

A Genetic Algorithm for Register Allocation

K. M. Elleithy
Computer Engineering Department
King Fahd University of Petroleum and
Minerals
Dhahran 31261, Saudi Arabia
Email: elleithy@kfupm.edu.sa

E. G. Abd-El-Fattah
Computer Department
Moshamit Behera Company
Alexandria Egypt

Abstract

In this paper we introduce a new genetic algorithm for register allocation. A merge operator is used to generate new individual solutions. The number of steps required to examine all pairs in the population matrix to generate n^2 (n is the population matrix size). Generating an offspring from the parents needs m steps (m number of nodes). The total number of steps required by the algorithm is n^2m , that is, the genetic algorithm has a linear time complexity in terms of number of nodes. The experimental results show optimal solutions in many of the graphs used for testing.

1. Introduction

A recently proposed paradigm for solving hard optimization problems is Genetic Algorithms (GA). GA has been successfully employed for solving many combinatorial problems [1-4] in areas such as pattern classification, machine learning, scheduling, and VLSI placement and floor planning. Genetic algorithms are search algorithms that emulate the natural process of evolution as a means of processing toward the optimum. A genetic algorithm starts with an initial set of random configurations called a population. Each individual in the population, termed chromosome, is a string of symbols called genes. A chromosome represents a solution for the optimization problem. During each iteration, two individuals at a time, called parents, are selected from the population based on a fitness value. A number of genetic operators, as crossover or inversion, are applied on the selected parents to generate new individual solutions called offsprings. These genetic operators combine the features of both parents. An outline of a genetic algorithm is shown in Figure 1.

2. A new algorithm for register allocation

In this paper we introduce a new algorithm for register allocation. The algorithm is shown in Figure 2. The algorithm uses the following:

Procedure Genetic

```
{  $N_p$  : population size}
{  $N_g$  : number of generations}
{  $N_o$  : number of offsprings}
{  $P_i$  : Inversion probability}
{Population : population matrix of size  $N_p$ }

begin
Generate an initial valid population;
for j=1 to  $N_p$ 
do evaluate fitness(population[j]);
for i=1 to  $N_g$  do
begin
for j=1 to  $N_o$  do
begin
choose parents with probability proportional to
fitness value;
perform crossover to generate offsprings;
for k = 1 to  $N_p$  do
apply inversion(population[k]) with probability
 $P_i$ ;
Evaluate fitness(offspring[j]);
end;
population ← select(population, offspring,  $N_p$ )
end;
end;
Return highest scoring configuration in population;
end.
```

Figure 1 Procedure for Genetic Algorithm.

2.1 Initial population

An initial population consists of any random valid solutions or it can be generated using a starting procedure. The advantage of using a starting procedure is to start with a good solution that can be improved.

2.2 Hamming distance

Let A and B be any two individual strings of length N. The hamming distance is defined as the total number of positions where $A_i \neq B_i$.

Example

Let A = [0101100], B = [1011100]. A and B are different in positions 1, 2, and 3, i.e., the hamming distance is 3.

2.3 Merge procedure

Let A and B be any two individual strings of length N. An offspring C is generated by merging as follows:

$$c_i = a_i \quad \text{if} \quad a_i = b_i$$

$$c_i = a_i \quad \text{if} \quad a_i \neq b_i, a_i \geq 0 \text{ and}$$

$$\text{netsave}(a_i) \geq 0$$

$$\text{if} \quad \dots \text{ and}$$

otherwise

2.4 Fitness value

It is the profit function that is defined as follows :

The genetic procedure for register allocation is shown in Figure 2.

3. Conclusions

The genetic algorithm presented in this paper may be used to enhance previous results obtained using a starting approach, e.g., simulated annealing. Let n be the population matrix size and m be the vector length (no of nodes). Steps are required to examine all pairs in the population matrix to generate candidate offspring. Generating an offspring from the parents needs m steps. The total number of steps required by the algorithm is $n^2 \cdot m$, i.e., the genetic algorithm has a linear time complexity in terms of number of nodes.

4. References

- [1] J. P. Cohoon, S. U. Hegde, W. N. Martin and D. S. ed Genetic Algorithms for Floorplan
4, April 1991, pp. 483-492.
- [2] Proceedings of International Conference on Genetic Algorithms and their Applications, 1987, pp. 136-140.
- [3] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Intelligence*, Addison-Wesley, 1989.
- [4] Standard Cell Placement using Meta-Genetic Parameter Optimiza
.., vol. 5,
May 1990, pp. 500-512.

Acknowledgements

The first author would like to acknowledge the support received from KFUPM in preparation and presentation of this paper.

Procedure Register-Allocation-by-Genetic Algorithm

```
{maxtime : is the total time allowed for the genetic
process}

begin

get the population-matrix and the corresponding profit
vector ;

for x = 1 to maxtime do
begin

Evaluate the hamming-distance between two
individuals;

If hamming-distance > 1 then
Merge the two individuals to generate the new
offspring;

If offspring-profit > max(parents' profit) then

Accept the offspring as a new individual in the
population-matrix by deleting any of the
parents and replacing the generated
offspring;

end;
end.
```

Figure 2 Procedure for Register Allocation by Genetic Algorithm.