Research on a Heuristic GA-Based Decision Support System for Rice in Heilongjiang Province

Ran Cao*, Yushu Yang, and Wei Guo

Engineering college of Northeast Agricultural University, Haerbin, China
Tel.: 13796637319
caohefan@sina.com

Abstract. As we know rice is the main food in China. In recent years, the rapid development of agricultural decision support system provides new management methods for rice production. GA has been regarded as an effective analytic tool and stochastic search technique to solve large and complicated problems. However, traditional GA-based decision support system does not always result in good solutions and search efficiency due to random method. This study presents a heuristic genetic algorithm-based decision support system to search the optimal solution. Heuristic method provides a robust and efficient approach for solving complex real-world problems. This hybrid algorithm incorporates concepts from GA and heuristic method and creates individuals in a new generation not only by crossover and mutation operations as found in GA but also by heuristic mechanism. The result of experiments reveals the proposed algorithm is more computationally effective and efficient than GA alone in terms of solution quality.

Keywords: Decision support system; Genetic algorithms; Heuristic method.

1 Introduction

Heilongjiang Province is a major production base of rice in China. Cultivation area of rice is nearly 166,700 hectare. It makes up19% percent of the arable land area in our province. However, Heilongjiang Province is located in the cold and remote area. Economic advantages of rice are not play at all. In order to improve the output of rice, it is inevitable to make use of modern information technology to reform traditional crop cultivation.

With the fast development of information technology, there are more and more tools available to decision makers. The most widely deployed systems are decision support systems. Decision support systems (DSS) are computer technology tools that provide solutions that can be used to support complex decision making and problem solving. In a word, it is designed to assist in identifying patterns, problems, opportunities, and eventually in making decisions [1].

In order to more accurately find an optimal solution in DSS, this paper significantly presented a GA scheduling algorithm which was enhanced by heuristic

* Supported by Heilongjiang Youth Science Funds project (QC2009C87).

D. Li, Y. Liu, and Y. Chen (Eds.): CCTA 2010, Part II, IFIP AICT 345, pp. 322–328, 2011.

method. Taking advantage of the compensatory property of GA and heuristic method, the proposed algorithm combines the evolutionary natures of both (denoted as HGA). The robustness of HGA will be tested by a set of experiments and the results are compared extensively with GA and this hybrid algorithm.

2 Structure of the System

The framework of this DSS includes three parts:

First: decision-making. The system provides a variety of possible strategies for users. It is based on the data of species, weather, soil and measures of cultivation management which is provided by database. Second: optimization. HGA algorithm analyzes a variety of strategies and then acquires optimal solution. Third: comparison and feedback. The system compares and analyzes suggestions from domain experts and users with the optimal solution to derive the best practice approach, and then feeds it back into the database in order to provide a favorable basis for the future decision. The system structure diagram is shown in Fig.1.

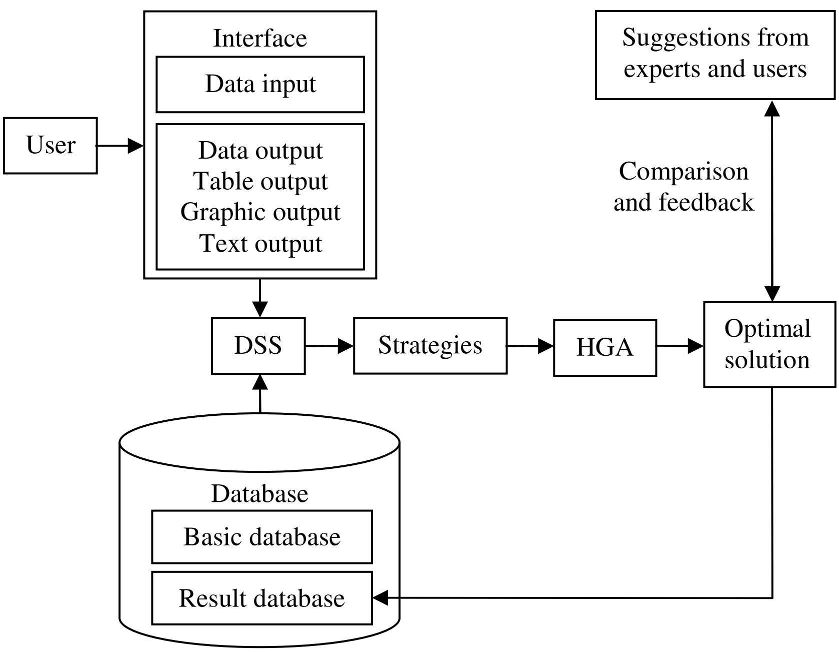

Fig. 1. Architecture of decision support system

3 Genetic Algorithm (GA)

There are three well-known randomization optimization techniques, SA, TS, and evolution algorithms (EA). The results of past research indicate that SA can converge to an optimal solution in all probability. However, it is practically impossible to satisfy certain conditions that ensure the global convergence of SA. In fact, the computation burdens of SA are extremely heavy. TS depends heavily on heuristic techniques. Hence, although they can converge faster than other randomization optimization techniques, they frequently fall into local optima. EA include all algorithms simulating the phenomena of living systems. GA is the most popular among all algorithms of EA [2]. GA is stochastic search techniques based on analogy to Darwinian natural selection. Individuals who fit the environment best should have a better chance to propagate their offspring. By the same reason, solutions that have the best "fitness" should receive higher probability to search their "neighbors". The main advantage of GA lies in its powerful implicit parallelism. Moreover, a GA starts its global search simultaneously from many initial points, while both SA and TS start from only one initial point and search sequentially. Therefore, compared with the other randomization techniques mentioned above, a GA probably has the highest possibility of reaching global optima in a defined time interval and makes the best compromise between solution effectiveness and efficiency. Because all advantages of GA we have said above, GA is adopted in this paper.

However, GA have three shortcomings. If the fitness value is set improperly, convergence can occur prematurely. The second is that the ability of local search is poor and the speed of convergence is slow. The third is that a traditional GA is a random process to generate initial population for each chromosome [3]. Most of the randomly generated chromosomes have poor total performance and thus pass unfavorable traits to their offspring. As we known, the heuristic method can not guarantee the optimal solution but only get the approximate optimal solution. We combine genetic algorithm with heuristic method because complementary characteristics of genetic algorithm and heuristic method. This hybrid algorithm can improve the effect of genetic algorithm.

4 Proposed Heuristic Genetic Algorithm (HGA)

The hybrid method between genetic algorithm and heuristic method is various. For example, we can integrate heuristic method with initialization in order to generate initial population with good performance. We also can use local search heuristics as basic cycle plug-ins of genetic algorithm in order to execute quick local search [4]. This paper adopted these two methods.

4.1 Heuristic Search

We introduced heuristic search in this algorithm. Heuristic search is that evaluating each search location in the state space to get the best location and then searching from this location until it reaches the target. It can omit a large number of insignificant search path, thereby improving efficiency. It is very important to evaluate the location in heuristic search. Different evaluation may produce different results.

Evaluation is denoted as evaluation function in heuristic search. Such as: $f(X)=g(X)+h(X)$, where $f(X)$ is the evaluation function of node X. Assumed that setting out from starting node (denoted as S). We will calculate the minimum evaluation of reaching the target node (denoted as T) which is pass through node X. Evaluation function f(X) consists of two parts: one part is the minimum cost which is from S to X, denoted as g(X). The other is the minimum cost which is from X to T, denoted as h(X). In the actual process we use $f(X)=C_{SX}+l(X,x)+h(x)$, where C_{SX} is denoted as the minimum cost which is from S to current node X. l(X,x) is the straight length between the current node X to expanding node (denoted as x). h(x) is the path length between x and T. l(X,x)+h(x) is heuristic function.

We introduced heuristic search to improve the overall performance of population. It can make local search which is based on the corresponding phenotype of each individual in population because of the introduction of heuristic search. Local search can identify local optimal solution in the current environment.

4.2 Encoding and Initial Population

In solving problems by GA, first task is to represent a strategy of the DSS as a chromosome. Each solution of the DSS has a value. This value corresponds to the actual performance of the solution. In this paper, strategies will be sorted from big to small according to their value and then make a queue. Fig.2 shows the encoding used in the GA. Each number represents the value of a unique strategy, with 0 being used to delimit different queues.

8 6 5	0	9 8	0	7 3 1	0	6 2

Fig. 2. Encoding of HGA

The distribution of the initial population has an important impact on convergence rate of genetic algorithm and the phenomena of avoiding premature. Too fragmented initial population will affect convergence rate. However, the oversimplified distribution of population will result in the lack of plurality and the genetic algorithm is prone to premature. Premature is the main reason of trapping in local optimization. It is very important to think about how to utilize some prior knowledge to restrict random distribution by combining heuristic rules and how to improve convergence rate of system.

We propose heuristics to obtain good solutions for the initial population. While the first five chromosomes of the initial population are created by heuristics, others are randomly generated.

We referred to priority rules as the heuristic information in this research. Priority rule is a basic heuristic rule. The individual which is based on priority rules usually has a high fitness. And the individual which is generated randomly ensures the diversity of initial individual. The combination of the two methods could unite diversity and high adaptability of initial population. It also could accelerate the evolutionary process.

4.3 Fitness Function

It doesn't utilize external information when genetic algorithm is in evolutionary search. It is only based on fitness function and makes use of fitness value of each individual in population to search. Therefore, fitness function directly affects convergence rate of genetic algorithm and whether the global optimal solution can be found.

In order to not producing invalid chromosomes, it doesn't need to introduce penalty function when we evaluate individual fitness. Therefore, the objective function itself is considered as the fitness function. Accordingly, feasible and infeasible solutions can be obtained for each population group. In this case if solution is feasible, value of fitness function is value of the objective function, otherwise, value of fitness function is equal to zero.

4.4 Genetic Operators

An initially chosen chromosome leads to several other chromosomes through operators such as crossover and mutation. These operators are properly used to obtain improved results in each generation and finally the process is stopped when the improvement process stops.

4.4.1 Crossover

Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining both chromosomes' features [5]. There are several methods to crossover, partial-mapped crossover (PMX) is adopted in the proposed GA.

4.4.2 Mutation

Similar to crossover, mutation is done to prevent the premature convergence and explore new solution space. However, unlike the crossover, mutation is usually done by modifying gene within a chromosome [6]. There are also a number of methods to mutate, such as insertion, swap and reversion. Reversion is employed in this paper.

4.5 Termination Condition

When the stopping conditions are met, the evolution of the population will stop. This is to prevent the GA from running forever. We use two stopping conditions: First, there is an upper bound on the maximum number of generations, to guarantee evolution will halt. Secondly, if the fitness value of best individual in continuous t generations is no longer raise, the evolution will stop.

5 Computational Results

The parameters of HGA are set as follows: population size is 100, crossover rate is 1.1, and mutation rate is 0.02. The termination condition when fitness values of chromosomes in a population are the same. And when the number of generations exceeds 100,000, searching will stop automatically. Because genetic algorithm is a stochastic searching heuristic, the result of every test is unlikely to be the same. In order to compare the average performance, 9 tests were conducted.

To evaluate the algorithm's efficiency and effectiveness, comparing the performance of HGA and traditional GA by exhaustive simulation running. The simulation results are depicted in Table 1 and Fig.3.

Table 1. Comparison between the performance of pure GA and HGA

Number of solutions from DSS	Mean computation time(s)		Average improvement of HGA over GA(%)
	HGA	GA	
3	3.58	26.75	0.11
5	12.24	29.31	0.32
6	23.16	34.72	0.84
8	25.05	37.82	1.41
12	48.00	63.25	1.62
15	75.16	97.10	1.94
18	100.06	216.42	1.98
19	142.43	292.45	2.15
22	153.48	346.12	2.23

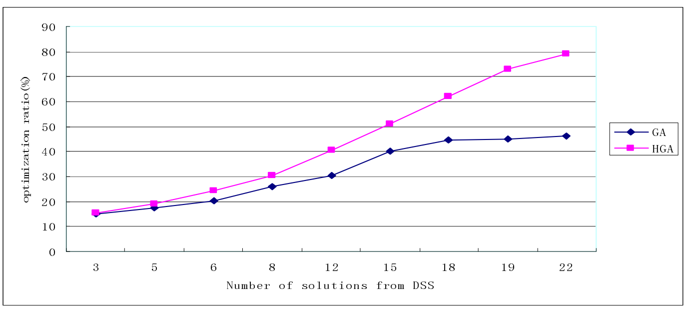

Fig. 3. Comparison between pure GA and HGA on optimization

Table 1 reports the CPU time, and the improvement rate of HGA over GA in our experiment according to the number of strategies. From Table 1, it is clear that GA and HGA have similar performances on small number. However, when the number of strategies became large HGA is about 1.5–2.0% better than GA at the cost of additional computational time.

And then examine the optimization behaviors. As shown in Fig.3, when generation number is small the optimization ratio of HGA is a little higher than GA. However, with the generation number increases the gap of them became bigger. And GA reached a saturation point soon.

Through the simulation experiments, we can see that HGA had better efficiency and effectiveness than GA.

6 Conclusion

This paper proposes a heuristic genetic algorithm to optimize solutions from DSS. Although the proposed HGA adopted classical genetic operators, its evolutionary mechanism is completely different and much more effective than classical GA. First of all, in this research, the algorithm has a more powerful search-oriented ability by using heuristic search method. Additionally, performance of the algorithm has been further improved by using sorted encoding and heuristic method of priority rule. This paper also introduced the design of fitness function and the choice of various genetic operators in details. In a word, a major gain is achieved in the addition of optimization ratio by the heuristic GA when compared with the traditional GA.

References

1. Zahedi, F., Song, J., Jarupathirun, S.: Web-based decision support. Handbook on Decision Support Systems 1, 315–338 (2008)
2. Low, C., Yeh, Y.: Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated. Robotics and Computer-Integrated Manufacturing 25, 314–322 (2009)
3. Preechakul, C., Kheawhom, S.: Modified genetic algorithm with sampling techniques for chemical engineering optimization. Journal of Industrial and Engineering Chemistry 15, 110–118 (2009)
4. Lai, Y.C., Ding, M., Grebogi, C.: Controlling Hamiltonian chaos. Phys. Rev. E, 86–92 (1993)
5. Tariq, A., Hussain, I., Ghafoor, A.: A hybrid genetic algorithm for machine-part grouping. Computers & Industrial Engineering 56, 347–356 (2009)
6. Fulya, A., Mitsuo, G., Lin, L., Ismail, K.: A steady-state genetic algorithm for multi-product supply chain network design. Computers & Industrial Engineering 56, 521–537 (2009)