2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Optimization of Dispensing Parameter Fine Tuning Program Using Genetic Algorithm

Edwin Y.S. Sim[1], S.P. Koh[2], C.P. Chen[3], S.K.Tiong[4], Albert Y.C. Fong[5+]

Department of Electronic & Communication Engineering
Universiti Tenaga Nasional, Km 7, Jalan Kajang-Puchong
43009 Kajang, Selangor

Abstract. Development in automated controlled system for material processing in materials-mixing plants had examined various benefits of its applications in the industry. One of the most prominent methods used is by having a dispensing system in handling the end process for materials-mixing. This paper involves in developing an automatic dispensing parameter fine tuning program via Genetic Algorithm (GA) that is capable of performing independent learning capability and optimization for the dispensing parameters using GA operators**.** The practical usage of GA techniques could overcome limitations of the manual parameter tuning presently applied in the coatings industry. The fast and accurate tuning system which is modularly built out of individual dispense valve is able to handle different fluids with varying viscosity.

Keywords: Dispensing System, GA Dispensing Parameter Fine Tuning, Genetic Algorithm, Coatings

1. Introduction

Coatings manufacturers have been looking at ways to help printers dispense coating more accurately and efficiently. Accuracy is a key word here. If the system is not accurate and relatively simple to use, the printer is going to be frustrated with the process and the results, which will be more bad quality coating. Size also matters: no one really wants a bulky system taking up an inordinate amount of floor space. A computerized system using state-of-the-art software is critical. If an operator can call up a pre-programmed formula for a specified coating, press a button and have the blend within minutes, the job will run more smoothly.

To meet these needs by printers, some coating companies, in conjunction with dispenser machine manufacturers such as Vale-Tech [1], Stork Prints [2], Novaflow [3], Rexson [4] and GFI [5] have developed automatic Gravimetric Dispensing System that help printers accurately and efficiently dispense their spot coatings. There is considerable demand, especially among printers who use large volumes of coating and have needs for blending of special coatings. Printers want to be able to have a wide variety of specialized spot coating available on short notice. These systems simplify the printer's work and offer several advantages to the printer. It eliminates the need for the pressperson to climb up on the press and physically scoop the coating out of a can to fill the coating fountain and to have to monitor the coating consumption. There is no coating skinning in the tube, kit or drum, so it eliminates the inconvenience of scraping the coating skin from the coating surface. Waste is also reduced since these systems can operate with the minimum amount of coating in the fountain which subsequently reduces the amount of discarded coating.

However, most of the coatings manufacturers are facing the perennial problem of obtaining optimum dispensing parameters from their existing systems. The problem is especially acute when involving manual tuning by the operators. In order to facilitate inexperienced and experienced operators, an automatic dispensing parameter fine tuning program via GA [6-10] would simplifies the complex solution currently faced in the coatings industry as well as increases the process reliability.

+ *E-mail address*: (edwin@uniten.edu.my, johnnykoh@uniten.edu.my, chaiphing@uniten.edu.my, siehkiong@uniten.edu.my, albert@uniten.edu.my).

2. Methodology

A number of tests were conducted using the GA parameters as depicted in Table 1. Note that the required size of population is relatively small as each given gene in the chromosome is set within a fixed range that would reduce the searching space, hence reducing the number of chromosomes and generations required for convergence.

Table 1: GA Parameters

Parameters	Specification
Population Size	10
Type of Selection	Roulette Wheel
Type of Crossover	2 point Dynamic Crossover
Crossover Rate	80% of the Population
Type of Mutation	1 bit Mutation
Mutation Size	1 Chromosome
Criteria Termination	20 Generations

The flow chart of the GA dispensing parameter fine tuning program is illustrated in Figure 1. The program starts with the initialization of dispensing parameters within the range and encoding of the GA parameters (Table 1). This follows by generation of the initial population. The dispensing software would then use the parameters from the initial population to obtain the chromosomes for accuracy and speed. The chromosomes are then evaluated and confirmed whether the solution has been obtained. In the event of failure to obtain the solution, a selection by Roulette Wheel subroutine and 2 point dynamic crossover subroutine ensues. Once the crossover operation is completed, the chromosomes are then again evaluated to determine whether the solution has been found. The program would stop if the solution is obtained. Else, the program would then go through the 1 bit mutation subroutine, where the selected chromosome would be mutated and re-evaluated. The program would stop if the solution is obtained. Otherwise, it would generate the next population with good offspring and repeat the process. The best solution obtained in the current generation would then be brought forward as one of the chromosome in the new generation, while other chromosomes would be generated randomly. This process would be repeated until either the solution is found or upon reaching the maximum of 20 generations.

The chromosome is an element that contains the dispensing parameters (genes). The genes are set according to the real dispensing parameters where the values are within the valid range that is suitable for dispensing application. The genes form the chromosome and define a complete solution. Several chromosomes form a population which then evolve and create new generations of chromosomes.

The chromosome could experience crossovers and mutations. In crossovers, a large portion of its genes are interchanged with another chromosome where each of the chromosome is characterized by a fitness level. In mutations, a mutation point is generated randomly using Equation 1 and the corresponding bit would be mutated.

$$\text{Bounded Number} = [(\text{Max} - \text{Min}) * \text{Random Number}] + \text{Min} \quad (1)$$

where Max = Max value in the fixed range
Min = Min value in the fixed range

The overall fitness function (F) for these applications contains terms that quantify the fitness of accuracy (F_e) and speed (F_s). Both F_e and F_s consists of actual elements of accuracy and speed that are obtained from online measurement from the precision balancer and dispensing time. The fitness function is defined by Equation 2 to 5.

$$F = 1 / \{\exp[\, u\, F_{es}\,]^{v}\,\} \quad (2)$$

Where

$$F_{es} = F_e * F_s \quad (3)$$

$$F_e = 1 / \{1 - \exp[\, w\, (\text{Accuracy})^{x}\,]\} \quad (4)$$

$$F_s = 1 / \{1 - \exp[\, y\, (\text{Speed})^{z}\,]\} \quad (5)$$

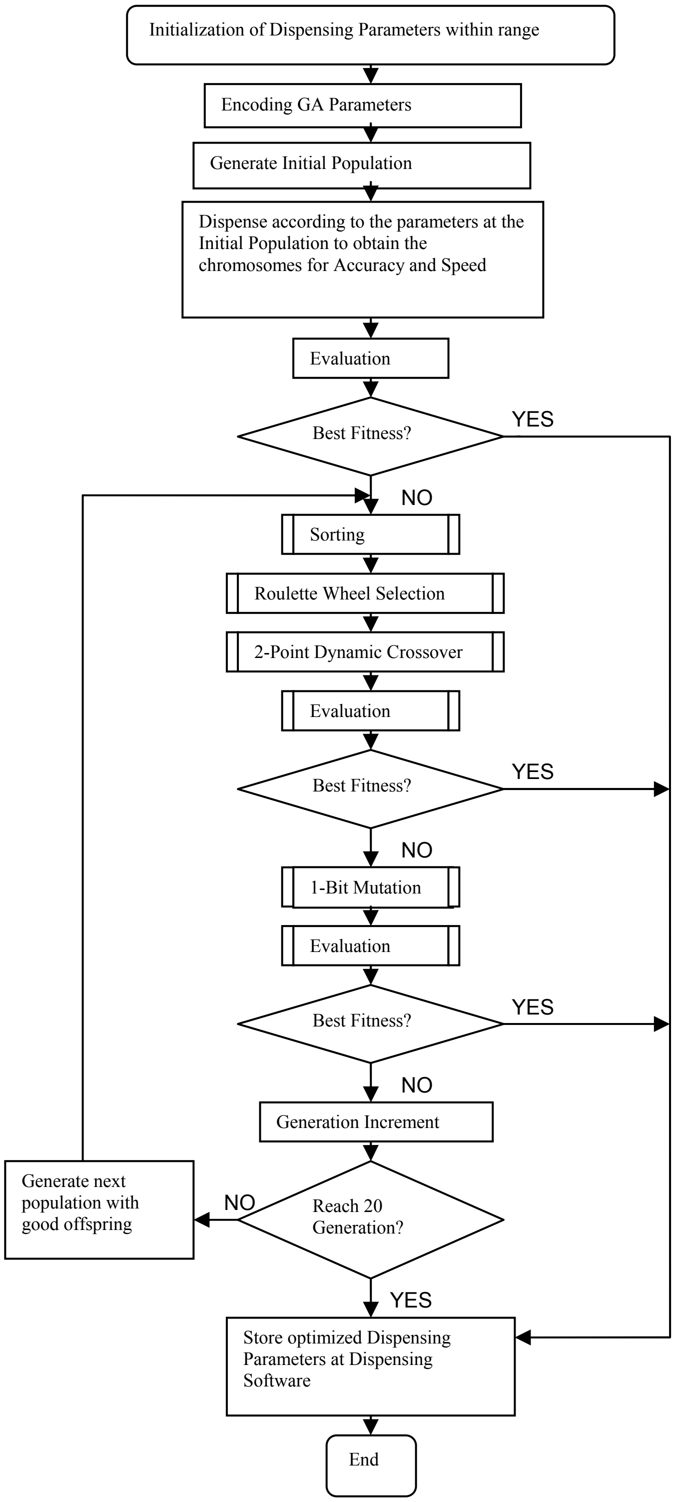

Figure 1: Flow Chart of the GA dispensing parameter fine tuning program

The exponential function is applied in the GA optimisation process so that the overall fitness function (*F*) is able to peak gradually. Coefficient *u* determines the sensitivity of the fitness function (*F*). In this application, coefficients *w* and *x* were set to -0.015 and 0.6 while coefficients *y* and *z* were set to -0.02 and 0.6. These settings are to prioritise accuracy over speed in the overall fitness function (*F*). To further fine-tune the overall exponential fitness, coefficients *u* and *v* are set to -0.3 and 0.45. The performance of *F* using different values of u and *v* are illustrated in Figures 2 to 4. From human's observation, the parameters shown give a clear indication on how good and bad genes are segregated as reflected by the overall fitness function (*F*). Fitter genes have higher probabilities of being selected while bad genes still stand a small chance to be selected instead of total rejection to keep the population size constant. After several generations, the algorithms converge to the best chromosomes, which are the optimum solution to the problem.

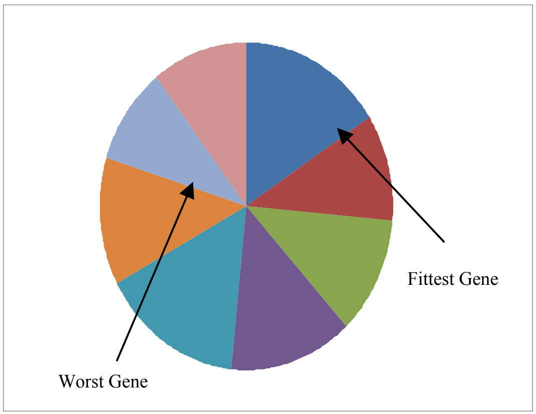

Figure 2: Performance of F with u = -0.2 and v = 0.3

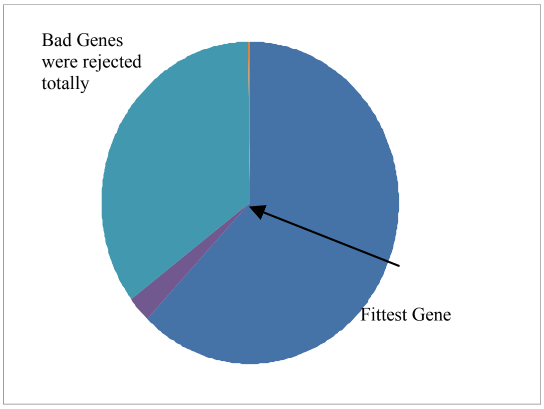

Figure 3: Performance of F with u = -0.4 and v = 0.6

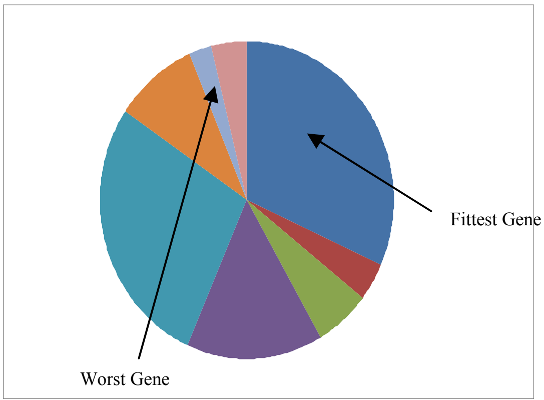

Figure 4: Performance of F with u = -0.3 and v = 0.45

3. Experimental Results

The dispensing parameters are fine-tuned using GA to obtain the optimum dispensing parameters. The population is selected randomly within the preset range for all the parameters. The system is then dispensed according to the array of chromosomes which represent the solutions to obtain the primary fitness function. Thereafter, the primary fitness function crosses-over and mutates to obtain the best fitness in terms of accuracy and speed.

A number of tests were conducted to assess the performance of the developed GA. The maximum generation was initially set to 40, with the population size at 10 along with 80% crossover rate and 1-bit mutation of the good solution. From the result shown in Figure 5, it is observed that a maximum generation of 20 was good enough to have high convergence rate. Besides, it is also observed that the fitness of the chromosome increases when the number of maximum generation increases. In each generation, fittest chromosomes (solutions) are selected as the parents to produce the offspring (new solutions) that inherited the strength of both parents. The offspring would then replace the bad chromosomes which improves the solution consequently.

The fitness of the solution is proportional to the population size. A better solution is obtained with a larger population size. A larger population size offers the opportunity to search a wider area of the solution space to obtain the optimum solution. A number of tests were conducted to assess the performance of population size which is illustrated in Figure 6. It is observed that all 3 sets of individuals provide similar fitness when the population size increases. The good population size is about 10 to 20. However the population size should not be too large since it would affect the speed of the optimized dispensing time. Therefore, a population size of 10 individuals was selected taken into account of computational time.

A number of tests were conducted to assess the performance of the crossover rate within its typical range of 0.45 to 0.96. The crossover rate would determine the frequency when crossover operation occurs. The performance of crossover rate is illustrated in Figure 7. It is observed the accuracy would increases when the crossover rate increases and therefore result in better solution. As a result of that, a crossover rate of 80% is selected.

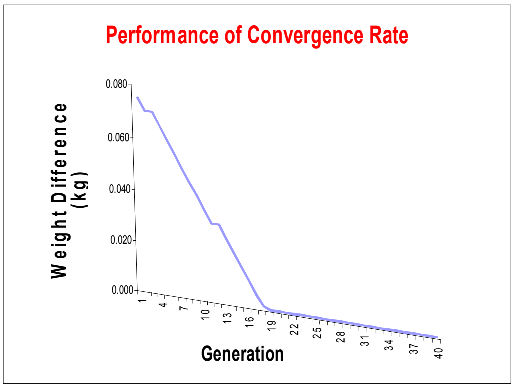

Figure 5: Performance of Convergence Rate

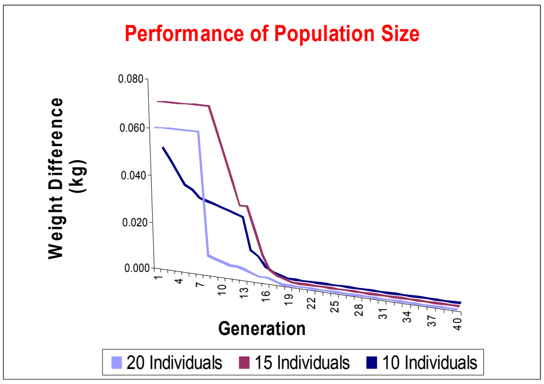

Figure 6: Performance of Population Size

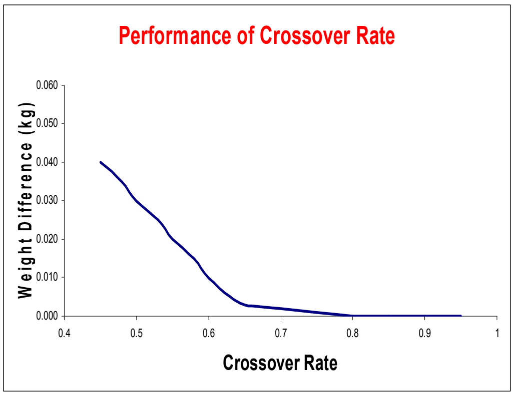

Figure 7: Performance of Crossover Rate

A number of tests were conducted to assess the performance of the mutation size. The mutation operator arbitrarily alters randomly a selected chromosome by restoring the unexpected genetic material into the mutation routine. Based on the computational and analysis data depicted in Table 2, 1-bit mutation was implemented as it represents the highest improvement in terms of accuracy.

Table 2: Computational and Analysis Data on Mutation Operator

	Improvement in accuracy per test							
Number of Bit mutated	1	2	3	4	5	6	7	8
1	X	X	√	X	X	√	X	√
2	X	X	√	X	X	X	X	X
3	√	X	X	X	√	X	X	X
4	X	X	X	√	X	X	X	X
5	X	X	X	X	X	X	√	X

where √ : Accuracy of +/- 2g
X: Accuracy of > 2g

4. Conclusions

Tests depicted above were conducted to assess the performance of genetic operators. It is observed that most of the fitness (solution) is obtained by crossover operators, followed by random generation and mutation operators. Crossover operators achieved a success percentage of about 75%, followed by 20% of random generation and 5% of mutation. The mutation rate is always kept to the minimum as the occurrence of mutation is also very rare in nature. The performance of the genetic operators is illustrated in Figure 8. Good agreement has been found between a proposed performance of the genetic operators and the experimental results.

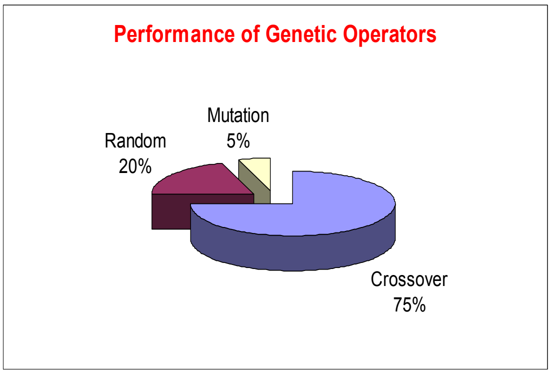

Figure 8: Performance of Genetic Operators

5. References

[1] Vale-Tech official website. http://www.vale-tech.co.uk/.

[2] Stork Prints official website. http://www.storkprints.com/.

[3] Nova Flow official website. http://www.novaflow.com/.

[4] Rexson official website. http://www.rexson.co.uk/.

[5] GFI Innovations official website. http://www.gfiinnovations.com/.

[6] Goldberg D., Richardson J. "*Genetic Algorithm with Sharing for Multimodal Function Optimization*", In Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications, pp. 41-49, 1987.

[7] David S.J., Lyle A.M. "*The Traveling Salesman Problem: A Case Study in Local Optimization, Local Search in Combinatorial Optimization*", E. H. L. Aarts and J. K. Lenstra (eds.), John Wiley and Sons, pp. 215-310, 1997, London, U.K.

[8] Lee Y., Stanislaw H.Z. "*Designing a genetic neural fuzzy antilock-brake-system controller*", IEEE Transactions on Evolutionary Computation, pp. 198-211, 2002.

[9] Jensen M. "*Generating robust and flexible job shop schedules using genetic algorithms*", IEEE Transactions on Evolutionary Computation, pp. 275-288, 2003.

[10] Chong K.H., Aris I.B., Senan M.A. "*The Optimization of Digital Circuit Structure with the Application of Evolutionary Algorithm*", AIML'06, International Journal on Automation and Intelligent Machine Learning, vol. 6, no. 3, 2006.