Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

A MONTE CARLO STUDY OF GENETIC ALGORITHM INITIAL POPULATION GENERATION METHODS

Raymond R. Hill

Department of Operational Sciences
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765, U.S.A.

ABSTRACT

We briefly describe genetic algorithms (GAs) and focus attention on initial population generation methods for two-dimensional knapsack problems. Based on work describing the probability a random solution vector is feasible for 0-1 knapsack problems, we propose a simple heuristic for randomly generating good initial populations for genetic algorithm applications to two-dimensional knapsack problems. We report on an experiment comparing a current population generation technique with our proposed approach and find our proposed approach does a very good job of generating good initial populations.

1 INTRODUCTION

Genetic algorithms are search procedures inspired by biology and the workings of natural selection. Conceived by John Holland and his colleagues, GAs are now applied in many diverse applications, for instance, mathematical optimization, simulation parameterization, and real-time control. The broad focus of this paper is GA applied to optimization problems, and in particular initial population generation methods for a GA. Good initial populations facilitate a GA's convergence to good solutions while poor initial populations can hinder GA convergence. We propose an approach for obtaining good populations in the context of two-dimensional knapsack problems of the following form:

$$\begin{aligned} &\text{Maximize } \textstyle\sum_j C_j x_j \\ &\text{s.t. } \textstyle\sum_j A_{ij} x_j \le b_i \quad i=1,2 \\ &\qquad\qquad x_j \ge 0 \ \ \forall j \end{aligned} \tag{1}$$

This paper is organized as follows. Section 2 provides a brief overview of GAs and section 3 discusses initial population generation methods for a GA. Section 4 suggests a new, heuristic approach to generating initial GA populations based on knowledge of the optimization problem being solved. Section 5 describes how we study this approach while section 6 presents the results. Finally, section 7 summarizes this work and provides concluding comments.

2 OVERVIEW OF GENETIC ALGORITHMS

The GA name "originates from the analogy between the representation of a complex structure by means of a vector of components, and the idea, familiar to biologists, of the genetic structure of a chromosome" (Reeves, 1993). In biology, natural selection reinforces characteristics most amenable to a species survival. Genes within the chromosomes of the stronger members, corresponding to the more desirable characteristics, pass to subsequent generations through the reproduction process.

This paradigm fits optimization applications. Problem solutions (phenotypes) are encoded (genotypes), usually in binary format (genes). The set of solutions under consideration form a population with each solution considered a chromosome. The fitness of each member is generally the functional value of the phenotype, although specific applications may modify the fitness function, for example, to penalize problem constraint violations (penalty-based fitness function) in constrained optimization.

Fit chromosomes combine to produce chromosomes for subsequent populations. Member pairs of the population are selected for reproduction, usually based on some function of their individual fitness value. Genes from each parent are combined according to some predefined strategy to produce offspring (derivative chromosomes). The next generation is based on selecting parents and offspring for survival again according to some predefined strategy. Non-selected chromosomes "die" and are removed from consideration.

The fundamental concept in GA optimization applications is that better solutions share "good" gene combinations, or schema. Better schema produce fitter chromosomes in each generation and carry over during the reproduction process. Over many generations, these schema dominate and yield a population containing the best, possibly even the optimal, solutions.

There is no guarantee a GA will converge to an optimal solution, although experience suggests that a properly parameterized GA performs quite well. Parameters involved in a GA generally include: population size, number of generations to simulate, mating selection method, diversification or mutation rate, and the reproduction strategy. Our focus is on generating initial populations.

The initial population for a GA is a set of solutions to the optimization problem. Just as an initial starting point dictates the quality of a gradient-based non-linear optimization algorithm, the initial population can affect GA solution convergence. Some characteristics of any population are objective function value, feasibility of the solution, and level of infeasibility for any infeasible solutions.

3 POPULATION GENERATION AND REPAIR METHODS

There are a variety of approaches to generating initial populations. We consider a common approach and suggest a new approach

3.1 Random Generation

A common (often default) method of population generation is random generation. Each gene for a chromosome assumes a value of 1 with probability *p* and a value of 0 with probability *1-p.* Quite commonly Pr(X=1)=*0.5*. This approach is efficient and provides a population covering the feasible region, but the entire initial population may be infeasible. This means subsequent generations may remain infeasible with feasible solutions evolving slowly. A fitness function penalizing infeasibility is common. However, random solutions can be far from feasible and have large objective function values yielding poor performance for penalty-based fitness functions (Eravsar 1999).

3.2 Generation Based on Problem Structure

We suggest using information about the problem structure to arrive at better probability values for building initial populations randomly. This approach, and its motivation are discussed next.

4 A NEW HEURISTIC FOR RANDOM POPULATION GENERATION

We propose that initial populations for GA applications be randomly generated with Pr(X=1) based on problem knowledge. In particular, we suggest finding some feasible solution to the two-dimensional knapsack problem with a quick running heuristic and use the proportion of active decision variables (X_j =1) as Pr(X=1). This approach is motivated by Reilly (1998).

4.1 Proportion Of Feasible Solutions For Knapsack Problems

Reilly (1998) shows how to estimate the proportion of feasible solutions for the 0-1 knapsack problem and considers two-dimensional knapsack problems when the coefficients of the constraints are correlated. The key points of his effort are based on 0-1 knapsack problems (equation (1) with i=1).

Let

$$b = t\sum_j a_j, 0 < t < 1.$$

Then

$$F = \sum_j a_j x_j - t\sum_j a_j$$

is a random variable asymptotically normally distributed with mean $\mu_F = n(p-t)\mu_A$ and variance $\sigma_F^2 = n((p + t(t-1))\sigma_A^2 + \mu_A^2/4)$, where *p* is Pr(X=1) (normally Pr(X=1)=0.5 and is used by Reilly (1998)), *t* is the ratio used to establish the right-hand side values of a sample problem (the constraint slackness measure), and μ_A and σ_A are the mean and standard deviation of the distribution defined for the constraint coefficient vector, *A*. The probability of randomly generating a feasible solution is

$$\Pr(F \le 0) = \Phi(-\mu_F / \sigma_F),$$

where Φ is the cumulative distribution function for the standard normal random variable (Reilly 1998).

Tables 1 and 2 provide the probability that a randomly generated problem is feasible for a range of slackness ratios, *t*, and Pr(X=1) for two different distributions. (the distributions in our experiment). Three important points are apparent. First, *t* dictates the probability of a feasible solution more than the constraint coefficient distribution. Second, tighter constraints mean using a smaller Pr(X=1) to ensure a reasonable feasibility probability. And finally, feasible solutions are easy to generate with loose constraints.

If A^1~U(1,40) and A^2~U(1,15) for (1) then the probabilities in Tables 1 and 2 bound above the proportion of random solutions feasible with respect to **both** constraints. However, correlation between these constraints affects these probabilities, and in fact only under perfect positive correlation are these bounds attained (the constraints are identical). Reilly (1998) shows that as correlation decreases so does the probability a random solution is feasible for the problem.

Table 1: Probability of Feasible Random Solutions for A~U(1,40)

	Slackness Ratio - t					
Pr(X=1)	0.30	0.40	0.50	0.60	0.70	0.80
0.10	1.00	1.00	1.00	1.00	1.00	1.00
0.15	1.00	1.00	1.00	1.00	1.00	1.00
0.20	0.98	1.00	1.00	1.00	1.00	1.00
0.25	0.84	1.00	1.00	1.00	1.00	1.00
0.30	0.50	0.97	1.00	1.00	1.00	1.00
0.35	0.18	0.83	1.00	1.00	1.00	1.00
0.40	0.04	0.50	0.97	1.00	1.00	1.00
0.45	0.00	0.19	0.81	1.00	1.00	1.00
0.50	0.00	0.04	0.50	0.96	1.00	1.00
0.55	0.00	0.01	0.20	0.80	0.99	1.00
0.60	0.00	0.00	0.05	0.50	0.95	1.00
0.65	0.00	0.00	0.01	0.21	0.79	0.99
0.70	0.00	0.00	0.00	0.06	0.50	0.94
0.75	0.00	0.00	0.00	0.01	0.22	0.78
0.80	0.00	0.00	0.00	0.00	0.07	0.50
0.85	0.00	0.00	0.00	0.00	0.01	0.23
0.90	0.00	0.00	0.00	0.00	0.00	0.08
0.95	0.00	0.00	0.00	0.00	0.00	0.02

Table 2: Probability of Feasible Random Solutions for A~U(1,15)

	Slackness Ratio - t					
Pr(X=1)	0.30	0.40	0.50	0.60	0.70	0.80
0.10	1.00	1.00	1.00	1.00	1.00	1.00
0.15	1.00	1.00	1.00	1.00	1.00	1.00
0.20	0.98	1.00	1.00	1.00	1.00	1.00
0.25	0.84	1.00	1.00	1.00	1.00	1.00
0.30	0.50	0.97	1.00	1.00	1.00	1.00
0.35	0.18	0.83	1.00	1.00	1.00	1.00
0.40	0.04	0.50	0.97	1.00	1.00	1.00
0.45	0.00	0.19	0.82	1.00	1.00	1.00
0.50	0.00	0.04	0.50	0.96	1.00	1.00
0.55	0.00	0.01	0.19	0.80	0.99	1.00
0.60	0.00	0.00	0.05	0.50	0.95	1.00
0.65	0.00	0.00	0.01	0.21	0.79	0.99
0.70	0.00	0.00	0.00	0.05	0.50	0.94
0.75	0.00	0.00	0.00	0.01	0.22	0.78
0.80	0.00	0.00	0.00	0.00	0.06	0.50
0.85	0.00	0.00	0.00	0.00	0.01	0.23
0.90	0.00	0.00	0.00	0.00	0.00	0.07
0.95	0.00	0.00	0.00	0.00	0.00	0.02

4.2 Using Problem Information To Infer Reasonable Probability Values

Idealistically, one might pre-process a problem, determine the slackness ratio values, determine the interconstraint correlation, and compute a reasonable value for Pr(X=1). This value provides an expected proportion of the decision variables to set to a value of 1. We suggest an easier approach. Solve the problem with a greedy heuristic, use the ratio of active (X_j=1) to total decision variables as Pr(X=1), and then randomly generate the initial population. Our conjecture is that this approach will yield a good portion of feasible solutions, and moreover these solutions should be “good” both in the sense of objective function value and in terms of near-feasibility.

5 THE EXPERIMENT

Discrete distributions were used to generate two-dimensional knapsack sample problems, specifically C ~ U(1,100), A^1 ~ U(1,40), and A^2 ~ U(1,15). Problem correlation structure was controlled across the complete range of feasible correlation structures (45 feasible correlation structures). Additionally, four settings for the right-hand side coefficients were considered: t_1={0.3, 0.7} and t_2={0.3, 0.7}. A total of 180 problems were generated. This generation scheme was used in Hill (1996) so optimal solutions were available for the problems generated.

For each problem, 100 solutions were generated randomly and according to the proposed heuristic. Random numbers were synchronized between the approaches. Each solution was evaluated and a level of infeasibility determined, if the solution was in fact infeasible. Of interest is the frequency with which each method yields infeasible solutions, the overall quality of the solution generated, and how close to feasibility were infeasible solutions. The heuristic of Toyoda (1975) was used to solve the problems to set Pr(X=1).

6 RESULTS

The proposed heuristic approach faired extremely well and represents a reasonable approach for GA population generation. As Tables 1 and 2 suggest, the challenge is to produce good populations for the more difficult, tightly constrained problems. Table 3 summarizes how often each approach, the random and our proposed heuristic, produced feasible solutions.

Table 3: Percentage Feasible Solutions Produced by Each Approach

	Approach	
Slackness	Random	Heuristic
$t_1 = t_2 = 0.3$	0.0 %	20.2 %
$t_1 \neq t_2$	0.1 %	6.7%
$t_1 = t_2 = 0.7$	100 %	41.5 %

The random approach, with $Pr(X_j = 1) = 0.5$, performs as predicted by Tables 1 and 2. The heuristic uses a dynamically set $Pr(X_j = 1)$ improving over the random approach when constraints are tight although the approach does yield more infeasible solutions when both constraints are loose.

Table 3 results can be misleading since infeasible solutions vary by degree of infeasibility. A GA employing a penalty-based fitness measure may handle near-feasible solutions quite well. Table 4 summarizes the infeasibility levels using the following function for each constraint:

$$f_i = \max\left\{\left(\sum_j x_j a_{ij} - t_i \sum_j a_{ij}\right) / t_i \sum_j a_{ij}, 0\right\}, i = 1,2$$

Whenever t=0.3, random solutions are very infeasible. Infeasible heuristic solutions are close to feasible. These better heuristic solutions facilitate a penalty-based fitness function.

Table 4: Average Infeasibility Ratios, f_i, for Infeasible Solutions

	Approach			
	Random		Heuristic	
Slackness	f_1	f_2	f_1	f_2
$t_1 = t_2 = 0.30$	0.67	0.68	0.29	0.29
$t_1 \neq t_2$	0.34	0.33	0.27	0.25
$t_1 = t_2 = 0.70$	0.00	0.00	0.06	0.07

Solution quality is also measured by objective function value. Table 5 summarizes the average objective function value by constraint slackness settings.

Table 5: Average Objective Function Values by Constraint Slackness Settings

		t_2 Values	
t_1 Values	Approach	0.30	0.70
	Random	2507.40	2491.96
0.30	Heuristic	1837.40	2255.64
	Random	2557.11	2490.48
0.70	Heuristic	2270.49	3464.43

Since most (if not all) randomly generated solutions when either $t_1 = 0.3$ or $t_2 = 0.3$ are infeasible, the corresponding objective function values are inflated (too many $X_j = 1$). In some cases, these values are not much larger than the heuristic solution values, whose solutions are feasible or close to feasibility. When all constraints are loose, the heuristic yields stronger solutions than the random approach (higher objective function values, very close to feasibility), despite the 100% feasibility of the random approach. This is due to the setting of $\Pr(X_j = 1)$ values by the heuristic

Table 6 summarizes the heuristic's $\Pr(X_j = 1)$ values by constraint slackness settings. When any constraint is tight, $\Pr(X_j = 1)$ is reduced. Reilly's (1998) formula predicts a higher probability of attaining a feasible solution when $\Pr(X_j = 1)$ is reduced in this fashion. The proportion of feasible solutions attained agreed with this predictive formula (correlation over 0.98 between achieved and predicted proportion of feasible solutions). Notice when constraints are loose, the heuristic produces a very high $\Pr(X_j = 1)$ value. This causes some of those problems to be infeasible, but as demonstrated those solutions are still very close to feasibility and yield very good objective function values.

Table 6: Average $\Pr(X_j = 1)$ Values by Constraint Slackness Settings

	t_2 Values	
t_1 Values	0.30	0.70
0.30	0.366	0.454
0.70	0.442	0.693

Reilly shows that interconstraint correlation, $\rho(A^1 A^2)$, effects solution feasibility probabilities. We examine the effect of $\rho(A^1 A^2)$ in Table 7.

Table 7: Infeasibility Ratios, f_i, and Average $\Pr(X_j = 1)$ Values by $\rho(A^1 A^2)$ Setting

	Target $\rho(A^1 A^2)$ Values				
	-0.99	-0.49	0.0	0.49	0.99
f_1	0.16	0.17	0.22	0.27	0.30
f_2	0.12	0.18	0.20	0.27	0.30
$\Pr(X_j = 1)$	0.43	0.47	0.49	0.52	0.53

As $\rho(A^1 A^2)$ drops, problems get more difficult to solve both in terms of proportion of feasibility (based on Reilly (1998)) and in solution procedure performance (Hill, 1996). As $\rho(A^1 A^2)$ drops, the heuristic reduces $\Pr(X_j = 1)$ values and actually reduces infeasibility ratios. A corresponding table for the random generation approach would show all f_i values around 0.33 and $\Pr(X_j=1)=0.5$ throughout the table.

7 SUMMARY AND CONCLUSIONS

GAs are an increasingly popular heuristic method for optimization applications and meta-heuristic applications. Reilly's (1998) discussion of how problem structure effects solution space density prompted the GA initial population heuristic approach we propose. Compared to default random generation methods, this heuristic performs especially well. The near-feasible solutions produced by the heuristic, especially under the tougher conditions of tight constraints and decreased correlation between constraints, should be especially attractive to penalty-based fitness function applications of GA.

A next logical step is to compare GA performance using the initial population produced by our heuristic against other initial population methods. Our conjecture is that this proposed heuristic will provide an initial population of sufficient quality and diversity to produce favorable convergence to good solutions.

REFERENCES

Eravsar, Mehmet. 1999. A Comparison of Genetic Algorithm Parameterization on Synthetic Optimization Problems. Masters Thesis, AFIT/ENS/99M-05, Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Hill, Raymond R. 1996. *Multivariate Sampling with Explicit Correlation Induction for Simulation and Optimization Studies*. Ph.D. Dissertation, Department of Industrial and Systems Engineering, The Ohio State University, Columbus, OH.

Reeves, C. 1993. Genetic Algorithms. In *Modern Heuristic Techniques for Combinatorial Problems,* ed. Colin R. Reeves, 151-196. New York: John Wiley & Sons.

Reilly, Charles H. 1998. Properties of Synthetic Optimization Problems. *Proceedings of the 1998 Winter Simulation Conference*, eds. D.J. Dedeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan. 617-621. Institute of Electrical and Electronics Engineers, Washington, DC.

Senju and Toyoda. 1968. A Approach to Linear Programming with 0-1 Variables. *Management Science*, **15**(4), B196-B207.

Toyoda, Y. 1975. A Simplified Approach for Obtaining Approximate Solutions to Zero-One Programming Problems. *Management Science*, **21**(12), 1417-1427.

AUTHOR BIOGRAPHY

RAYMOND R. HILL is an Assistant Professor of Operations Research in the Department of Operational Sciences at the Air Force Institute of Technology. His research interests include modeling and simulation, applications of optimization, and military applications of modeling.