cdots, n)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, \cdots, m)$$

$$x_{ij} \in \{0, 1\}$$

$$(i = 1, \cdots, n, j = 1, \cdots, m)$$

where c_{ij} are stochastic variables. When a person i is assigned to a job j, x_{ij} equals 1. Elsewhere, x_{ij} equals 0. In fact this problem can be solved as deterministic Integer Programming Problem. However, the reason why this simple problem is selected is that we can evaluate the performance of the proposed method by comparing exact solutions with those obtained by the proposed method.

As GA strategy, ordinal representation for genotype is used to avoid the formation of fatal gene. Then, one point crossover and one point mutation are used. As the fitness function, the objective function is used.

The example of gene phenotype, job number and gene genotype by ordinary representation are shown in Figure 3. In this case, the first person is assigned to the second job, the second person is assigned to the third job, then, the third person is assigned to the first job. In genotype expression for the individual, the first person is assigned to the second job, the second person is assigned to the second job in the remainders, then the third person is assigned to the first job in the remainder.

Phenotype : 010 001 100
Job number: 2 3 1
Genotype : 2 2 1

Figure 3: An example of phenotype, job number and genotype.

The one point crossover is performed for a pair of parent-individuals with certain probability. The division point is randomly selected. At the division point, the chromosome is divided into two parts, and then each part of the chromosome is combined with that from different parent for getting a new pair of child-individuals.

Then, one point mutation is also performed with certain probability. The mutation gene is randomly selected. The allele of the selected gene is changed to one of others.

In this experiment, the population size is 100, the generation size is 1500, the probability of crossover is 0.6, and the probability of mutation is 0.05. When these conditions were used for the ordinary GA, the individual with highest frequency through all generations was the same as the individual associated with optimum solution to the deterministic Optimal Assignment Problem which was previously solved to find out the optimum solution by the branch and bound method.

The transformation from the stochastic problem to the deterministic one at each generation is explained with the simplest example where the stochastic variable has two discrete values, $c^*_{ij} - 0.5$ and $c^*_{ij} + 0.5$, with even probability, where c^*_{ij} are generated according to the uniform distribution over [1,11]. As the first procedure at each generation, the random number is generated for each stochastic variable, according to its distribution function. For this purpose, uniform random number between 0 and 1 is generated.

For every stochastic variable, the generation of random number for not only discrete distribution but also continuous one is independently performed in the usual way[33].

Table 1 shows the summary of numerical results, where the accuracy means the ratio of cases where the solution with highest frequency corresponds to the optimum one. In the small system where $n = m = 3$, six types of stochastic distribution function are adopted, as demonstrated in Table 1. Four types are discrete, while the others are continuous. One of continuous types is the normal distribution. Another is made by combining the two normal distributions with the ratio of 2 to 1. The representative values of $c^*_{ij}, (i = 1,2,3, j = 1,2,3)$ are independently generated according to the uniform distribution over [1,11]. Three examples of each distribution type are generated for testing, where GAUCE is performed 5 times for each example. On the other hand, the expected value is calculated for every case, in order to evaluate the accuracy of this method. Furthermore, the probability that each solution has the highest $\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}$ value among all solutions is calculated for the discrete distribution case. In the case of the discrete distribution, the optimum solution in terms of the expected value corresponds to the solution having the highest value for the probability that the $\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}$ value of the solution is highest among those of all solutions.

Table 1: Summary of numerical results.

Type	Stochastic distribution	Accuracy
D1	Cij[probability]; C*ij-0.5 [0.5], C*ij+0.5 [0.5]	5/5, 3/5, 2/5
D2	Cij[probability]; C*ij-0.5 [0.25], C*ij [0.5], C*ij+0.5 [0.25]	5/5, 5/5, 5/5
D3	Cij[probability]; C*ij-0.5 [0.5 or 0.3], C*ij [0.5], C*ij+0.5 [0.3 or 0.5]	5/5, 5/5, 5/5
D4	Cij[probability]; C*ij-1.0 [0.125], C*ij-0.5 [0.25 or 0.375], C*ij [0.125], C*ij+0.5 [0.375 or 0.25], C*ij+1.0 [0.125]	5/5, 5/5, 5/5
C1	Normal distribution with the average of C*ij	1/5, 4/5, 5/5
C2	Combination of two normal distributions with each average of C*ij-1.0 or C*ij+1 and the ratio of 1 to 2 or 2 to 1	5/5, 4/5, 5/5

C*ij : value generated with uniform random number of 1 to 11

As shown in Table 1, in almost all cases, this method gives the optimum solution as the individual with highest frequency. However, when the expected value difference between the highest and the second highest is less than 2.5 %, the solution with the second highest expected value happens to be selected as the individual with highest frequency in some cases. In these cases, the true optimum solution has the second highest frequency. Figures 4 and 5 show the relationship between the normalized frequency and the normalized expected value, as the typical examples of numerical results for the discrete and continuous distributions respectively. As shown in the typical example in Figures 4 and 5, this method gives the optimum solution as the individual with highest frequency.

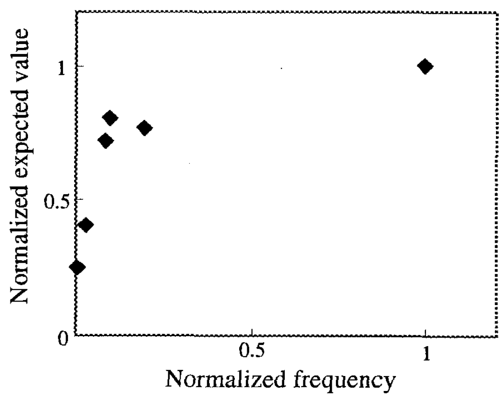


Figure 4: An example of relationship between the normalized frequency and the normalized expected value. Stochastic distribution; D4(C_{ij} has 5 discrete values) in Table 1.

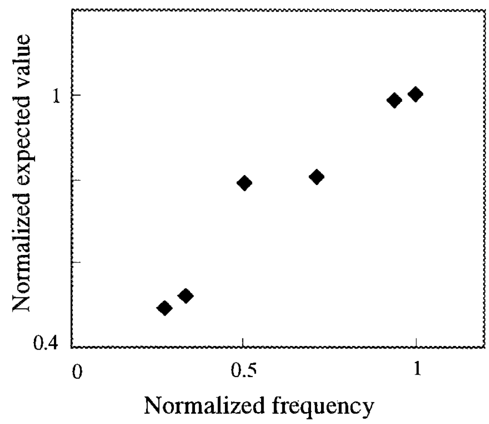


Figure 5: Another example of relationship between the normalized frequency and the normalized expected value. Stochastic distribution; C2(C_{ij} has the combined normal distribution) in Table 1.

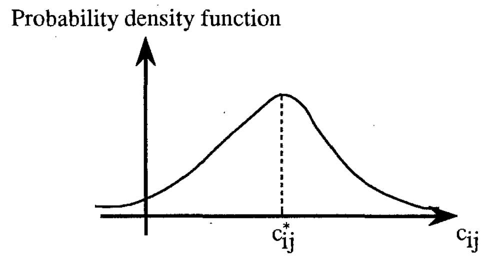
Figure 6 shows an example of numerical result for bigger system where $n = m = 10$ and the normal distribution are used. In this case, the most suitable job for each person is randomly selected, and the value of c^*_{ij} for the most suitable job is 30 which is much higher than that for others. The c^*_{ij} value for others is generated with the uniform distribution over [5,11]. Moreover, the population size increases to 2500 because of the bigger system. In this case, this method also gives the optimum solution as the solution associated with the individual with highest frequency.

Next, we try to find out an additional merit of this method. In this case, the solution rank decision is selected as the first trial. Figure 7 shows the example of rank decision. The gene of the solution with highest frequency in the first GAUCE is treated as the fatal gene in the second GAUCE. Next, the solution with highest frequency in the second GAUCE is selected among all remainders. Then, the gene of the solution with highest frequency in the second GAUCE is also treated as the fatal gene in the third GAUCE. All ranks are decided in the same way in order. In some cases, the optimum solution which may often be selected as the solution with highest frequency in the first GAUCE may be not realistic by some reason. In such a case, the ranking may be useful. The ranking ability is considered to be one advantage of this method.

The relation between the generation size and the accuracy in the both cases of GAUCE and GA is investigated, where the accuracy means the right answer ratio among 50 numerical experiments. In this case, the discrete stochastic distribution of 2 values with even probability is used. In the case of GAUCE, the quality of the calculation is judged by how often the solution with highest frequency coincides with the highest expected value. On the other hand, in the case of GA, the quality of the calculation is judged by how often the solution with highest frequency coincides with the highest fitness value. In this case, the accuracy of GAUCE reaches 100 % at 50th generation, while that of GA reaches 100 % at 20th generation. The CPU time required by GAUCE and GA until reaching 100 % accuracy on Sun SPARC station 20 workstation are 0.4 sec. and 0.1 sec. respectively. However, to find out optimum solution with GA, we must perform GA for every case on $c_{ij}, (i = 1, 2, 3, j = 1, 2, 3)$ having discrete stochastic distribution of 2 values with even probability. We have $2^9 = 512$ cases in this problem. Therefore, when the calculation time of GAUCE is compared with the total calculation time for GA performed for all cases, the

1) Condition

Normal distribution where the average and the standard deviation are C*ij.



C*ij values

i \ j	1	2	3	4	5	6	7	8	9	10
1	6.321	5.044	7.161	6.824	9.124	30.000	11.000	7.494	7.025	8.501
2	5.864	30.000	6.800	6.469	9.913	6.120	5.896	7.129	5.395	9.811
3	6.526	9.891	7.517	30.000	10.636	9.487	6.318	6.539	10.586	5.393
4	6.000	7.200	5.684	8.240	30.000	7.689	7.99	7.431	5.998	9.756
5	9.648	30.000	9.213	7.310	9.055	8.588	9.515	10.295	8.413	9.579
6	8.555	7.479	5.684	6.273	6.503	8.777	30.000	7.941	8.845	7.804
7	9.466	9.632	8.053	10.731	5.826	10.368	30.000	6.352	9.281	10.715
8	30.000	10.446	8.275	8.285	7.174	6.500	7.541	8.341	10.650	10.711
9	8.539	5.022	6.114	10.576	6.791	30.000	6.683	7.852	7.907	7.533
10	5.662	5.174	30.000	7.832	10.332	9.959	7.974	8.545	10.155	8.713

2) Results

No [Frequency]	Solution	Expected value	(Rank)
1[1327072]	6 2 4 5 8 7 10 1 9 3	238.917	(1)
2[637039]	6 2 4 5 8 7 9 1 10 3	237.109	(5)
3[163385]	6 2 4 5 9 7 10 1 8 3	236.980	(6)
4[113068]	6 2 4 5 9 7 8 1 10 3	232.298	(28)
5[96101]	6 2 4 5 8 10 7 1 9 3	236.006	(12)
6[95480]	6 2 4 5 10 7 9 1 8 3	236.712	(7)
7[94261]	6 2 4 5 8 7 10 1 3 9	217.279	(54)
8[64666]	6 2 4 5 10 7 8 1 9 3	233.838	(23)
9[59897]	6 2 4 5 8 9 7 1 10 3	236.673	(8)
10[44525]	6 2 4 5 8 7 3 1 10 9	216.036	(80)
11[44368]	6 2 4 5 8 7 9 1 3 10	214.403	(141)
12[32841]	6 2 4 5 8 10 9 1 7 3	214.063	(158)
13[31491]	6 2 4 5 8 9 10 1 7 3	216.538	(70)
14[27235]	6 2 4 5 8 7 1 3 10 9	195.724	(436)
∫	∫		∫
38758[1]	1 7 2 4 3 9 5 6 8 10	77.297	(37759)
38759[1]	6 10 3 4 7 8 9 5 2 1	100.163	(32378)
38760[1]	6 2 10 1 7 5 3 9 8 4	121.798	(23947)
38761[1]	6 10 3 7 2 5 1 9 8 4	127.622	(18824)
38762[1]	3 7 6 5 10 9 1 2 8 4	106.564	(27925)
38763[1]	6 3 4 9 10 5 8 1 7 2	137.089	(15756)

Figure 6: An example of numerical result for bigger system where $n = m = 10$ and normal distribution are used.

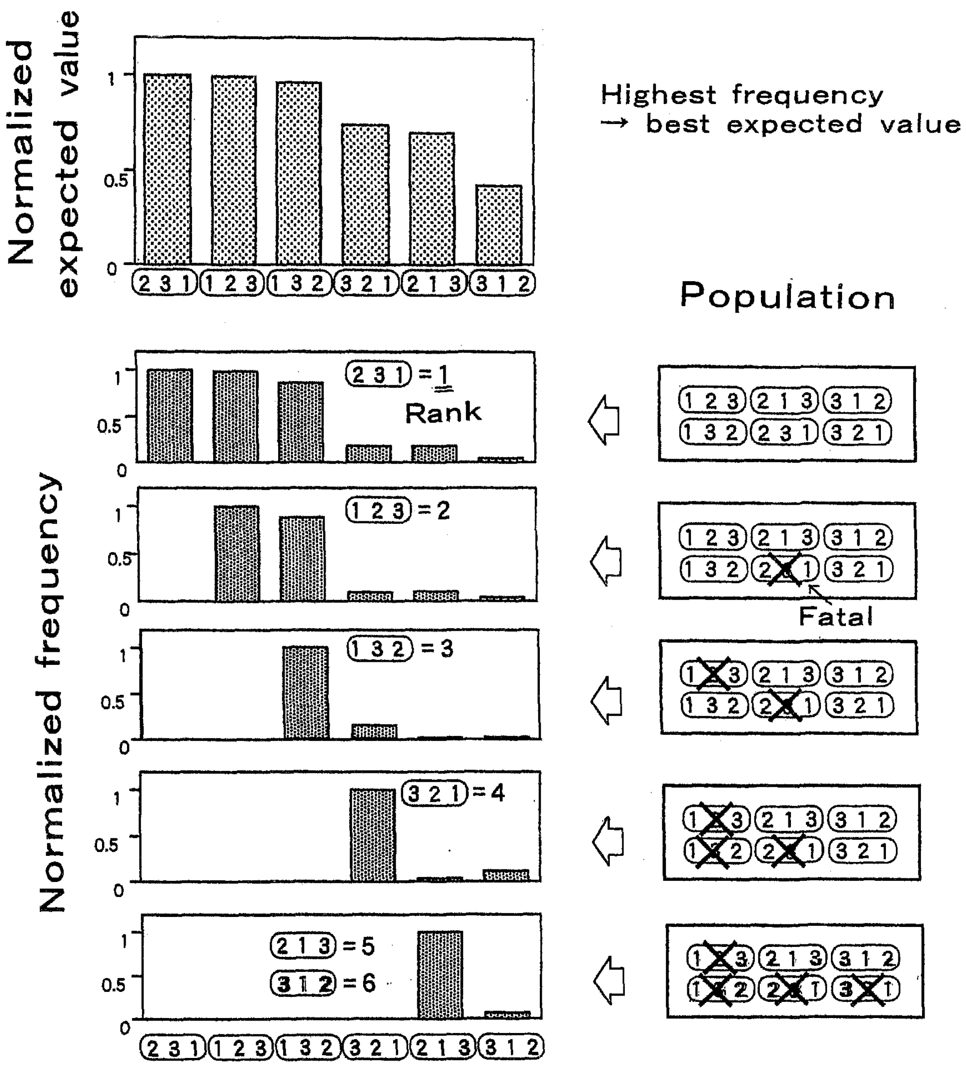


Figure 7: An example of order decision, where stochastic distribution; D4(C_{ij} has 5 discrete values) in Table 1.

calculation time for GAUCE is less than 1 % of that for GA. Furthermore, when the number of cases is increased, the time shortening by GAUCE is considered to be more remarkable. When c_{ij} have continuous stochastic distribution-functions, the calculation time for GA for all cases is infinite because the number of cases is infinite. Therefore, ordinary GA is not appropriate for solving Stochastic Programming Problem.

Since the expected value is maximized in this study, the Roulette strategy is selected as GA operation. As a result, the solution with highest frequency equals the optimum solution in almost all cases. When the expected value difference between the highest and the second highest is less than, for example, 2.5%, the solution with second highest expected value happens to be selected as the solution with highest frequency in some cases. Moreover, in rare cases, the locally optimum solution is selected as the solution with highest frequency. However, even in these two kinds of unfavorable cases, we may be able to get the optimum solution if we perform GAUCE several times, for example, 2 to 5 times. For the most strict usage, the optimum solution may be obtained through calculating the expected values of the high rank solutions through 2 to 5 times calculations of GAUCE. Moreover, the additional advantage is the ability of rank decision of solution. This ranking may be useful in some cases where the optimum solution is not realistic by some practical reason. The calculation time for GAUCE is much less than that for ordinary GA applied to all cases.

3.2. Stochastic knapsack problem

The next example is Stochastic Knapsack Programming Problem formulated as follows.

$$P_2: \quad \textit{Maximize} \quad E\left(\sum_{i=1}^{n} c_i x_i\right) \tag{3.2}$$

$$\textit{Subject to} \quad P\left(\sum_{i=1}^{n} a_i x_i \le b\right) \ge 1-\varepsilon$$

$$x_i \in \{0,1\} \quad (i=1,\cdots,n)$$

where a_i and / or c_i are stochastic variables. In this example, the objective function is also the maximization of the expected value under some stochastic constraints. In fact the expected value is calculated as the constrained expected value.

As GA strategy, genotype equals phenotype. As the fitness function, f, the following equation is used.

$$\begin{aligned} f &= \sum_{i=1}^{n} c_i x_i \quad \text{if} \quad (\sum_{i=1}^{n} a_i x_i \le b) \\ &= 0 \quad \text{if} \quad (\sum_{i=1}^{n} a_i x_i > b) \end{aligned} \tag{3.3}$$

Two points crossover is performed with a pair of parent-individuals with certain probability. The division point is randomly selected. At the division points, the chromosome is divided into three parts, and then each part of the chromosome is combined with that from different parent, for getting a pair of child-individuals. Then, one point mutation is performed with certain probability. The mutation gene is randomly selected. The allele of the selected gene is changed to another.

Table 2 shows the representative values for stochastic variables, where the number of variables, x_i, is 8. These values are used as, for example, the average value of stochastic distribution function. As constant b value, 121 is used. In this experiment, the population size is 500, the generation size is 1500, the probability of crossover is 0.6, the probability of mutation is 0.1. When these conditions were used for the ordinary GA, the individual with highest frequency through all generations was the same as the individual associated with optimum solution to the deterministic Knapsack Programming Problem which was previously solved to find out the optimum solution by the branch and bound method.

Table 2: Representative values for c_i,a_i.

i	1	2	3	4	5	6	7	8
c_i	42	12	45	61	89	32	47	88
a_i	39	13	68	20	31	15	71	16

At each generation, the objective function and/or the constraint are determined by random number generated according to the stochastic distribution-functions for the stochastic variables.

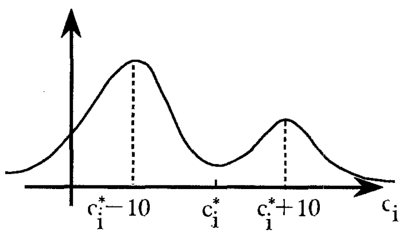
Table 3 shows the summary of numerical results, where the accuracy means the ratio of cases where the solution with highest frequency corresponds to the optimum one. In this numerical experiment, three patterns, (A), (B), and (C), as stochastic conditions are adopted, where (A); the objective function has stochastic variables, (B); the constraint has stochastic variables, (C); both of them have stochastic variables. In the case of (C), the three variables for $c_i(i = 1, \cdots, 8)$ and the three variables for $a_i(i = 1, \cdots, 8)$ are randomly selected, and treated as deterministic parameters for avoiding the trouble from the memory shortage in a computer. Each pattern has some examples of discrete and/or continuous type for stochastic distribution function. One continuous type is normal distribution. Another is made by combining two normal distributions with the rate of 2 to 1. For each distribution, GAUCE is performed 5 times. On the other hand, the expected value is calculated for every case in order to evaluate the accuracy of this method. Furthermore, the probability that each solution has the highest $\sum_{i=1}^{n} c_i x_i$ value among all solutions is calculated for the discrete distribution case. As shown in Table 3, in almost all cases, this method gives the optimum solution as the individual with highest frequency. In the case of OC1 or CC1 in Table 3, where the objective function or the constraint has the stochastic variables with normal distribution, the locally optimum solution happens to be selected as the solution with highest frequency at the ratio of 1/5. In the case of the discrete distribution for both (A) and (B), the optimum solution in terms of the expected value corresponds to the solution having the highest value for the probability that the $\sum_{i=1}^{n} c_i x_i$ value of the solution is highest among those of all solutions. On the other hand, in the case of discrete distribution for (C), the optimum solution in terms of the expected value does not correspond to the solution having the highest value for the probability that the $\sum_{i=1}^{n} c_i x_i$ value of the solution is highest among those of all solutions.

Figures 8, 9 and 10 show the example of numerical results for (A), (B) and (C) respectively. Then, Figures 11, 12 and 13 show the relation between the normalized frequency and the normalized expected value, linked to Figures 8, 9 and 10 respectively. As shown in the typical example in Figures 11, 12 and 13, this method gives the optimum solution as the individual with highest frequency.

1) Condition

Combining two normal distributions with the rate of 2 to 1, where the average and the standard deviation in each normal distribution are C*i - 10 or C*i + 10.

Probability density function



Average value for each normal distribution (Combining ratio)

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
32 (0.667)	2 (0.333)	35 (0.667)	51 (0.667)	79 (0.333)	22 (0.333)	37 (0.333)	78 (0.667)
52 (0.333)	22 (0667)	55 (0.333)	71 (0.333)	99 (0.667)	42 (0.667)	57 (0.667)	98 (0333)

2) Results

(1) True values

No	Solution	Expected value
[1]	10011101	308.66
[2]	11011001	288.66
[3]	01011101	285.34
[4]	10011001	273.32
[5]	00011101	270.00
[6]	11001101	266.34
[7]	10001101	251.00
[8]	01011001	250.00
[9]	11011100	239.34
[10]	00011001	234.66
[11]	11010101	231.66
[12]	11001001	231.00
[13]	01001101	227.68
[14]	00001011	227.34
[15]	10011100	224.00
	⌇	⌇
[99]	00100000	45.00
[99]	00000010	50.34
[100]	00100000	41.66
[101]	10000000	38.66
[102]	00000100	35.34
[103]	01000000	15.34

(2) Numerical results by GAUCE

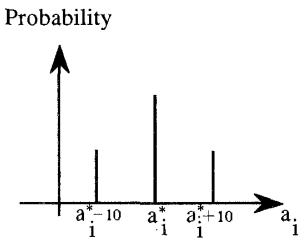
Frequency	Solution
418875	1 0 0 1 1 1 0 1
66499	1 0 0 1 1 0 0 1
57839	1 0 0 0 1 1 0 1
57714	1 0 0 1 0 1 0 1
50862	0 0 0 1 1 1 0 1
9896	1 0 0 0 1 0 0 1
9539	1 1 0 1 0 1 0 1
9414	0 0 0 1 1 0 0 1
9367	1 1 0 1 1 0 0 1
7692	0 1 0 1 1 1 0 1
7549	1 0 0 1 0 0 0 1
7110	0 0 0 1 0 1 0 1
6086	1 1 0 0 1 1 0 1
5822	0 0 0 0 1 1 0 1
5691	1 0 0 0 0 1 0 1
3839	0 1 0 1 1 0 0 1
2656	0 1 0 1 0 1 0 1
2056	1 1 0 0 1 0 0 1
1504	1 1 0 1 0 0 0 1
1366	0 1 0 0 1 1 0 1

(Lower rank solutions are omitted)

Figure 8: A numerical result for the case that c_i has the combined distribution of two normal distribution(OC2 in Table 3).

1) Condition

Three discrete probabilities, 0.25, 0.5, 0.25, where the values of a_i are $a^*_i - 10$, a^*_i, $a^*_i + 10$, respectively.



a_i value
(Probalility)

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
29	3	58	10	21	5	61	6
(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)
39	13	68	20	31	15	71	16
(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)
49	23	78	30	41	25	81	26
(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)	(0.25)

2) Results

(1) True values

No	Solution	Expected value
[1]	00011101	268.945
[2]	01011101	266.578
[3]	01011001	250.000
[4]	10001101	242.176
[5]	10011001	239.531
[6]	00011001	238.000
[7]	11001001	222.879
[8]	10010101	222.129
[9]	01001101	221.000
[10]	10001001	219.000
[11]	00001101	209.000
[12]	11010001	202.207
[13]	11010101	194.609
[14]	10011101	194.391
[15]	01011100	194.000
	⋮	
[172]	11101001	0.270
[173]	11111000	0.243
[174]	11001110	0.217
[175]	11101100	0.215
[176]	11010111	0.069

(2) Numerical results by GAUCE

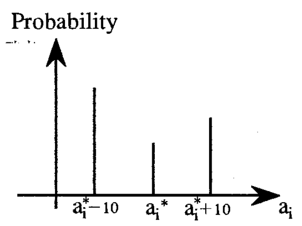
Frequency	Solution
369127	0 0 0 1 1 1 0 1
156950	0 1 0 1 1 1 0 1
73416	0 0 0 1 1 0 0 1
35638	0 1 0 1 1 0 0 1
29038	0 0 0 0 1 1 0 1
19885	0 0 0 1 0 1 0 1
14779	0 1 0 0 1 1 0 1
11068	1 0 0 1 1 1 0 1
9078	0 1 0 1 0 1 0 1
6037	0 0 0 0 1 0 0 1
4044	1 0 0 1 1 0 0 1
3581	0 1 0 0 1 0 0 1
3491	0 0 0 1 0 0 0 1
1770	0 1 0 1 0 0 0 1
1744	1 0 0 0 1 1 0 1
1282	0 0 0 0 0 1 0 1
1129	1 0 0 1 0 1 0 1
839	1 1 0 1 1 1 0 1
759	1 1 0 1 1 0 0 1
684	0 1 0 0 0 1 0 1

(Lower rank solutions are omitted.)

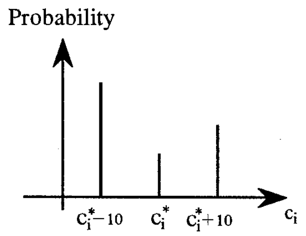
Figure 9: A numerical result for the case that a_i has three values(CD1 in Table 3).

1) Condition

Three discrete probabilities, 0.5(0.3), 0.2, 0.3(0.5), where the values of a_i are a^*_i - 10, a^*_i, a^*_i + 10, respectively.



Three discrete probabilities, 0.5(0.3), 0.2, 0.3(0.5), where the values of c_i are c^*_i - 10, c^*_i, c^*_i + 10, respectively.



i	1	2	3	4	5	6	7	8
a_i (P)		3 (0.5)			21 (0.3)	5 (0.5)	61 (0.5)	6 (0.3)
	39 (1.0)	13 (0.2)	68 (1.0)	20 (1.0)	31 (0.2)	15 (0.2)	71 (0.2)	16 (0.2)
		23 (0.3)			41 (0.5)	25 (0.3)	81 (0.3)	26 (0.5)
c_i (P)	32 (0.5)		35 (0.3)	51 (0.3)	79 (0.5)	22 (0.5)		
	42 (0.2)	12 (1.0)	45 (0.2)	61 (0.2)	89 (0.2)	32 (0.2)	47 (1.0)	88 (1.0)
	52 (0.3)		55 (05)	71 (0.5)	99 (0.3)	42 (0.3)		

2) Results

(1) True values

No	Solution	Expected value
[1]	00011101	268.000
[2]	01011101	260.260
[3]	01011001	250.000
[4]	00011001	238.000
[5]	10001101	226.625
[6]	10010101	221.000
[7]	01001101	217.000
[8]	10001001	215.000
[9]	11001001	209.975
[10]	10011001	208.500
	⁞	
[142]	11000111	8.137
[143]	00011111	7.088
[144]	01011111	3.679

(2) Numerical results by GAUCE

Frequency	Solution
414801	0 0 0 1 1 1 0 1
125424	0 1 0 1 1 1 0 1
77028	0 0 0 1 1 0 0 1
32378	0 0 0 0 1 1 0 1
27070	0 1 0 1 1 0 0 1
21419	0 0 0 1 0 1 0 1
10342	0 1 0 0 1 1 0 1
8692	1 0 0 1 1 1 0 1
7301	0 1 0 1 0 1 0 1
5522	0 0 0 0 1 0 0 1
3750	0 0 0 1 0 0 0 1
2456	1 0 0 1 1 0 0 1
2112	0 1 0 0 1 0 0 1
1467	0 1 0 1 0 0 0 1
1355	0 0 0 0 0 1 0 1

(Lower rank solutions are omitted)

Figure 10: A numerical result for the case that c_i and a_i have three values, respectively(OD2CD2 in Table 3).

Table 3: Summary of numerical results.

Stochastic	Type	Distribution	Accuracy
(A)	OD1	ci[probability]; c*i-10 [0.25], c*i[[0.5], c*i+10 [0.25]	5/5
Objective function	OD2	ci[probability]; c*i-10 [0.5 or 0.3], c*i[[0.2], c*i+10 [0.3 or 0.5]	5/5
	OC1	Normal distribution with the average of c*i	4/5
	OC2	Combination of two normal distributions with each average of c*i-10 or c*i+10 and the ratio of 1 to 2 or 2 to 1	5/5
(B)	CD1	ai[probability]; a*i-10 [0.25], a*i[[0.5], a*i+10 [0.25]	5/5
Constraint	CD2	ai[probability]; a*i-10 [0.5 or 0.3], a*i[[0.2], a*i+10 [0.3 or 0.5]	5/5
	CC1	Normal distribution with the average of a*i	4/5
(C)	OD1CD1	OD1 & CD1	5/5
Objective function & Constraint	OD1CD2	OD1 & CD2	5/5
	OD2CD1	OD2 & CD1	5/5
	OD2CD2	OD2 & CD2	5/5

c*i, a*i : representative value shown in table 2

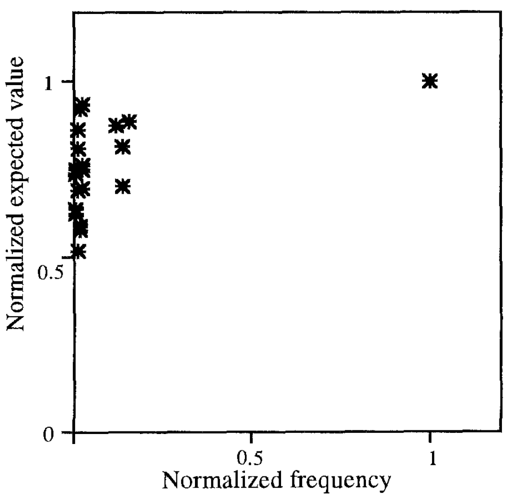


Figure 11: An example of relationship between the normalized frequency and the normalized expected value, linked to Figure 8. Stochastic distribution(OC2 in Table 3); c_i has the combined distribution of two normal distribution.

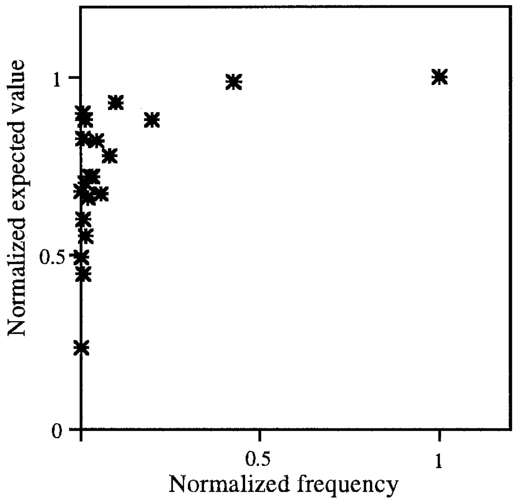


Figure 12: An example of relationship between the normalized frequency and the normalized expected value, linked to Figure 9. Stochastic distribution(CD1 in Table 3); a_i has three values.

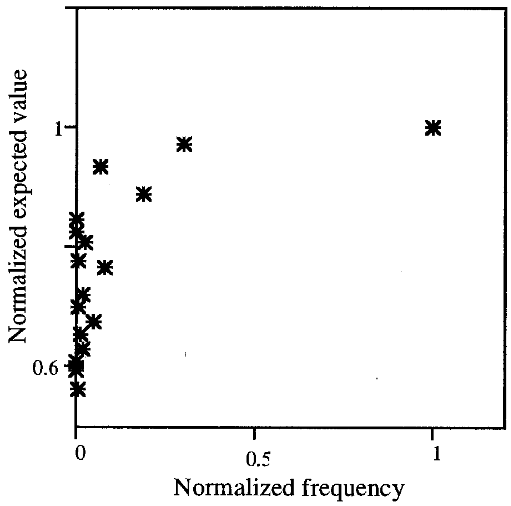


Figure 13: An example of relationship between the normalized frequency and the normalized expected value, linked to Figure 10. Stochastic distribution(OD2CD2 in Table 3); c_i and a_i have three values, respectively.

3.3. Stochastic image compression problem

The next example is Stochastic Image Compression Problem. For simplicity, the image is monochrome and static. The volume of image is compressed with use of optimized condition by this method. When only one image is stored, the image compression is not very meaningful from the point of reducing the volume for storing. However, since many images are stored in almost all cases, the optimization for image compression has been demanded. Namely, the optimization for image compression is for the assembly of images. In this sense, the problem is newly formulated as the Stochastic Programming Problem. Discrete Cosine Transform (DCT) and Huffman coding[38] which are the standard procedure in both Joint Photographic Experts Group (JPEG)[22, 26, 36, 38] and Moving Picture Experts Group (MPEG)[1, 11, 21, 20, 38] are used in this study. The basis of image compression used in this study is described in the references[37, 38].

Firstly, the flowchart of basic image compression used in this study is shown in Figure 14. The left side in Figure 14 is a flow of image compression for getting coded data, while the right side is a flow to get decoded image. As a basic procedure for getting coded data, (1)DCT(8×8 pixel), (2)Quantization, (3)Entropy coding (for direct current element; Huffman coding for difference of DCT coefficient between the DCT block and the next one, for alternating current; Huffman coding for the run length code for DCT coefficient arranged by zigzag scan from the low frequency element to the high one), are performed in order[37]. In (2)Quantization, the DCT coefficient is divided by the product of the coefficient for quantization and the corresponding value in quantization table, and then the divided value is rounded off to get its integer part called a quantized datum. The product is used as the unit-value for quantization. Then, (3)Entropy coding is performed for the quantized data obtained in (2). On the other hand, Entropy decoding, Inverse-quantization and Inverse-DCT (IDCT) are performed in this order for transforming coded data to decoded image. The compression rate is calculated as volume of coded data divided by original image volume. The quantization is approximation and is not reversible. Therefore, we have some error in decoded image inevitably. The condition of quantization is a target of optimization.

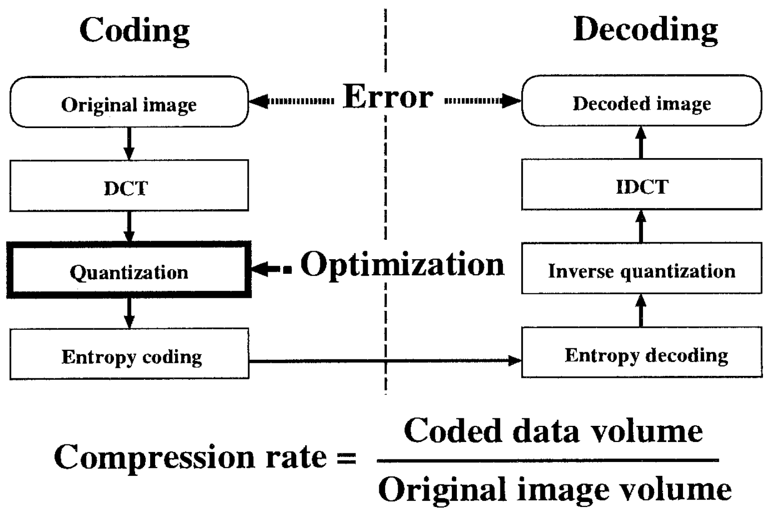


Figure 14: A flowchart of image compression.

The trade-off relationship between the error in decoded image and the compression rate is demonstrated in Figure 15. The quantization is a procedure changing DCT coefficient to the corresponding integer value with unit value. The unit value is the product of the table value and the quantization coefficient. The quantization procedures for certain frequency component (u, v) are demonstrated in Figure 15. In case of bigger unit value, the kind of quantized data is less, and, consequently, the same integer among quantized data appears more frequently. The Entropy coding presents small bit per data to high frequency data. Therefore, in case of bigger unit, coded data has smaller volume. On the contrary, in case of bigger unit which results in rougher approximation in the quantization process, error in decoded image is bigger. The condition of quantization is usually manually adjusted by changing the quantization coefficient. Since image compression has been recently receiving big attention in the field of information science and related business, we have been longing for a method for optimizing the quantization condition.

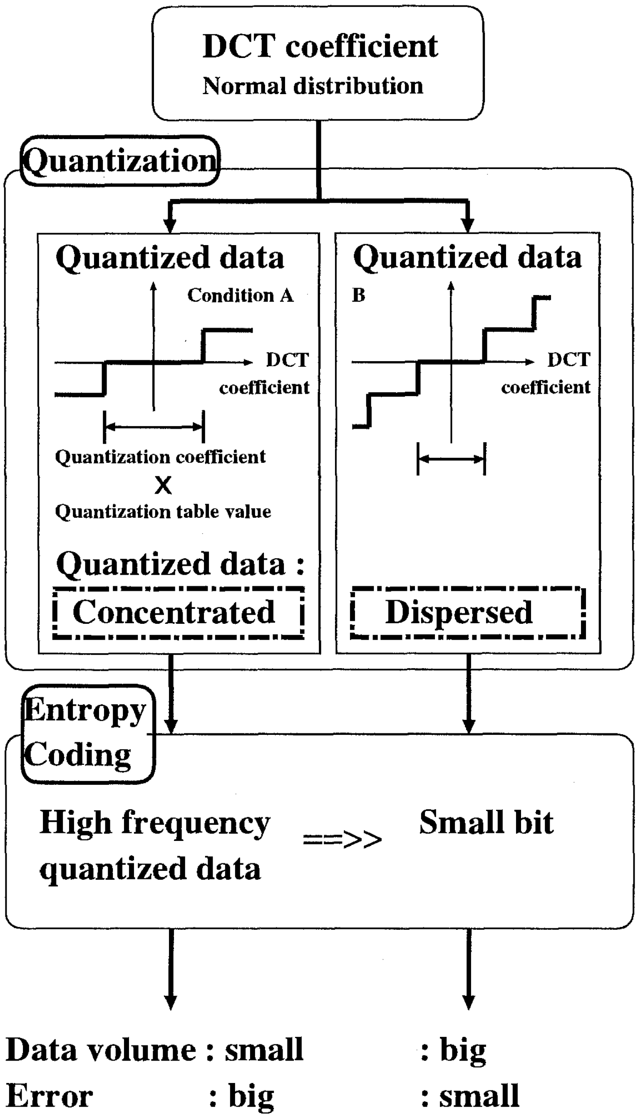


Figure 15: A trade-off relationship between coded data volume and error in decoded image.

It is well-known that the stochastic distribution of DCT coefficient is approximately normal distribution[38]. Therefore, we newly make formula for Stochastic Image Compression Problem using DCT coefficient as stochastic variable. When only one image is stored, the image compression is not very meaningful for reducing volume for storing. However, since many images are stored in almost all cases, the optimization for image compression has been demanded. Namely, the optimization for image compression is for the assembly of images. In this sense the problem is formulated as Stochastic Programming Problem with the approximation that the stochastic distribution function for the DCT coefficient is normal distribution.

The Stochastic Image Compression Problem is formulated as follows.

$$P_3: \quad Maximize \quad E\left(1/e(x_i, F_k(u,v))\right) \tag{3.4}$$

$$Subject\ to \quad P\left(r(x_i, F_k(u,v)) \le b\right) \ge 1-\varepsilon$$

$$x_i \in \{0,1\} \quad (i=1,\cdots,n)\,,\quad (k=1,\cdots,m)\,,\quad (u=0,\cdots,7)\,,\quad (v=0,\cdots,7)$$

$$e(x_i, F_k(u,v)) = \frac{\sum_{k=1}^{m}\sum_{i=0}^{7}\sum_{j=0}^{7}\left|\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)(F_k(u,v) - F_k^*(x_i, F_k(u,v)))cos\frac{(2i+1)u\pi}{16}cos\frac{(2j+1)v\pi}{16}\right|}{\sum_{k=1}^{m}\sum_{i=0}^{7}\sum_{j=0}^{7}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F_k(u,v)cos\frac{(2i+1)u\pi}{16}cos\frac{(2j+1)v\pi}{16}}$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, & (\mathrm{u=0}) \\ 1, & (\mathrm{u \ne 0}) \end{cases}, \qquad C(v) = \begin{cases} \frac{1}{\sqrt{2}}, & (\mathrm{v=0}) \\ 1, & (\mathrm{v \ne 0}) \end{cases}$$

where $e(x_i, F_k(u,v))$ is the error between original image and decoded image, $r(x_i, F_k(u,v))$ is the compression rate obtained as the volume ratio of the original image and the coded data, m is the number of block for DCT, $F_k(u,v)$ is the stochastic variable expressing the coefficient of DCT at block k, $F_k^*(x_i, F_k(u,v))$ is the representative value for quantized $F_k(u,v)$, the $0-1$ variable, x_i, is used for giving the condition for quantization of the DCT coefficient.

In $e(x_i, F_k(u,v))$, the denominator corresponds to the sum total of grey level in original image, and the numerator corresponds to the sum total of difference in grey level at each pixel between original and decoded images. The denominator in $e(x_i, F_k(u,v))$ is for normalization. For calculated gray level in original image , IDCT to $F_k(u,v)$ is performed. Moreover, for calculating gray level in decoded image, the process of quantization, inverse quantization and IDCT is performed.

$F_k^*(x_i, F_k(u,v))$ corresponds to an approximate value for $F_k(u,v)$. The defference between $F_k(u,v)$ and $F_k^*(x_i, F_k(u,v))$ corresponds to error in quantization. When the unit value for quantization is smaller, the error is smaller. The unit value which is decided by x_i values is the product of the table value and the quantization coefficient. On the contrary, as demonstrated in Figure 15, $r(x_i, F_k(u,v))$ is bigger, when the unit value for quantization is smaller. $r(x_i, F_k(u,v))$ is also calculated with the unit value which is decided by x_i values. For calculating coded data volume in $r(x_i, F_k(u,v))$, the process from quantization to entropy coding is performed. Moreover, for calculated original image volume in $r(x_i, F_k(u,v))$, IDCT to $F_k(F_k(u,v))$ is performed.

In this example, the objective function is also the maximization of the expected value under some stochastic constraints. In fact the expected value is also calculated as the constrained expected value, as mentioned in Stochastic Knapsack Problem. Under the constraint on the compression rate, the expected value of inverse value of error between original image and decoded image is maximized. In this problem, both the objective function and the constraint have stochastic variables. The average and the standard deviation of DCT coefficient for every block with 8×8 pixels are measured for the sampling images. The normal distribution of DCT coefficient is determined with the average and the standard deviation of DCT coefficient for the sampling images. Then, the original image is generated with DCT coefficient determined by random number and IDCT to the DCT coefficient. Moreover, the header for encoding is ignored for calculating the compression rate.

At each generation of GAUCE, DCT coefficients are determined by random number. In the generation, the error in decoded image and compression rate are calculated for each individual. As the fitness function value, the inverse value of error between original image and decoded image is used when the constraint on the compression rate is satisfied. On the other hand, the fitness function value is 0 when the constraint on the compression rate is not satisfied.

Here, if we express the table for quantization and the quantization coefficient with 0-1 variables, the number of variables is very big. As a result, the population size and the generation size should be very big for getting the optimum solution. In such a case, it might take long time to calculate the process of GAUCE with a computer. This is mainly because both DCT and IDCT require calculation in the process of GAUCE. Then, for shortening the time for calculation of GAUCE, the condition for quantization of the DCT coefficient is formulated as follows.

$$q(u,v) = \begin{cases} \gamma C d \quad , & \text{(u,v)=(0,0)} \\ \gamma C(a\sqrt{u^2+v^2}+b) \quad , & \text{(u,v)} \neq (0,0) \end{cases} \tag{3.5}$$

$$a = 2^2 x_1 + 2^1 x_2 + 2^0 x_3 + \alpha \tag{3.6}$$

$$b = 2^2 x_4 + 2^1 x_5 + 2^0 x_6 + \beta \tag{3.7}$$

$$C = 2^1 x_7 + 2^0 x_8 + \delta \tag{3.8}$$

where $q(u,v)$ is the product of the coefficient for quantization and the corresponding value for (u,v) in quantization table, and $x_i \in \{0,1\}$, $\alpha, \beta, \gamma, \delta, d$ are positive constant. Here, γC corresponds to the coefficient for quantization and $a\sqrt{u^2+v^2}+b, d$ correspond to the values of quantization table. The formulae expressed by equation (3.5) are supported by the fact that it is difficult for human to detect the high frequency component in the image. Moreover, in equation (3.5), the table value for $(u,v) = (0,0)$ which is direct current element is separated from those for alternating current element because of difference of coding formation between direct and alternating current elements.

The basic part of this problem is the same as Stochastic Knapsack Problem. Namely, 0-1 variables, x_i, should be optimized under some stochastic conditions. As GA strategy, genotype equals phenotype, where the number of variables, x_i, is 8. Moreover, two point crossover and one point mutation are used as in the same way as that mentioned in Stochastic Knapsack Problem. The objective function and/or the constraint are determined at each generation with random number generated according to the stochastic distribution-functions for the stochastic variables.

In this experiment, the population size is 200, the generation size is 100, the probability of crossover is 0.6, the probability of mutation is 0.1. The condition is based on preliminary investigation where ordinary GA was applied to one image.

In each numerical experiment, the original images are TV news with 100 images having 160×120 pixels. The TV images are transformed into monochrome images. The 100 images are stored as the sampling images at the rate of 1 image per 0.1 *sec.*. The average and the standard deviation of DCT coefficient for every block with 8×8 pixels are measured for the sampling images in order to determine normal distribution for each DCT coefficient. The values of constants, i.e., $d = 16, \alpha = \beta = \delta = 1, \gamma = 0.5$, are decided so that the condition expressed by solution includes the region on the standard quantization in JPEG.

Figure 16 shows the numerical results. In the case of $b = 1$, optimum solution is apparently 00000000 which gives the smallest unit value for quantization for DCT coefficient, because the condition of $b = 1$ means no constraint. For other conditions, we can't find out optimum solution analytically.

(1) No constraint

Frequency	Solution
13295	0 0 0 0 0 0 0 0
2023	0 0 0 0 0 1 0 0
1258	0 0 0 0 1 0 0 0
623	0 0 0 1 0 0 0 0
528	0 0 1 0 0 0 0 0
318	0 1 0 0 0 0 0 0
284	1 0 0 0 0 0 0 0
270	0 0 0 0 0 0 1 0
161	0 0 0 0 1 1 0 0
143	0 0 0 1 0 1 0 0

(2) b=0.2

Frequency	Solution	Frequency	Solution	Frequency	Solution
1931	1 0 0 1 0 0 1 0	3128	1 0 0 1 0 0 1 0	1889	1 0 0 1 0 0 1 0
1688	1 0 0 1 1 0 1 0	3015	1 0 0 1 0 1 1 0	1887	1 0 0 1 0 1 1 0
1166	1 0 0 1 0 1 1 0	1676	1 0 0 1 1 1 1 0	1571	1 0 1 1 0 1 1 0
1037	1 0 0 1 1 1 1 0	1437	1 0 1 1 0 1 1 0	1534	1 0 1 1 0 0 1 0
425	1 0 1 1 1 1 1 0	1396	1 0 0 1 1 0 1 0	1032	1 0 1 1 1 0 1 0
420	1 0 1 1 0 0 1 0	949	1 0 1 1 0 0 1 0	860	1 0 0 1 1 0 1 0
357	1 1 0 1 0 0 1 0	948	1 0 1 1 1 1 1 0	837	1 0 1 1 1 1 1 0
354	1 0 1 1 1 0 1 0	645	1 0 1 1 1 0 1 0	644	1 0 0 1 1 1 1 0
338	1 1 0 1 1 0 1 0	557	1 1 0 1 0 1 1 0	536	1 0 1 0 0 1 1 0
250	1 0 1 1 0 1 1 0	435	1 1 0 1 0 0 1 0	531	1 1 1 1 1 1 1 1

Figure 16: Solution with top 10 frequency.

As shown in Figure 16, all x_i values of solution with highest frequency are 0 when no constraint, i.e. $b = 1$, for image compression rate is applied. The numerical result is clearly right. In addition, when the upper limit of image compression is $b = 0.2$, three numerical results give the same solution. Namely, GAUCE presents the same solution as the solution with highest frequency every time. In this case, the optimum solution can not be derived with existing method. However, considering that GAUCE is successfully applied to Stochastic Optimal Assignment Problem and Stochastic Knapsack Programming Problem, GAUCE seems to have the ability for selecting the optimum solution or local optimum one at worst, even when some constraint for image compression rate is applied. Figure 17 shows an example of the original image and the decoded image with this method in case of $b = 0.2$.

Figure 17: An example of application; Left:Original image, Right:Decoded image.

In this study, for simplicity, the image is monochrome and static. The forthcoming paper will present the expanded results for the color and/or dynamic image.

3.4. Other possible applications

This method is applied to Stochastic Optimal Assignment Problem, Stochastic Knapsack Problem and newly formulated Stochastic Image Compression Problem. Moreover, GAUCE may be applied to many Stochastic Programming Problem, for example, Stochastic Scheduling Problem, Stochastic Traveling Salesman Problem, where GA coding is available. In Stochastic Scheduling Problem, processing times for operation can be treated as stochastic variables. In Stochastic Traveling Salesman Problem, times for traveling between two cities can be treated as stochastic variables. In these cases, GAUCE will be effective for getting good solution as that with highest frequency.

In this study, the Roulette strategy is adopted for selecting the good solution in terms of the expected value. However, there are other formulae as Stochastic Programming Problem. The solutions which present the highest probability for (A) the best solution or (B) the objective function value higher than a certain constant will be obtained, when GA operators are properly designed. Moreover, the distribution of the objective function value can be analyzed with this method.

However, in applying this method to Stochastic Programming Problem, the objective may be limited in some kinds, for example, expected value maximum and two cases associated with (A) and (B), described above.

The forthcoming paper will present other applications of this method.

4. Conclusions

A method for solving Stochastic Programming Problem has been developed with GA in uncertain environments. This method is successfully applied to Stochastic Optimal Assignment Problem, Stochastic Knapsack Problem and newly formulated Stochastic Image Compression Problem.

In this study, the Roulette strategy is adopted for selecting the optimum solution in terms of the expected value. As a result, the solution with highest frequency equals the optimum solution in almost all cases. When the expected value difference between the highest and the second highest is less than, for example, 2.5%, the solution with second highest expected value happens to be selected as the solution with highest frequency in some cases. Moreover, in rare cases, the locally optimum solution is selected as the solution with highest frequency.

Moreover, it is found that GAUCE has potential to search the second, the third best and the lower rank solution in terms of the expected value of objective function. This ranking may be useful in some cases where the optimum solution is not practical by some reason.

There are other formulas as Stochastic Programming Problem. The solutions which present the highest probability for (A) the best solution or (B) the objective function value higher than a certain constant will be obtained by GAUCE, when GA operators are properly designed. Moreover, the solution distributions can be analyzed with this method. GAUCE may be applied to many Stochastic Programming Problem, for example, Stochastic Scheduling Problem, Stochastic Traveling Salesman Problem, where GA coding is available.

Acknowledgment

The authors would like to thank Professor T.Ibaraki of Kyoto University for his valuable advice in our starting the study. The authors would like to thank Professor H.Ishii of Osaka University for his valuable suggestions on Stocahstic Programming Problem. The authors would like to thank Professor M.Tanaka of Konan University and T.Tagami of Assistant Professor of Kagawa Junior College for their valuable suggestions on GA.

References

[1] R. Aravind, G. L. Cash, D. L. Duttweiler, H. M. Hang, B. G. Haskell and A. Puri: Image and video coding standards. *ATT Technical Journal*, (1993) 67-88.

[2] E. M. L. Beale: On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, **B.17**(1955) 173-184.

[3] A. Charnes and W. W. Cooper: Chance constrained programming. *Management Science*, **6**(1959) 73-79.

[4] A. Charnes and W. W. Cooper: Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, **11**(1963) 18-39.

[5] G. B. Dantzig: Linear programming under uncertainty. *Management Science*, **1**(1955) 197-206.

[6] D. Dasgupta and D. R. Mcgregor: Nonstationary function optimization using the structured genetic algorithm. In: *Parallel Problem Solving from Nature, 2*, (1992) 145-154.

[7] C. Dinkelbach: On nonlinear fractional programming. *Management Science*, **13**(1967) 492-498.

[8] T. C. Fogarty: Varying the probability of mutation in the genetic algorithm. *Proceedings of 3rd International Conference on Genetic Algorithms*, (1989) 104-109.

[9] D. B. Fogel: An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Nwtworks*, **5**(1994) 3-14.

[10] C. M. Fonseca and P. J. Fleming: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of 5th International Conference on Genetic Algorithms*, (1993) 416-423.

[11] D. L. Gall: MPEG ;A video compression standard for multimedia applications. *Communications of the ACM*, **34**(1991) 46-58.

[12] D. Goldberg: *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, 1989).

[13] J. Grefenstette, R. Gopal, B. Rosmaita and D. Van Gucht: Genetic algorithms for the traveling salesman problem. *Proceedings of 1st International Conference on Genetic Algorithms*, (1985) 160-168.

[14] J. J. Grefenstette: Genetic algorithms for changing environments. In: *Parallel Problem Solving from Nature, 2* (Elsevier Science Publishers B.V., 1992), 137-144.

[15] M. I. Henig: Risk criteria in a stochastic knapsack problem. *Operations Reserch*, **38**(1990) 820-825.

[16] J. H. Holland: *Adaptation in Natural and Artificial Systems* (The Univ. Michigan Press, 1975 and MIT Press, 1992).

[17] J. Horn, N. Nafpliotis and D. E. Goldberg: A niched pareto genetic algorithm for multiobjective optimization. *Proceedings of 1st IEEE Conference on Evolutionary Computation*, (1994) 82-87.

[18] H. Ishibuchi and T. Murata: Multi-objective genetic local search algorithm. *Proceedings of IEEE International Conference on Evolutionary Computation*, (1996) 119-124.

[19] H. Ishii, T. Nishida and Y. Nanbu: Generalized chance constrained programming. *Journal of Operations Research Society of Japan*, **21**(1978) 124-145.

[20] ISO/IEC CD 13818-2: Generic coding of moving pictures and associated audio. ISO/IEC/JTC1/SC29/WG11, (1993).

[21] ISO/IEC IS 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, (1993).

[22] (JPEG)ISO CD 10918: Digital compression and coding of continuous-tone still images. ISO/IEC JTC1/SC2/WG10, (1991).

[23] S. Kataoka: A Stochastic programming model. *Econometrica*, **31**(1963) 181-196.

[24] C. Kosack, J. Marks and S. Shieber: A parallel genetic algorithm for network-diagram layout. *Proceedings of 4th International Conference on Genetic Algorithms*, (1991) 458-465.

[25] F. Kursawa: A variant of evolution strategies for vector optimization. In: *Parallel Problem Solving from Nature* (Springer-Verlag, Berlin, 1991), 193-197.

[26] A. Leger, T. Omachi and G. K. Wallace: JPEG still picture compression algorithm. *Optical Engineering*, **30**(1991) 947-954.

[27] A. M. Madansky: Inequalities for stochastic linear programming problem. *Management Science*, **6**(1960) 197-204.

[28] T. Murata and H. Ishibuchi: Multi-objective genetic algorithms. *Proceedings of 2nd IEEE-ICEC Conference*, (1995) 289-294.

[29] R. Nakano: Conventional genetic algorithm for job shop problems. *Proceedings of 4th International Conference on Genetic Algorithms*, (1991) 474-479.

[30] J. D. Schaffer: Multi-objective optimization with vector evaluated genetic algorithms. *Proceedings of 1st International Conference on Genetic Algorithms*, (1985) 93-100.

[31] M. Sniedovich: Preference order stochastic knapsack problems. *Journal of Operational Research Society*, **31**(1980) 1025-1032.

[32] E. Steinberg and M. S. Parks: A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of Operational Research Society*, **30**(1979) 141-147.

[33] A. Ueda: *Computer Simulation*(in Japanese) (Asakura, Tokyo, 1990), 77-85.

[34] F. Vavak and T. C. Fogarty: Comparison of steady state and generational genetic algorithms for use in nonstationary environments. *Proceedings of IEEE International Conference on Evolutionary Computation*, (1996) 192-195.

[35] D. W. Walkup and R. J. B. Wets: Stochastic programming with recourse. *SIAM Journal of Applied Mathematics*, **15**(1967) 1299-1314.

[36] G. K. Wallace: The JPEG still compression standard, *Communications of the ACM*, **34**(1991) 30-44.

[37] N. Yagi, S. Inoue, M. Hayashi, M. Okui and S. Goushi: *Practical Digital Picture Processing with C Language*(in Japanese) (Ohmusha, Tokyo, 1995), 240-256.

[38] H. Yasuda and H. Watanabe: *Basis of digital image compression*(in Japanese) (Nikkei BP Publication Center, Tokyo, 1996).

Yasunari Yoshitomi
Department of Computer Science and Systems Engineering,
Faculty of Engineering,
Miyazaki University
1-1 Gakuen Kibanadai Nishi, Miyazaki 889-2192, Japan
E-mail:yoshi@cs.miyazaki-u.ac.jp