Viral Systems Implementation for Minimizing Mean Tardiness of Job Shop Scheduling Problem

Alfian Tan, Dedy Suryadi
Parahyangan Catholic University - Indonesia
Email: alfian.tan@gmail.com, dedynamo@yahoo.com

Abstract: This research is conducted to develop a heuristics based algorithmto find a satisfying solution to job shop scheduling problem with paralel machine. Heuristics is widely used in optimization problems because its efficiency and simplicity. Viral Systems is a heuristics that imitates virus life cycle to find a good solution to a certain problem. We develop five interrelated-Viral Systems-based algorithms to solve the scheduling problem. Mean tardiness is used to measure the goodness of a schedule. These five algorithms are implemented to 2 hypothetical cases.There are several parameters tested in each case. The results show that there is only one parameter which gives significant effect to the first case solution but none for the second case. These algorithms are also compared to other heuristics, which are Genetic Algorithm with Multistep Crossover and Artificial Immune System. The results show that Viral Systems has the lowest performance among them. One possible weakness of this method is ineffective mutation method used.

Keywords: Viral systems, heuristics, scheduling, mean tardiness

Introduction

Scheduling can be explained as an activity of allocating a number of jobs to a number of available resources in a specific order. The result of this activity is information about when to start doing the job and when the job should be completed. There are many kinds of scheduling activities in industry. One of them is job shop scheduling with parallel machine. Typically, job shop scheduling problem consists of several jobs with different process orders and also different machines/resources used. The number of machines available could be more than one unit so that we have to consider the process predecessor of a certain job and the other jobs as well. Scheduling is really important in industry. This activity is influential for organization productivity. A good schedule will increase efficiency of doing jobs and it will give more profit to the organization directly. In the past, scheduling is done in order to minimize production duration and maximize resource utilization. However, it has different goals now, punctuality in fulfilling customer orders are one of critical factor to satisfy customer (Kolahan and Kayvanfar[5]).

There are many methods that have been developed to prevent tardiness of fulfilling orders. Heuristics is one of them. According to Reeves [7], heuristics is a technique that seeks a good solution at a reasonable computational cost without a guarantee either it is an optimal or feasible solution and how close the solution to the optimal one. Based on that definition, we know that heuristics is able to give a good solution efficiently so that it is widely used to solve complex problems. A good solution is not necessarily an optimal one but it can satisfy the decision maker. Sometimes, this matter is claimed to be the weakness of heuristics. Optimization methods could be used to give an optimal solution but it is not efficient enough to be used in complex problems. In this paper, we consider a new heuristics called Viral Systems. This method is quite new so there have been just a few researches about its application by now. Success in the first implementation of this method (Cortés *et al.*[2]) encourages the writer to adopt it to another problem which is scheduling activity. In this paper, we implement viral systems model to solve job shop scheduling with identical parallel machine problem. We use mean tardiness as a criteria to measure how good the solution is so that the objective function of this problem is to minimize mean tardiness of the schedule. Not only an application of viral systems in scheduling problem, we also want to know whether there are significant effects of parameters to the solution produced.

There are some other heuristic algorithms that have been developed to solve scheduling problem. There are Genetic Algorithm with Multisteps Crossover, Artificial Immune System Algorithm, Artificial Neural Network, and Particle Swarm Optimization. In order to give more insight about how heuristics works, we make some comparisons among several algorithms. We compare Genetic Algorithm with Multistep Crossover (Librata[6]), Artificial Immune Systems (Gandajaya[3]), and Viral Systems in solving a certain scheduling case.

Literature Review

Job Shop

Job shop is a characteristic of machine environment. In this case there is more than one process that has to be done in one job. Each job has different process route. This is one of several differences between job shop and flow shop production system. In flow shop environment, all jobs have the same process orders so that they will be the same as machines orders. Job shop problem can be classified into two categories based on the kinds of machines used in the process. There are pure job shop and general job shop. In a pure job shop, each process in a certain job has its own machine/resource. It means that one machine can be use only once by each job. It is different from general job shop where there is a possibility for a certain job to use a certain machine more than once. Baker [1] uses triplet notation (i,j,k) to describe state of a job in the job shop environment where letter 'i' is for job-i, 'j' is for the j^{th} operation/process of the i^{th} job, and 'k' is for the machine used for j^{th} operation of i^{th} job.

It is usual to use matrix in describing information about processing time and machine used for a certain process of a job. There are two kinds of matrix which are usually used. There are operation time matrix and routing matrix. Operation time matrix gives information about how long a certain process takes to be done and routing matrix show which machine is used for each process of a certain job. In job shop scheduling, there are many alternatives of schedule produced. According to Baker [1], all those schedules can be classified into 5 categories as follows:

1. Feasible schedule: every schedule which doesn't violate resource and precedence constraints of the job.
2. Semi-active schedule: a group of feasible schedules without any possibilities of doing local left shift process in the condition of maintaining other processes orders.
3. Active schedul: a group of feasible schedules without any possibilities of doing global left shift without delaying other processes.
4. Non-delay schedule: a group of feasible schedules where there is no machines idle when there is an operation waiting for the machine.
5. Optimal schedule: a schedule which has the best performance

Viral Systems

Viral systems is a heuristic method which imitates virus life cycle in organism. Generally, there are three steps to follow when we wants to adopt this model into a problem(Cortés *et al.*[2]):

1. Initialization process
   In this step, we decides which kind of infection used, initial solutions for next step iteration, and how to measure quality of a solution.
2. Steady state
   In this second step, the iteration is running. There are interactions among organism and viruses in term of finding a good solution. Cells in organism will produce antibody againts viruses, but there are cells that don't produce antibody so that they will be infected by viruses and finally will either collapse or mutate.
3. Ending
   This step is the final step of finding a good solution. There are two possible conditions of this step, which are succeed in finding the solution or otherwise. The first condition is analogous to death of organism while the second one is isolation of viruses in organism.

There are three components which get involved in viral systems. They are virus, organism, and interaction between them. Each component has their own attributes. Attributes of virus consist of state, input, output, and process. State shows us which cell in organism which is infected by the virus. Input shows the information got from the organism. The information could be neighbouring cells for a certain cell infected. Output shows a certain kind of virus replication which is done in infection state. Process shows virus behaviour which is able to change the state.

The second component is organism. This component consist of two attributes, which are state and process. State describes an organism in its initial condition. This condition could be known from clinical picture of the organism. Clinical picture is a group of cells infected which describe health of an organism. Process shows activity of an organism which produce antibody againts viruses. Each cell in clinical picture has probability of producing antibody so that they are immune to viruses and excluded from clinical picture.

The third component is interaction. Virus and organism will give a response to each other. Virus will infect a cell and the cell will probably respond by producing antibody. There are two possible virus life cycle in organism, which are lytic replication and lysogenic replication. Lytic replication is done by generating a number of new viruses. These viruses will infect other cells massively or selectively. Lysogenic replication is done by mutating a certain cell. Virus DNA will join DNA of a certain cell. This mutated cell will replace the original cell in the clinical picture.

Methods

In order to develop algorithms for scheduling problem, there are five steps that have been done. There are determination of performance indicator of a solution, determination of coding concept for the solution, determination of infection process type, development of algorithms and application of algorithms.

Performance Indicator

Performance indicator which is used to evaluate the goodness of solutions is mean tardiness. Tardiness is defined as a positive lateness of a job. Mean tardiness is calculated from all jobs' tardiness. According to this indicator, due date of a job will be a relevant input to this problem. The objective function is to minimize mean tardiness of the jobs.

Coding Concept

Coding conceptwhich is used in this research is called operation based representation (Gen and Cheng [4]). A job will be represented by a certain number. For instance, we have 3 jobs which have to be completed, using operation based representation we will use three different numbers for the jobs, number 1 for the first job, number 2 for the second job, and number 3 for the third job. There may be more than one appearance for a certain number in an operation based code. It depends on how many operations a certain job has. Figure 1 shows an example of operation based code for three jobs.

According to Figure 1, we know that each job has two operations to be completed. The priority of operations scheduling follows the order of the code. The first operations to be scheduled will be the first operation of job 3. The second one is the first operation of job 1. The third one is the first operation of job 2, and so on.

Infection Process Type

There are two kinds of virus life cycle; they are lytic and lysogenic replication. Lytic replication is done by producing new viruses so that when the infected cell collapses, the viruses will come out and infect other cells (neighbouring cells). There are two infection processes, which are massive and selective infection. Selective infection is done by infecting the unhealthiest cell (i.e. the best neighbouring solution). In order to know the quality of the cell (solution), we have to interpret this cell first. This infection process will result only one new solution with additional interpretation step in infection process.

Massive infection is a process that will result several infected cells. It will give us more options to explore or exploit. This way of infection is expected to give higher probability and higher speed in finding the best solution. It doesn't need to interpret the cell first so that it can save several steps in the algorithm. This developed algorithm will use massive infection to find other alternative solutions.

3	1	2	2	3	1

Figure 1. Example of Solution Code

Development of Algorithms

There are five algorithms which are developed for the problem. Figure 2 shows the whole algorithms developed and their connections. There are 5 kinds of inputs which is needed to find a solution. There are routing matrix, operation time matrix, due date of a job, number of machines, and viral systems parameter. Routing matrix shows an order of machines which will be used to complete all operations of a job. Operation time matrix shows the duration for completing an operation of a certain jobs. Due date of a job is used to calculate its tardiness. Number of machine is a relevant input because we are dealing with paralel machine scheduling. In this problem, there may be more than one unit of machine used. The last input is categorized into viral systems parameter value. This parameter value is expected to influence the quality of solutions.

There are nine parameters in viral system as below.
1. POB = clinical picture capacity
2. p_i = infection probability
3. p_r = replication probability
4. p_{lt} = lytic replication probability
5. p_{lg} = lysogenic replication probability which is 1 - p_{lt}
6. p_{an} = the probability of producing antibody
7. LNR^0 = limit of initial lytic replication
8. LIT^0 = limit of initial lysogenic replication
9. N_{max}/ITER = maximum number of iteration cycle

Input for p_{an}should meet the lower bound of its value so that we have to calculate this parameter's lower bound first.

In the development of algorithms, we use several variables as follow.
1. n = number of jobs
2. m_i = number of operations of the i^{th} job.
3. N = iteration number
4. $LNR_{cell\text{-}x}$ = maximum replication number of virus for the x^{th} cell
5. $LIT_{cell\text{-}x}$ = maximum lysogenic iteration number for the x^{th} cell
6. Initial NR = initial virus replication number of a cell (*lytic*)
7. Final NR = final virus replication number of a cell
8. Initial IT = initial iteration number of a cell (*lysogenic*)

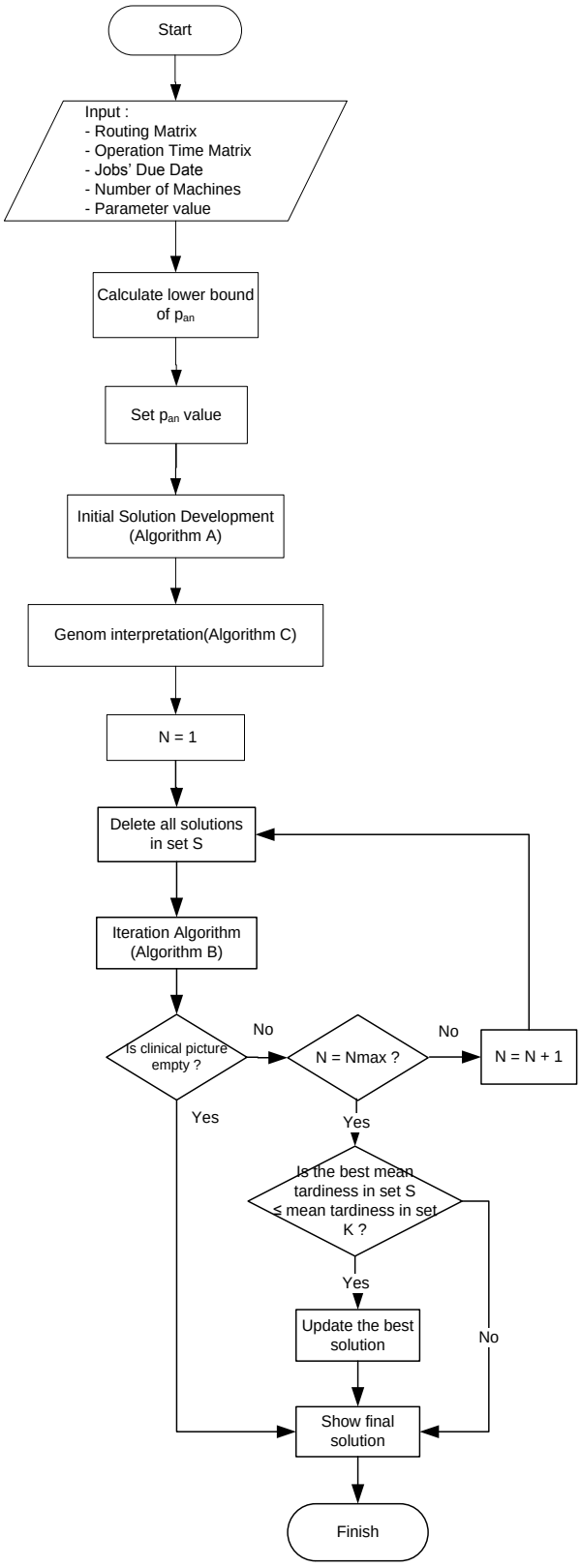

Figure 2. Flow diagram of iteration procedure

9. Final IT = final iteration number of a cell
10. R_{kl}= availability time of the l^{th} machine-k to be assigned.
11. R^* = the smallest availability time of a machine.
12. l^* = machine-k with the smallest availability time
13. C_{ij} = completion time of the j^{th} operation of job-i.
14. t_{ij} = processing time of the j^{th} operation of job-i
15. O = order of genes in a genom
16. {S} = a set of new solutions
17. {K} =a set of best solutions

Each algorithm which is developed in this research can be explained as follows.

1. Initial solution development (Algorithm A)
   Inital solution is generated randomly as many as clinical picture capacity. As mentioned before, we use operation based representation code for the solution. The randomized code is the job code. If there are three jobs, there will be number 1 until 3 to be randomized. The emergence of each number is limited by the number of processes of each job. It is determined that there must not be duplications for initial solutions. A series of code is called a genom. Each code in a genom is called a gene. It can be noticed from the example in Figure 1. The complete algorithm A can be shown as follows.
   1. Set x = 1
   2. Set O = 1.
   3. Encode the problem
      3.1 Generate a random numberU[0,1].
      3.2 Multiply the random number and n
      3.3 *Roundup* the result
      3.4 Check the emergence number of the round up value. If the emergence number of a certain value (representing job-i) $\leq m_i$, then place that value in the O^{th} order of the genom. Otherwise, back to step 3.1.
      3.5 Check the number of O. If $O = \sum_{i=1}^{n} m_i$ then continue to the fourth step, otherwise do another step 3 and set O = O + 1.
   4. Check the genom
      If the genom has been available in clinical picture, then delete it, otherwise put the genom in clinical picture.
   5. Check the value of x.Ifx = POB then stop the algorithm, otherwise set x = x +1 and go back to the second step.
   6. Setthe best *mean tardiness* for initial iteration (iteration) equals to a very big number (∞) .
2. Iteration algorithm (Algorithm B)
   This algorithm is started by comparing mean tardiness of each solution in clinical picture with the best value which has been obtained. If we have got a solution with zero mean tardiness, then the iteration will be stopped because there is no tardy jobs any more. Otherwise, we first record a better solution we've got into set K and continue the iteration. Solutions which are recorded in clinical picture will be deleted by the probability

of antibody (p_{an}). A random number will be generated for each cell in clinical picture. If the random number is less than p_{an}, then the cell will be excluded from clinical picture because it is assumed that the cell produces an antibody so that it can not be infected by viruses. This cell will leave an empty space in clinical picture. A second random number is generated for each remaining cell. This number will be compared with probability of lytic replication (p_{lt}). If the random number is less than p_{lt}, then the virus in that cell will have lytic replication. NR is an indicator of lytic replication in each cell. Every virus which has lytic replication in certain cell will increase the value of NR of that cell based on replication probability (p_r). NR will increase until it reach the maximum number of replication (LNR) and the cell will collapse and will be excluded from clinical picture. A number of viruses in the cell will infect neighbouring cell using algorithm D.

Lysogenic replication happens if the second random number generated is more than p_{lt}. Indicator for this replication is IT. Every cell has this indicator. The value of IT will increase by 1 point until it reaches the maximum number of lysogenic replication (LIT). When IT reaches LIT, cell will mutate using algorithm E and it will replace the original cell in clinical picture. New cells generated from algorithm D and E should be ensured not a duplication of other solutions. The number of new solutions will also be ensured no more than the capacity of clinical picture. It will be done by deleting the solution candidates randomly. The remaining new solution will be translated using algorithm C. These solutions will be included to clinical picture. If empty spaces in clinical picture are not enough to hold these new solutions, then the previous solutions in clinical picture will be deleted started from the worst solution until there are enough spaces for the new solutions. The complete steps of algorithm B can be shown as follows.

1. Determine the best solution in the iteration-N. If the best*mean tardiness* ofclinical picture-N ≤ the previous *mean tardiness*, thenupdate the best solution in set K. Otherwise, move to step 3. If there are more than one the best alternative solutions, choose the last alternative.
2. Checkthe best solution that we've got. If mean tardiness of this solution> 0, then move to step 3. Otherwise set $N = N_{max}$ so that the iteration will stop.
3. Determine the cells in clinical picture which have antibody.
   3.1 Generate a random number of U[0,1] for each cell in clinical picture.
   3.2 If the random number≤ p_{an} , then delete the cell

   Do these steps for all cells in clinical picture.
4. Determine virus replication type for each cell
   4.1 Generate a random number of U[0,1] for cell-x in clinical picture.
   4.2 If the random number ≤ p_{lt} ,then the virus in that cell will have lytic replication, otherwise move to step 4.3.
      4.2.1 Generate a random number of U[0,1]
      4.2.2 Calculate $LNR_{cell\text{-}x}$
      4.2.3 Calculate probability of replicating z number of viruses according to LNR of the cell with procedures below.
         4.2.3.1 Calculate the probability of replicating 0 viruses [P(Z=0)] until $LNR_{cell\text{-}x}$number of viruses [P(Z= $LNR_{cell\text{-}x}$).
         4.2.3.2 Sum up those probabilities, starting from P(Z=0) to P(Z=z) untill we get for the first time the sum up value which is greater than or equal to the random number in 4.2.1.
      4.2.4 Check the NR initial number of the cell
      4.2.5 Add up z viruses from 4.2.3.2.to the initial NR so that it becomes a final NR number for the cell.
      4.2.6 Check the final NR number
         If final NR≥ $LNR_{cell\text{-}x}$, thenfindneighbouring solutions using algorithm D and deletecell-x from clinical picture, otherwise update the NR value of the cell in clinical picture.
   4.3 Virus in the cell has lysogenic replication.
      4.3.1 Set final IT = initial IT + 1;
      4.3.2 Calculate $LIT_{cell\text{-}x}$
      4.3.3 Check final IT value
         If final IT ≥ $LIT_{cell\text{-}x}$, then do the mutation process using algorithn E and delete cell-x from clinical picture, otherwise update the number of IT.

   Do step 4 for all cells in clinical picture
5. Check the duplication among members of set S. Delete a member of set S which is the same as another member until there is no duplications in set S.
6. Compare each member of set S with solutions in clinical picture after replication process. If there is a same solution, then delete the solution from set S.
7. Check the number of solution in set S.
   If the number of new solutions ≤ POB then move to step 9, otherwise generate a random number of U[0,1] for each solution in set S.

Order the solution by its random number, delete x number of solutions with smallest random number until there are only POB number of remaining solutions.

8. Translate all new solutions in set S using algorithm C.
9. Check the number of empty gaps in clinical picture. If the number of empty gaps ≥ the number of new solutions, then put the all new solutions into clinical picture with NR and IT value equals to 0.Otherwise, delete several cells in clinical picture starting from the worst solution until there are enough spaces for the new solutions and then put all the new solutions into clinical picture.

3. Genom interpretation (Algorithm C)
Genom is a code of solution. To decide how good the solution is, we have to translate the genom into the objective function value. Genom will be translated from the first gene. Each code/gene and its occurence shows a certain operation/process in a certain job. The assignment of that process will consider which machine is used and processing time of the process which are available in routing and operation matrix. This assignment will consider the completion time of certain operation and availability of the machine. If the availability time of a machine is less than completion time of predecessor operations, then the operation will be assigned when the predecessor has been completed, otherwise the operation will be assigned when the machine is available.

The assignment of parallel machine will consider the machine load level. A machine with shortest completion time will be selected. If there are machines with the same completion time, then the first one will be selected. After completing the assignment, then the completion time of each job will be compared to its due date. Tardiness will happen if the completion time of a job is greater than its due date. Mean tardiness will be calculated by adding up all jobs' tardiness and dividing it by number of jobs. Here we get the objective function value of all alternative solutions. The steps of algorithm C can be shown as below.

1. Use the triplet notation concept (ijk) for the assignment process.
2. Set $R_{kl} = 0$.
3. Set $C_{ij} = 0$.
4. Set j = 0.
5. Set O = 1.
6. Do the jobin the O^{th} gene of the genom.
7. Set i = value of the O^{th} gene.
8. Check the last j value for job-i. Set j = j+1 for job-i.
9. Check the machine which will be used for the j^{th} operation of job-i from routing matrix. Set k = the machine number which will be used.
10. Set $R^* = \min \{R_{kl}\}$
11. Set l^* = machine-k which has R^*. If there are more than one machines with R^*set $l^*=1$.
12. Check t_{ij}value.
13. Check $C_{i(j-1)}$ value
    If $C_{i(j-1)} \leq R^*$thenput the j^{th} operation of job-i into the l^*th machine-k at $t = R^*$. Set $R_{kl^*} = C_{ij} = R^* + t_{ij}$. Otherwise, put it into the l^*th machine-k at $t = C_{i(j-1)}$. Set $R_{kl^*} = C_{ij} = C_{i(j-1)} + t_{ij}$
14. Check the O value. If $O = \Sigma m_i$ thencontinue to step 15. Otherwise, set O=O+1, and go back to step 6.
15. Calculate tardiness of each job.
16. Calculate *mean tardiness*

4. Neighbouring solution development (Algorithm D)
Viruses which come from a collapse cell will infect neighbouring cells. Neighbouring cells are generated by changing a pair of adjacent code (gene) from a certain genom so that in a genom with n number of gens, we can get at most n-1 number of neighbouring solutions/cells. These new cells will be infected by viruses with probability of p_i. A random number will be generated for each new cell, if this random number is less than p_i then the cell will be infected, otherwise the cell will be deleted. After these processes then we will generate second random numbers for the remaining neighbouring cells. If this number is less than p_{an} then the cell will be deleted. After this second step, the remaining cells will be the candidate of solutions and saved temporarily in a set of solutions (set S). According to Figure 1, two of six possible alternative solutions for that genom are as follows.

The first alternative solution is generated by changing the position of the first and the second gen, while the second one by changing the second and the third gen. The complete algorithm D can be show as follows.

1. Set O = 1.
2. Changethe O^{th} gene withthe$(O+1)^{th}$ gene
3. Generate the first random number of U[0,1]
4. If the random number ≤ p_i , thenkeep the *genom* and continue to step 5. Otherwise, delete the genom and then move to step 8.
5. Generate a second random number of U[0,1].
6. If the second random number ≤ p_{an} , then delete the genom and then move to step 8. Otherwise, go to step 7.
7. Keep the genom as a new solution in set S.

1	3	2	2	3	1

Figure 3.1st Alternative of Neighbouring Solution

3	2	1	2	3	1

Figure 4.2nd Alternative of Neighbouring Solution

2	1	2	3	3	1

Figure 5. Mutated Genom

8. Set O = O+1.
9. Check the O value. If O = $\sum_{i=1}^{n} m_i$ thenstop the D algorithm. Otherwise, go back to step 2.

5. Mutation algorithm (Algorithm E)
   Every cell with IT greater than LIT will mutate. Genom of a cell will mutate by changing a pair of gene randomly. This way of mutation is called reciprocal exchange mutation (Gen and Cheng [4]). The mutated genom will be saved in set S. Figure 5 shows an example of mutated genom . This code is generated by changing the position of the first and the fourth gene of genom in Figure 1.

The complete steps of algorithm D can be shown as follows.

1. Determine a pair of gene that will be changed
   1.1. Generate the first random number U[0,1].
   1.2. Multiply the random number with $\sum_{i=1}^{n} m_i$
   1.3. Roundup the value
   1.4. Generate a second random number U[0,1].
   1.5. Multiply the random number with $\sum_{i=1}^{n} m_i$.
   1.6. Roundup the value
   1.7. Check the second roundup value. If it has the same value with the first one then go back to step 1.4. until it results a different value.
2. Change the two genes in the genom according to the gene order numbers resulted from the first step.
3. Put the mutated genom in set S.

Application of Algorithms

This step is started by developing a program. The developed algorithms will be translated into programming language so that we have a program that can be used to test several scheduling cases. After we get a verified program, we develop a hypotetical case to use. Actually, there are two casesused. The first one consists of 10 jobs, 88 operations, and 7 kinds of machine with 2 units each. The second one consists of 30 jobs, 300 operations, and 10 machines with 2 units each. The second case is cited from Gandajaya[3].

Table 1. Parameter values

Parameter	Case 1		Case 2	
p_i	0.3	0.7	0.7	
p_{lt}	0.3	0.7	0.3	0.7
p_r	0.3	0.7	0.3	0.7
LIT^0	3	10	3	
LNR^0	5	15	5	

These two cases are used in order to observe how significant the influence of parameter values to the solution. There are 5 parameters which will be tested using the first case. They are p_i, p_r, p_{lt}, LNR^0, and LIT^0, while there are only two parameters for the second case which are p_{lt} and p_r.Not only identifying the significance of parameter values, we also compare the solution from viral systems to two other heuristics algorithms which are Genetic Algorithm with Multisteps Crossover (GA-MSX) [5] and Artificial Immune System (AIS) [6] in order to measure how good this viral systems algorithms among two others. This comparation is done using the solution from the second case.

Results and Discussion

As mentioned above, there are two application processes which will be done. The first one is significance testing and the second one is performance comparation. These two processes are explained as below.

Significance Testing

There are two cases for significance test. Each case has its own parameter values to test. Table 1 shows parameter values which will be used. The value of ITER and POB are set to be 100 for both cases.

Using full factorial experimental design, we have 2^5 combination of parameter values that have to be tested for the first case. We run the program ten times for each combination of parameter value. Finally, we have 320 data that can be used for this significance test. The summary of significance test result using analysis of variance concept can be shown as follow.

1st case result

Source	DF	AdjSSAdj MS	P
LIT0	1	0.0428 0.0428	0.819
LNR0	1	0.2258 0.2258	0.599
Source	**DF**	**AdjSSAdj MS**	**P**
plt	1	0.0070 0.0070	0.926
pr	1	5.3820 5.3820	0.011
pi	1	0.5363 0.5363	0.418
LIT0*LNR0	1	0.0428 0.0428	0.819
LIT0*plt	1	1.1163 1.1163	0.243
LIT0*pr	1	2.4675 2.4675	0.083
LIT0*pi	10.2475	0.2475 0.582	
LNR0*plt	1	2.6463 2.6463	0.073
LNR0*pr	1	0.1665 0.1665	0.651
LNR0*pi	1	0.1163 0.1163	0.706
plt*pr	1	1.9375 1.9375	0.124
plt*pi	1	0.0003 0.0003	0.985
pr*pi	1	0.2588 0.2588	0.573
LIT0*LNR0*plt	1	0.5040 0.5040	0.432
LIT0*LNR0*pr	1	0.1015 0.1015	0.724
LIT0*LNR0*pi	1	0.5200 0.5200	0.425
LIT0*plt*pr	1	0.1088 0.1088	0.715
LIT0*plt*pi	1	0.5528 0.5528	0.411
LIT0*pr*pi	1	0.0475 0.0475	0.809
LNR0*plt*pr	1	0.2153 0.2153	0.608
LNR0*plt*pi	1	0.1950 0.1950	0.625
LNR0*pr*pi	1	2.1288 2.1288	0.107
plt*pr*pi	1	0.0750 0.0750	0.762
LIT0*LNR0*plt*pr	1	2.1615 2.1615	0.104
LIT0*LNR0*plt*pi	1	0.8715 0.8715	0.302
LIT0*LNR0*pr*pi	1	0.4575 0.4575	0.454
LIT0*plt*pr*pi	1	1.4445 1.4445	0.184
LNR0*plt*pr*pi	1	0.8508 0.8508	0.308
LIT0*LNR0*plt*pr*pi	1	0.0878 0.0878	0.743
Error	288	234.5930 0.8146	
Total	319		

2nd case result

Source	DF	SS	MS	F	P
pi	1	6.981	6.98060	1.49	0.230
plt	1	6.061	6.06062	1.30	0.262
Interaction	1	1.180	1.17992	0.25	0.618
Error	36	168.221	4.67281		
Total	39	182.442			

The significance of parameter can be determined based on p-value of the test. P-value is stored in column 'P' in the result summary. Using 5% significance level, we conclude that only p_r that gives significant impact to solutions of the first scheduling problem. This result is justified with main effect plot in Figure 6. The steepest graph is shown by p_r. Further, we can estimate the level of $p_r$which give a better result in minimizing mean tardiness. According to Figure 6, value of p_r equal to 0.7 gives better result. It gives lower mean tardiness than 0.3 level. It is logically acceptable because the bigger value of p_r is, the faster each cell will collapse and explore other solutions in the solution space. It will collapse faster because there will be a higher probability to replicate the virus and reach the maximum level of lytic replication.

Move to the second case, we have 4 combination parameter values to test. The same as the first case, we run the program ten times for each combination, so that we get 40 data for significance test. Using full factorial ANOVA we get the result as follow.

From the 2nd case result, we can conclude that there are no significant effects from both parameters tested.

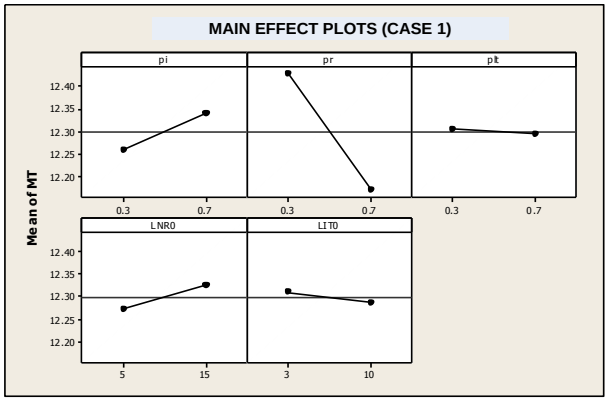

Figure 6. Main effect plots of mean tardiness

Performance Comparation

In this research, we also compare the performance of viral systems to Genetic Algorithms with Multisteps Crossover and Artificial Immune System in term of mean tardiness. We use the solution from the second case for this comparation process. Table 2 shows the data that we have from all three algorithms.

Using t-test with 5% significant level, we conclude that viral systems has the lowest performance (highest mean tardiness) among two others. This low perfomance may be because ineffective mutation method used. This mutation is considered to have a small exploration capability, while we know that mutation should be able to explore further solution space than exploitation process which is done by neighbouring solution development algorithm. This weakness is estimated to be worse when we are dealing with a very big problem which consists of hundreds of jobs and operations.We can predict it from the changes made by these two solution searching methods, especially the mutation method. This method may give no significant change of a solution/ genom so that the iteration process to get a new better solution is very slow for the second problem.

Conclusion

After completing all development and implementation steps, we can conclude that these new heuristics model can be implemented to scheduling problem. Heuristics method which is developed inthis paper may look complex while we know that it should be simpler than other methods such as optimization mathematics. We can say that it remains simple than optimization methods in term of mathematical logic. In optimization method, we have to develop several formulas such as mean tardiness calculation formula, constraints for machine assignment, etc., which obviously need more sophisticated thinking than algorithms in heuristics.That's why it is more preferable to use heuristics while solving a complex problem.

Table 2. Mean tardiness of three algorithms

	Algorithm		
Replication	GA-MSX	AIS	Viral
1	100.70	39.43	124.03
2	99.17	38.73	121.37
3	95.37	40.9	124.63
4	100.37	38.93	120.3
5	109.57	39.9	122.37
6			119.2
7			125.87
8			121.8
9			120.3
10			120.17

The development of algorithms should also consider the effectiveness of methods used (such as the exploitation and exploration method of solution space), because it will affect quality of the solutions.

Improvement of mutation method and robust heuristic method could be several next issues for developing this research. A more effective mutation method could support the whole algorithms toproduce better solutions. Another issue relates to robust heuristic method. Robustness relates to howsensitive the method will respond to changes of parameter values. The response means changes of solutions quality produced. The less sensitive the response is, the more robust the method will be

References

1. Baker, K.R., *Elements of Sequencing and Scheduling,*New York: John Wiley & Sons, 2001.
2. Cortés, P., García, J.M., Muñuzuri, J., and Onieva, L., Viral Systems: A New Bio-Inspired Optimisation Approach*, Computer & Operation Research*, vol. 35, 2007, pp 2840-2860.
3. Gandajaya, L.,*Penerapan Artificial Immune System dalam Penjadwalan Job Shop Statis pada Mesin Paralel Identik untuk Mengurangi Mean Tardiness.* Bandung: SkripsiJurusan Teknik Industri Universitas Katolik Parahyangan, 2010.
4. Gen, M. and Cheng, R.,*Genetic Algorithms & Engineering Design,* Canada: John Wiley & Sons, Inc., 1997.
5. Kolahan, F. and Kayvanfar, V., A Heuristic Algorithm Approach for Scheduling of Multi-criteria Unrelated Parallel Machines, *World Academy of Science, Engineering and Technology 59*, 2009.
6. Librata, H.S., *Penerapan Algoritma Genetika dengan Multi-Step Crossover dalam Permasalahan Penjadwalan Job Shop n Job m Mesin untuk Meminimasi Mean Tardiness.* Bandung: Skripsi Jurusan Teknik Industri Universitas Katolik Parahyangan, 2010.
7. Reeves, C.R., *Modern Heuristic Techniques for Combinatorial Problems,* United Kingdom: McGraw-Hill, 1995.