Title	4B1 DEVELOPMENT OF NEW ENCODING METHOD FOR THE COMPLEX PRODUCTION SYSTEMS(Technical session 4B: Genetic algorithm and Agent based-system)(本文(Fulltext))
Author(s)	AL-RAHMAN, Al-Momani Abd; ABU, Qudeiri Jaber E.; YAMAMOTO, Hidehiko
Citation	[Proceedings of International Symposium on Scheduling] vol.[2006] p.[94]-[99]
Issue Date	2006-07-18
Rights	Japan Society of Mechanical Engineers (社団法人日本機械学会)
Version	出版社版 (publisher version) postprint
URL	http://hdl.handle.net/20.500.12099/31861

International Symposium on Scheduling 2006
July 18 - 20, 2006, Tokyo, Japan
4B1

DEVELOPMENT OF NEW ENCODING METHOD FOR THE COMPLEX PRODUCTION SYSTEMS

Al-Momani Abd Al-Rahman [1], Jaber E. Abu Qudeiri [2], and HidehikoYamamoto [3]

[1] *B.Sc. in Computer and Control Engineering, Jordan.*

[2,3] *Intelligent Manufacturing Systems Laboratory*
Faculty of Engineering, Gifu University
1-1 Yanagido, Gifu Shi, 501-1193, Japan

[1] *momanki@just.edu.jo,* [2] *k3812203@guedu.cc.gifu-u.ac.jp,* [3] *yam-h@cc.gifu-u.ac.jp*

Abstract

The buffer size decision for production lines gains more and more importance because of growing production lines complexity and the production costs. Many researches have developed many techniques for getting the optimal buffer size which can aid in achieving maximum production rate. But some techniques were unable to deal with the complex production systems, and some were able to treat with complex production system but with a small growing rate of fitness, which means the need for a high number of generations to reach the optimal fitness. This paper aims to determine the optimal buffer size with minimum number of generations.

Keywords: Complex Production System, Multi Vector Encoding Method, Buffer Size, GA system, production simulator

1. INTRODUCTION

The buffer size decision for production lines gains more and more importance because of growing production lines complexity and the production costs. Many articles and researches related to the buffer size have been published (Fulya and Bulgak., 2002; Bulgak , *et al.*, 1995; Alabas, *et al.*, 2002; Hillier, *et al.*, 1993; Gershwin and Schor, 1997). One of the methods used for studying the buffer size in production lines is genetic algorithm (GA) (Diomidis and Chrissoleon, 2000; Goldberg, 1989; Forrest, 1996). GA is an evolutionary technique that uses crossover and mutation operators to solve optimization problems using a survival of the fittest idea. GA has been used successfully for various optimization problems, such as the buffer size problem.

The first and most important step in GA is encoding. The conventional GA operation is generally based on an individual with a linear gene arrangement, which represents the genes as a single line (vector) inside the individual. But for some Complex Production Systems (CPS) it is difficult to use a linear arrangement of genes in individuals to find a buffer size. This is because the CPS buffers are arranged in more than one line, and it is impossible to express such arrangement with conventional gene arrangement methods.

Another technique for solving this problem called Matrix Encoding Method, which is based on a matrix arrangement for the genes in the individual by dividing the CPS into cells and encoding the buffer in its matched cell in the matrix.

For some complex problems, the conventional type of crossover and mutation needs a lot of generations to reach the best answer (individual). This paper introduces a new method for GA encoding of a CPS called: Multi-Vector Encoding Method (MVEM). MVEM achieves the efficient use of the GA. Before going through the proposed MVEM the conventional encoding methods are explained in the following section.

2. CONVENTIONAL ENCODING METHODS

2.1. Linear Encoding

The Linear Encoding Method is the simplest case of encoding and it's used in many articles (Emanuel, 1998; David E. Goldberg, 1989; Homaifar, et al. 1992; Wroblewski, 1996). In linear encoding method, the individual is represented as a single line and we are dealing with this line as a chromosome. The individual can be represented as follows:

$$\begin{bmatrix} G_1 & G_2 & G_3 & \dots & G_{N-1} & G_N \end{bmatrix}$$

where N is the length (number of genes in the individual).

2.2. Matrix Encoding

In Matrix Encoding Method (Homaifar, *et al.*, 1992; Zbigniew, 1994), the CPS can be represented as a matrix contains a number of columns and rows depending on the architecture of the CPS. In this method the crossover and mutation is applied for the column of genes instead of the single gene in the linear encoding method.

3. MULTIPLE VECTORS ENCODING METHOD

This paper aims to determine optimal or near optimal buffer size and at the same time, to minimize the number of generations to reach this optimal buffer size for the CPS. The characteristics of the GA modified for our study are described in sections 3.1 – 3.4.

3.1. Encode

Here, in MVEM we divide the individual to multiple vectors; each vector contains part of individual which is representing a part of CPS. According to this arrangement method the buffer size arrangement for CPS can be defined as shown in Figure 1.

$$\left\{\begin{matrix} B_{1,1} & B_{1,2} & \cdots & B_{1,N_1} \\ B_{2,1} & B_{2,2} & \cdots & B_{2,N_2} \\ \vdots & \vdots & \vdots & \vdots \\ B_{K+11,1} & B_{K+11,2} & \cdots & B_{K+11,N_{K+11}} \end{matrix}\right\}$$

Fig. 1: Buffers Size Expressed By MVEM

where $K+11$ is the number of vectors in the individual, which is the number of lines in CPS, and N_i, $i=1,2,..,K+11$ is the number of buffers in the vector i. Noting that each buffer represents a gene in the individual, thus, the individual can be encoded as shown in figure 2.

$$\text{Individual} = \left\{\begin{matrix} V_1: & \text{For} & \text{line} & 1 \\ V_2: & \text{For} & \text{line} & 2 \\ \vdots & & & \\ V_n: & \text{For} & \text{line} & n \end{matrix}\right\}$$

which gives:

$$\left\{\begin{matrix} G_{1,1} & G_{1,2} & \cdots & G_{1,N_1} \\ G_{2,1} & G_{2,2} & \cdots & G_{2,N_2} \\ \vdots & \vdots & \vdots & \vdots \\ G_{K+11,1} & G_{K+11,2} & \cdots & G_{K+11,N_{K+11}} \end{matrix}\right\}$$

Fig. 2: Individual Expressed By MVEM

Noting that $N_1, N_2, ..., N_{K+11}$ are not equals, which it mean that this representation can deal with any CPS.

3.2. Initial Population

The initial population is randomly selected. The initial population contains (N) number of individuals. Each individual expresses genes as shown in equation 1 below:

$$G_{ij} = rand(0 \to S) \ \forall \ D_n, \tag{1}$$

$n=1,2,...,D$, $i=1,2,...,K+11$ and $j=1,2,...,N_{K+11}$ where:

G_{ij} : is the gene number j in vector number i.

S : is the maximum buffer size,

D_n : is the individual number n.

3.3. Crossover

In MVEM we use different arrangements for the genes from the other conventional arrangement methods. Moreover, we will use a different method for crossover and mutation now. The crossover of our GA's is carried out using the following steps:

1 - Two individuals are selected randomly from the current population.

2 - n Numbers of crossover points are selected randomly as follows:

$$P_i = rand(1 \to L_i), \quad i=1,2,...,n \tag{2}$$

Where:

P_i : is the cross point for the vector V_i.

L_i : is the length (number of genes) of vector V_i.

n : is the number of vectors in the individual.

3 - Each gene in the interval [P_i, L_i] in the vector V_i of the first individual swapped with the matching gene in vector V_i of the second individual. For example, in our CPS if we select a crossover points $P_1, P_2, ..., P_{K+11}$, then the selected individual can be represented as shown in Figure 3.

$$\left\{\begin{matrix} G^1_{1,1} & G^1_{1,2} & \overset{P_1}{\downarrow}\cdots & G^1_{1,N_1} \\ G^1_{2,1} & G^1_{2,2} & \overset{P_2}{\downarrow}\cdots & G^1_{2,3} \\ \vdots & & & \\ G^1_{K+11,1} & G^1_{K+11,2} & \overset{P_{K+11}}{\downarrow}\cdots & G^1_{K+11,N_{K+11}} \end{matrix}\right\}$$

Individual 1

$$\left\{\begin{matrix} G^2_{1,1} & G^2_{1,2} & \overset{P_1}{\downarrow}\cdots & G^2_{1,N_1} \\ G^2_{2,1} & G^2_{2,2} & \overset{P_2}{\downarrow}\cdots & G^2_{2,3} \\ \vdots & & & \\ G^2_{K+11,1} & G^2_{K+11,2} & \overset{P_{K+11}}{\downarrow}\cdots & G^2_{K+11,N_{K+11}} \end{matrix}\right\}$$

Individual 2

Fig. 3: Individuals Before Crossover (MVEM).

and after applying the crossover at all crossover points, the two selected individuals will be as shown in Figure 4.

$$\begin{Bmatrix} G^1_{1,1} & G^1_{1,2} & \cdots & G^2_{1,N_1} \\ G^1_{2,1} & G^1_{2,2} & \cdots & G^2_{2,3} \\ \vdots & & & \\ G^1_{K+11,1} & G^1_{K+11,2} & \cdots & G^2_{K+11,N_{K+11}} \end{Bmatrix}$$

Individual 1

$$\begin{Bmatrix} G^2_{1,1} & G^2_{1,2} & \cdots & G^1_{1,N_1} \\ G^2_{2,1} & G^2_{2,2} & \cdots & G^1_{2,3} \\ \vdots & & & \\ G^2_{K+11,1} & G^2_{K+11,2} & \cdots & G^1_{K+11,N_{K+11}} \end{Bmatrix}$$

Individual 2

Fig. 4: Individuals After Crossover (MVEM).

3.4. Mutation

As same as the crossover, we also use a different technique for mutation in our MVEM. The mutation technique is carried out using the following steps:

1 - One individual is randomly selected from the current population.

2 - n Couples of mutation points are randomly selected as follows:

$$P_{i1} = rand(1 \to L_i), \quad i = 1,2,\ldots,n$$
$$P_{i2} = rand(1 \to L_i), \quad i = 1,2,\ldots,n$$

Where:

P_{i1}, P_{i2} : are the two mutation points for the vector V_i.

L_i : is the length (number of genes) of vector V_i.

n : is the number of vectors in the individual.

3 - If $P_{i1} = P_{i2}$, reselect P_{i2}.

4 - For each vector V_i, swap the genes in the two mutation positions P_{i1}, P_{i2}

4. CASE STUDY

4.1. Typical Structure of CPS and model assumption

Typical Structure of Complex Production System introduced by (jangishan, 2003 a and b) is used in this application example. Figure 5 shows a typical structure of CPS, where the circles represent machines and rectangles are buffers. The system consists of one main line, one feeder line (assembly), K parallel lines, one feed-forward line and one rework loop. Subscripts a, f, r are used to denote the machines or buffers in feeder line, feed-forward line and rework loop, respectively.

A description of the notations of machines and buffers is introduced below:

Main Line :	$m_1,\ldots,m_p,m_q,\ldots m_M$
	$B_1,\ldots,B_p,B_q,\ldots,B_{M-1}$
Feeder Line:	$m_{a1},\ldots,m_{aA},B_{a1},\ldots,B_{aA}$
Feed-Forward Line:	$m_{f1},\ldots,m_{fF},B_{f1},\ldots,B_{fF+1}$
Rework loop :	$m_{r1},\ldots,m_{rR},B_{r1},\ldots,B_{rR+1}$
Parallel lines :	m_{ij},B_{ij} $(i=1,\ldots,k,\ j=1,\ldots,M_i)$
Merge machines :	m_a : assembly merge,
	m_{jr} : rework merge,
	m_{jf} : feed-forward merge;
Split machines :	m_r : rework split,
	m_f : feed-forward split,
	m_{rs} : scrap;

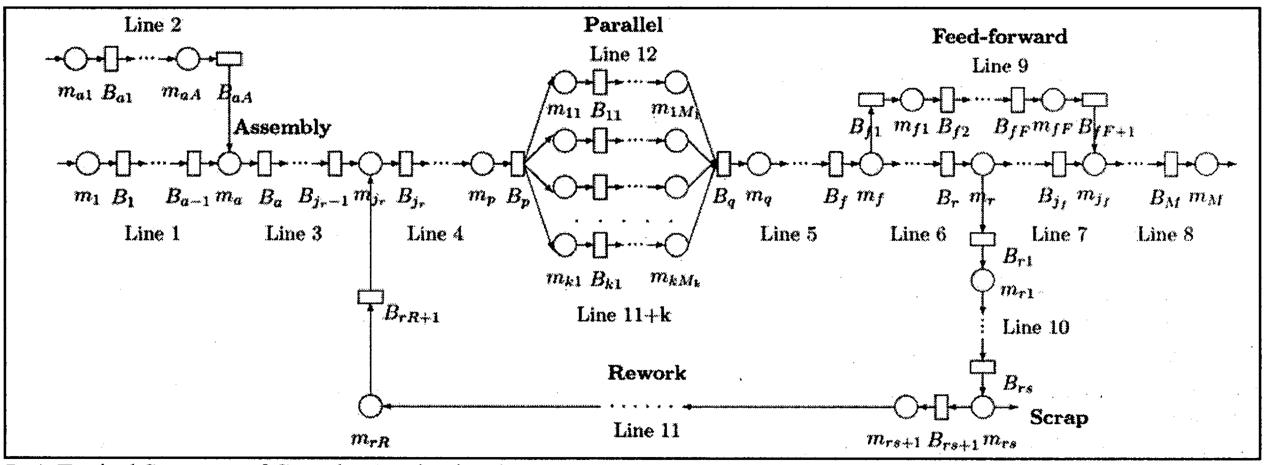

Fig. 5: A Typical Structure of Complex Production System .

For this system, the following assumptions are adopted:
1 - Each machine has two states: up and down. When up, machines in main line, feeder line, feed-forward line and rework loop, each of them is capable of producing with the rate S parts per unit of time. For machines in parallel lines, each machine, $m_{ij}, (i=1,...,k, j=1,...,M_i)$, in parallel line i, can produce with the rate S_i parts per unit of time when the machine is up. Moreover, $\sum_{i=1}^{k} S_i = S$. When the machine is down, no production takes place.
2 - Each buffer $B_i, \forall i$, has a finite capacity N_i, $0 < N_i < \infty$.
3 - A machine is starved at time t if upstream buffer is empty at time t. Machines m_1 and m_{a1} are never starved. In addition, machine m_a is starved if either buffer, B_{a-1} or B_{aA}, is empty. It is assumed that machine m_{jr} always takes part from B_{rR+1} (rework loop) first if it is not empty. Moreover, Machine m_{jf} can take parts either from B_{jf} or B_{fF+1} without any priority.
4 - A machine is blocked at time t if down stream buffer is full at time t. Machine m_M is never blocked. Machine m_r is blocked by main line if it produces a good part and buffer B_{r+1} is full, and blocked by rework loop if a defective part is produced and B_{r1} is full. Analogously, machine m_f can be blocked by main line and feed-forward line correspondingly.
5 - A part is defective with probability α at machine m_r, $0 \le \alpha < 1$. At machine m_f, a part has probability β, $0 \le \beta < 1$ to be sent to feed-forward line. A severe defective part is scraped with probability γ at machine m_s, $0 \le \gamma < 1$. Probabilities α, β and γ are referred to as the rework, feed-forward and scrap rates, respectively.
6 - The first machine in each parallel line, machine m_{i1}, $i=1,...,k$, has equal probability of taking a part from buffer B_p if it is not blocked, and the last machine in each parallel line, machine m_{iM_i} has equal probability of sending a part to buffer B_q if it is not starved.

4.2. Application example

Referring to the CPS shown in Figure 5 and noting that the symbols used here are refer to the same Figure, we assume that:
1- Number of machine tools as follows;
Line 01: 20 machine tools
Line 02: 20 machine tools
Line 03: 15 machine tools
Line 04: 10 machine tools
Line 05: 10 machine tools
Line 06: 10 machine tools
Line 07: 10 machine tools
Line 08: 20 machine tools
Line 09: 20 machine tools
Line 10: 5 machine tools
Line 11: 10 machine tools
Lines 12, 13, 14, 15, 16 and 17 have 5 machine tools for each. Here we choose $k = 5$.
2- The parts input into line 1 and line 2 are decided by one to one method (Yamamoto, 2000), the number of parts 10 input into line 1 and 10 input into line 2.
3- The machine time for each part (for the 20 parts) in each machine is chosen between 15-20 second.
4- Each machine tool stops 6 times an hour, and stopping time is 15 sec.
5- Each machine tool stops for a quality check every 100 parts. Stopping time =15 sec.
6- FTL cycle time is 20 sec.
7- Working time is 8 hours. The production ratio for each part is assumed to be between 50 and 100.
8- The maximum buffer capacity is 50.
9- A part is defective with probability $\alpha = 0.2$ at machine m_r. At machine m_f, a part has probability $\beta = 0.5$ to be sent to feed-forward line.
10- A severe defective part is scraped, if the part circles for the fourth time in the rework path.

4.3. Production simulator

To find the buffer size in front of the bay of machine tools in the CPS that maximizes the production efficiency of the FTL, we propose a PS. The developed PS consists of a GA and a discrete simulator as shown in Figure 6. PS searches the buffer size in front of the bay of each machine tool in the CPS by repeating discrete simulation and utilizing the GA. The Program using the C++ language is used to establish the PS. When the simulation is started the PS reads the specified production system characteristics, such as a number of machine tools, machine tools stops, machining time for each product in each machine tool, production ratio, number of defective products ,...etc, in the desires CPS. The CPS defined by assumptions (1-10) in section 4.2 is input into the PS, based on this input, the PS searches the buffer sizes that maximize the throughput of the FTL.

The relationships between the GA and the discrete simulator of PS are as follows: The initial individuals corresponding to a buffer size in front of each machine tool are randomly selected and are input into the discrete simulator. The discrete simulator simulates many hours' productions and finds the production efficiency. After the GA operations in conjunction with the results of the discrete simulator, the next generation's individuals are generated.

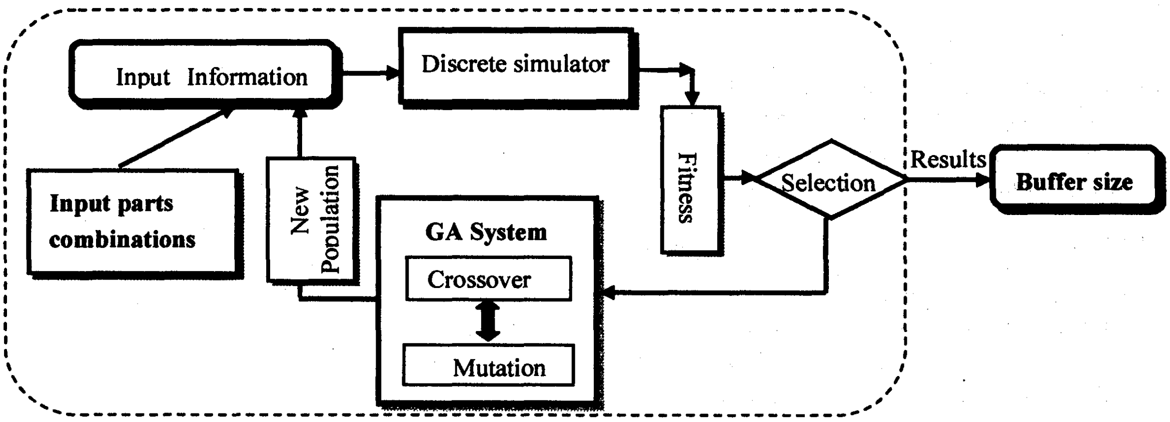

Fig 6. Genetic algorithm and simulation evaluation

The new individuals are input into the discrete simulator again. In this way, the cycle including the GA system and the discrete simulator is repeated until the better fitness is found.

4.4. CPS Performance Evaluation

To measure the production efficiency of this system, the PS find the fitness (F) for each individual, which defined by the following relations:

$$F = \frac{\text{Number of Products}}{\text{Theoretical Number Of Products}} \quad (1)$$

Where:
Number of products: is the Final number of products at the end of the simulation (Real time operation).
Theoretical number of products is given by the following relation:

$$\text{Theo.}\, P = \frac{T}{T_{mc}} - M \quad (2)$$

Where:
Theo. P : Theoretical number of products.
T : Working time.
T_{mc} : Machine cycle time.
M : Total number of machines.
Substituting (2) in (1) we have:

$$F = \frac{P}{T / T_{mc} - M} \quad (3)$$

Where P is the number of products (Real time operation).

4.5. Results

It is clear that the linear encoding method is the least efficient. So, we applying the Matrix Encoding Method and the MVEM and making the simulation to compare these two methods and we ignore the Linear Encoding Method.

Figure 7 shows the best fitness curves for the matrix encoding method and the MVEM. Table [1] shows the buffer size of the CPS when the fitness reaches it maximum value.

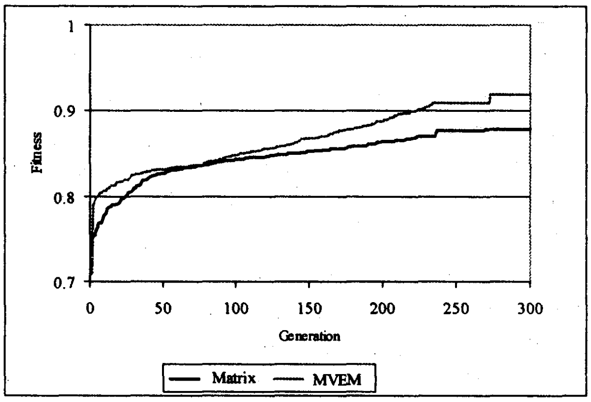

Fig 7. Best fitness curves for the matrix encoding method and the MVEM

5. CONCLUSIONS

This study described the MVEM for a CPS. MVDM for genes is adopted for the genes arrangement in each individual. The study proposed the PS to decide a buffer size. PS based on the GA and searches for the better buffer size, that can aid in achieving maximum production rate.

We applied PSS to some CPS examples. After a number of generations, the best size of the buffer of the CPS could be found. The results of the study can be used to improve the production plan, and production engineers can use these results when making decisions on a buffer size.

It was found that using the MVEM led us to an optimal buffer size with less number of generations.

Table [1]: Buffer size resulted by PS based on the MVEM

Line	Buffer size*				
L1	1 3 4 7 9 10 2 6 9 4 1 8 10 1 1 4 10 2 3				
L2	2 7 8 6 3 8 1 9 1 5 6 1 9 6 1 5 10 9 5 9 8				
L3	10 3 10 2 2 4 3 8 2 10 3 4 8 2 6				
L4	2 2 10 10 8 7 7 8 7 10 9				
L5	2 5 7 8 6 1 10 7 8 7				
L6	9 1 2 7 6 9 10 3 6 7				
L7	9 10 7 6 4 1 2 8 9 9				
L8	6 1 3 4 2 2 3 1 9 4 2 2 8 10 9 2 4 6 2 10				
L9	1 8 3 8 5 6 5 2 2 1 1 3 8 5 9 5 10 5 2 1 5				
L10	4 2 9 2 8				
L11	6 5 3 1 9 1 9 3 4 5 3				
L12	5 1 6 7	**L13**	9 3 2 10	**L14**	2 6 8 7
L15	2 5 2 10	**L16**	5 7 9 2	**L17**	9 9 1 1

* The buffer size of each line arranged respectively

References

Alabas C., F. Altiparmak and B. Dengiz, 2002. A comparison of the performance of artificial intelligence techniques for optimizing the number of kanbans. *Journal of the Operational Research Society*, 53(8):907-914.

Bulgak A. A., P. D. Diwan, and B. Inozu, 1995. Buffer Size Optimization in Asynchronous Assembly Systems Using Genetic Algorithms. *Computers and Industrial Engineering*, 28(2):309-322.

Diomidis D., Chrissoleon T., 2000. Stochastic Algorithms for Buffer Allocation in Reliable Production Lines. *Mathematical Problems in Engineering*, 5: 441–458.

Emanuel Falkenauer. *Genetic Algorithms and Grouping Problems*. JohnWiley and Sons, 1998.

Forrest. S. Genetic Algorithms. *ACM Comput. Surv.*, 28(1):77-83, Mar. 1996.

Fulya Altiparmak, Akif A. Bulgak,2002. Optimization of Buffer Sizes in Assembly Systems Using Intelligent Techniques. *In Winter Simulation Conference*, 1157-1162, San Diego, CA., USA.

David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

Gershwin S. and J. Schor, 1997. Efficient Algorithms for

Buffer Space Allocation. *In International Workshop on Performance Evaluation and Optimization of Production Lines*, pages 217-228, Samos, Greece, University of the Aegean, Department of Mathematics, May 1997.

Goldberg David E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

Hillier F. S., K. C. So, and R. W. Boling,1993. Notes:

Toward Characterizing the Optimal Allocation of Storage Space in Production Line Systems with Variable Processing Times. *Management Science*, 39(1):126-133.

Homaifar Abdollah, Shanguchuan Guan, and Gunar E. Liepins. *Schema analysis of the traveling salesman problem using genetic algorithms*. Complex Systems, 6(2):183-217, 1992.

Jingshan Li, "Overlapping Decomposition: A System-Theoretic Method for Modeling and Analysis of Complex Production Systems", *Technical Report*, General Motors Research & Development Center, 2003.

Jingshan Li, Modeling and Analyses of Complex Production System http://www.icsd.aegean.gr/aic2003/ Papers/Li.pdf.

Wroblewski Jakub. *Theoretical foundations of order-based genetic algorithms*. Fundamenta Informaticae, 28(34):423-430, 1996.

Yamamoto, H.. One-by One Parts Input Method by off-line Production Simulator System with GA. European Journal of Automation, Hermes Science Publication, pp. 1173 – 1186, 2000.

Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 2nd edition, 1994.