A Hybrid Metaheuristic Model For Job Shop Rescheduling Problem

ANIZA MOHAMED DIN, KU RUHANA KU MAHAMUD, YUHANIS YUSOF,
AND MASSUDI MAHMUDDIN
School of Computing, UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 Sintok, Kedah
MALAYSIA
{anizamd; ruhana; yuhanis; ady}@uum.edu.my

Abstract: - This paper discusses on developing a hybrid metaheuristic model to tackle the problem of changing environment in the job shop scheduling problem. The main idea is to use the model to develop building blocks of partial schedules that can be used to provide backup solutions when disturbances occur during production. Each partial schedule is assigned a fitness value for the selection of final population of best partial schedules. The results of the experiments show an improvement from a previous work. Future work on this study is also discussed.

Key-Words: - artificial immune systems, genetic algorithm, simulated annealing, great deluge, job shop scheduling, metaheuristic

1 Introduction

Job shop scheduling problem is concerned with tackling the problem of assigning *n* jobs to *m* machines. Several local search techniques such as genetic algorithm, simulated annealing, ant colony system and tabu search have been used to address the problem. This study specifically focuses on tackling the problem of changes in job shop environments. The changes include unexpected arrival dates of jobs in a factory. When jobs arrive too early, it might lead to jobs being stored for long periods of time and if they arrive late, it could cause delays in processing other jobs. An efficient method of rescheduling is needed to manage the problem.

This study aims to generate a range of partial schedules that could be used to produce backup schedules to maintain smooth flow of manufacturing process. In this paper, genetic algorithm and artificial immune system techniques are hybridised to build these partial schedules, which are then combined with local search to see whether there is improvement to the results. Past, complete schedules are used to build this collection of partial schedules. The data stems from [6] where the number of jobs used is 15, assigned to five machines. These processes will be explained in the next sections. Finally, findings from the experiments will be discussed.

2 A Hybrid Metaheuristic Model

The solution model for this study is developed from the theory of artificial immune system (AIS), which are then evolved using a genetic algorithm (GA).

Artificial immune systems (AIS) are inspired by the study of immunology. The biological immune system protects the body against antigens and generates antibodies that can bind to a specific antigen. The biological antibody evolves to enable it to cope with new antigens in addition to the common antigens. In [18], de Castro and Timmis discussed the classification of systems as artificial immune system. The system developed must include a basic model of an immune component and has to be designed based on theoretical or experimental ideas from immunology.

Previous research on scheduling has shown that AIS and GA can be used to solve scheduling problems in a manufacturing environment. Different scheduling problems have been addressed including the job shop scheduling problem [3,19,4,12,5,2, 24,29], flexible job-shop scheduling [1], the hybrid flow shop scheduling problem [22] and the job shop rescheduling problem [5,6,7], which is the main focus of this study. Hart and Ross built a block of partial schedules to tackle the job shop rescheduling problem [6]. There are many definitions given to the antibody and the antigen for the problem. This study employs the definition given by Hart and Ross. The key definitions used are described below:

ISBN: 978-1-61804-148-7

- An **antigen** is defined as "*the sequence of jobs on a particular machine given a particular scenario*" [6], which represents a complete schedule for the problem. For the experiments in this study, the antigens are represented by a sequence of numbers of length 15.
- An **antibody** is defined as "*a short sequence of jobs that is common to more than one schedule*" [6], which is also known as partial schedules. The antibodies are represented by sequences of numbers of length 5, where the length of an antibody is less than the length of an antigen.
- An **antigen universe** is considered to be a collection of antigens to be matched with the antibodies. An antigen universe has to be prepared before we can build an antibody population.
- An **antibody population** is a collection of partial schedules constructed from gene libraries.

The study is divided into three phases. First, an antibody population is generated and then evolved using a hybrid AIS and GA. The final population generated from the first phase is then used as initial solution to simulated annealing, and great deluge algorithms. This is to investigate the improvement in the fitness of the antibodies developed as both local search techniques have been proven to produce good results in examination timetabling problem [27,9]. The third phase is where the partial schedules will be selected to be recombined with incomplete schedules. In this paper, we are mainly concerned with the first two phases.

2.1 Phase I: Generate Antibody Population

The antigen universe for this study is the same used by Hart and Ross [7], which is based on a benchmark problem by Morton and Pentico [28]. The number of jobs used in this problem is 15 and the jobs have to be assigned to five machines. Hart and Ross created ten test scenarios by mutating the arrival dates of the jobs to a random date between 0 – 300 with a probability of 0.2. The arrival dates must not be less than p_t days before the due date, where p_t is the processing time of the job. A genetic algorithm developed in [11] is used to generate five schedules for each of these test-scenarios. This resulted in five sets of ten schedules; one for each machine, and these schedules became the antigen universe for the study. This study uses the antigen universe generated from one of the machines with the assumption that all machines have a similar pattern of jobs.

An antibody population is generated from gene libraries [4,6,7,23]. The gene libraries in this study are constructed from all the antigens in the antigen universe. The antigens are divided into five libraries, each consisting of ten partial schedules of size 3, also known as components. An antibody for this study is constructed based on a modular design method [25,26,20,17] where the length of each antibody is 1/3 the length of each antigen.

As an example, assume a set of gene libraries, consisting of four libraries and each library contains three components. Three genes (jobs) are allocated in each component. Following the modular design method, there are several ways to combine the genes from the components to produce an antibody. For example, the first component from Library 1 can be combined with the second component from Library 2 to produce an antibody. Since the length of an antibody is 5 jobs, a possible combination of

$$P\binom{n_1}{r_1} \times P\binom{n_2}{r_2} = \frac{n_1!}{(n_1 - r_1)!} \times \frac{n_2!}{(n_2 - r_2)!}. \quad (1)$$

can be constructed from this example, where n_1 and n_2 represent the number of jobs in the components from the first and second library, respectively, and r_1 and r_2 represent the number of jobs to be selected from the components. Therefore, a possible combination of three jobs from the first component and two jobs from the second component can be produced, and vice versa. This process is repeated until all the components in Library 1 have been combined with all the components in Library 2, as well as all the other libraries. It is also important to ensure no recurring jobs exist in one antibody. Each antibody generated in the population is filtered and antibodies with recurring jobs are eliminated. The process continues until a population of antibodies is generated.

A genetic algorithm based on GENESIS [13] is used to evolve the antibody population. Order-based crossover operator is used as it can ensure no job duplication in an antibody for any relationship between two parent antibodies. During crossover, tournament selection is applied to select the best antibody to be included in the next generation. The fitness of the children produced is evaluated and the values are then compared with the fitness of the parents. If the children produced have lower fitness than the parents, they will be discarded, and the parents are selected for inclusion in the next generation. Only the best antibodies, i.e. antibodies

with the highest fitness, will be considered for the next generation. A mutation operator, which randomly mutates each gene with a probability of 0.2, is also applied in [6].

The fitness of each antibody in the antibody population is then calculated using a matching function. A sample of antigens is first selected from the antigen universe. Each antibody is then matched against each of the antigens selected by aligning an antigen string with an antibody string and calculating a match score.

Antigen	1	2	7	4	3	9	6	8	14	5	Match score
	4	3	9	5	12						0
		4	3	9	5	12					0
			4	3	9	5	12				0
				4	**3**	**9**	5	12			15
					4	3	9	5	12		0
						4	3	9	5	12	0

Fig. 1. The process of matching an antibody with an antigen by aligning the antibody at every possible alignment position

Based on the example in Figure 1, antibody string '4 3 9 5 12' is aligned at every possible alignment position with the antigen string '1 2 7 4 3 9 6 8 14 5', job by job in order to calculate a match score. A match score is calculated by summing up the scores from the job matches where a match of each position contributed a score of five. Therefore, based on the number of matches between both the antibody and the antigen, the match score for the example given above is 15, which is the best possible match found (highest match score) by this process. Since an antibody is matched with each of the antigens in the sample, for antibody matched against more than one antigen, a total match score for the antibody is calculated by summing up the highest match scores from its match with each antigen.

Hart and Ross [6] selected certain samples of antibodies from the antibody population to be matched with a sample of antigens and repeated the matching process for a certain number of iterations based on the number of antigens selected. In this study, all the antibodies in the population are matched with the antigens and the matching process is run only once.

2.2 Phase II: Simulated Annealing and Great Deluge Algorithms

Local search methods are used on the final population generated from the first phase to improve the fitness of the antibodies. Two local search techniques are combined respectively with the hybrid model developed.

The simulated annealing (SA) algorithm simulates the process of annealing and a cooling schedule is employed to decrease the temperature and control the acceptance of a worse state [8,21]. Using this technique, the initial temperature, T_0 is set to 5000 and the final temperature, T_f to 0.05. The temperature is decreased by α, where α is defined as 0.98 which is found to be an effective value in the literature [14,15,16].

While the current temperature is greater than the final temperature, new antibodies, Ab_{new} are generated. This is done by applying two different operators, respectively in two different experiments; changing one job in *Ab* or swapping two jobs in *Ab*, where *Ab* represents the antibodies in the antibody population. The same matching function is used to calculate the fitness of each antibody. The new antibody will be kept if the fitness of the new antibody is better than the fitness of the current best antibody in the antibody population. Otherwise, it is accepted with a probability of $e^{-\delta T}$.

The Great Deluge (GD) algorithm is introduced by Dueck in [10]. This algorithm has a different acceptance process for worse solutions. The control parameter is called a level or boundary, which is used to determine or control the search. A worse solution is still acceptable as long as it is within the boundary, which at the beginning is set to the fitness of the initial solution. The boundary is then decreased by a fixed decay rate, β, at every iteration of the search.

The number of iterations, *iter* is set to 120, which is the possible number of new antibodies generated by an antibody. The estimated quality of the final solution, *f(EQ)*, which is the maximum fitness value for an antibody is also determined, depending on the number of antigens selected in the matching function. If the number of antigens selected is one, the maximum fitness value for *f(EQ)* is 25. This estimated quality represents the final estimated fitness value of an antibody. The boundary to the fitness of each antibody is decreased by a decreasing rate, β [3] and β is defined as below:

$$\beta = (f(Ab) - f(EQ)) / iter . \quad (2)$$

While the number of iterations does not exceed *iter*, new antibodies are generated by using the same two operators used in the simulated annealing algorithm, respectively in two experiments. The same matching function is used to calculate the fitness of each new antibody generated, *f(Ab)*. A new antibody which is worse than the old one will only be accepted if its fitness is less than the boundary. This loop will also stop if there is no more improvement in a fixed number of iterations.

3 Findings

Using a base problem jb11, taken from Morton and Pentico [28], ten test scenarios have been generated [7]. The schedules generated from the problem became the antigen universe for this study.

The antigen universe generates three types of antibody populations: 1) Type A - Population with antibody duplication (similar antibodies can exist in one population), 2) Type B – Population with no antibody duplication, and 3) Type C – Population with antibody duplication when the antibodies are constructed from different source libraries. These three types of antibody populations are generated as a test to see whether having a large number of similar antibodies in one population would affect the coverage of the antigen universe by the antibody population.

In the first phase, an initial population of size 100 was selected randomly from each type of antibody population. These populations were evolved using a genetic algorithm for 250 generations, with a crossover rate 0.7. Two mutation rates are used in the experiments. A mutation rate of 0.2, which is the same parameter used in [6] is applied so that it is easier for results comparison purposes. Then, a mutation rate of 0.001 is used as it gives a steady growth of the fitness of the antibodies in the antibody population. The antibodies evolved here were the antibodies with the highest fitness value in each generation. As the antibodies evolve, the average fitness of the antibodies also increases. At the end of the generation, the final population should consist of a collection of general and specific antibodies, which could either match many antigens or only one specific antigen.

Tables 1 and 2 show the average number of antigens that cannot be matched by any antibody for a match threshold ranging from 2 to 5. A match threshold, tm, is a guideline to determine whether an antibody and antigen are matched. The number of jobs to bind or match must be greater or equal to the threshold value of t_m [6]. This experiment tests the coverage of the antigen universe by the antibody population. Table 1 shows the results of the experiment by Hart and Ross. Table 2 shows findings from this study performed on final populations generated from the antibody population Type A, Type B and Type C, respectively with a mutation rate of 0.2.

Table 1. Average number of antigens (out of a possible 10) not matched by any antibody as generated by Hart and Ross[6]

Match Thres-hold	Ag = 1			Ag = 4			Ag = 8		
	Ab			Ab			Ab		
	5	10	30	5	10	30	5	10	30
2	0.9	0.0	0.0	2.2	0.9	0.0	3.5	2.5	0.9
3	5.3	2.6	1.6	5.4	3.2	2.0	5.5	4.7	4.1
4	8.7	7.1	5.2	7.8	7.3	6.3	8.6	8.1	8.2
5	9.7	9.5	8.8	9.5	9.5	8.7	9.7	9.6	9.5

Table 2. Average number of antigens (out of a possible 10) not matched by any antibody (modified algorithm for AIS)

Match Thres-hold	Ab = 100								
	Type A			Type B			Type C		
	Ag			Ag			Ag		
	1	4	8	1	4	8	1	4	8
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.4	0.0	0.0	0.9	0.1	0.0	0.8	0.1	0.0
4	6.5	3.6	1.3	6.2	3.4	1.4	6.6	3.2	1.3
5	8.5	6.3	4.7	8.3	6.6	5.3	8.2	7.1	5.8

In Table 1, the results from Hart and Ross created a trend where the average number of antigens not matched by any antibody decreases as the size of the antibody samples, *s* increases from 5 to 30. The analysis in Table 2 is in line with the trend where the average number of unmatched antigens decreases when the whole population is matched against the antigens. However, in this study, as compared to Hart and Ross, it is found that when the number of antigens increases, the average number of antigens that cannot be matched by any antibody decreases. While the result by Hart and Ross could be interpreted as evidence that more specific antibodies have been produced, it is believed that this study is able increase the fitness of the antibodies when more antigens are exposed to the antibodies. This results in more antigens getting matched or recognized.

The results depicted in Table 3 are the average number of antigens not matched by any antibody for both hybrid models with SA and GD, respectively with a mutation rate of 0.001 on antibody

ISBN: 978-1-61804-148-7

population Type A. The percentage of the fitness improvement on antibodies generated is also shown in the table.

Table 3. Average number of antigens (out of a possible 10) not matched by any antibody in population Type A

Match Thres-hold	Ab = 100								
	Hybrid GA+AIS			**Hybrid (GA+AIS) + SA**			**Hybrid (GA+AIS) + GD**		
	Ag			**Ag**			**Ag**		
	1	**4**	**8**	**1**	**4**	**8**	**1**	**4**	**8**
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.6	0.1	0.0	1.5	0.3	0.0	1.4	1.4	0.0
4	6.8	3.0	1.0	**6.6**	4.7	**0.6**	**6.5**	**6.5**	1.4
5	7.9	6.0	4.4	8.3	**5.3**	**3.5**	8.2	8.2	4.8
Fitness Diff. **(%)**				**28.5**	**11.7**	**4.8**	**28.8**	**10.7**	**4.7**

Table 3 shows that by combining the hybrid model developed with local search technique, the fitness of the antibodies can be improved. However, it does not improve the coverage of the antigen universe compared the previous experiment. This is probably due to the large number of general antibodies produced that can be matched with most of the antigens. Both models produced more specific antibodies and therefore could not cover most of the antigens.

The fitness of the antibodies in the population, however, does improve, as depicted in the last row in Table 3. The fitness of the antibody populations generated using the hybrid models increases more than 28% as compared to the previous experiment. However, the percentage drops gradually as the number of antigens selected increases.

4 Conclusion

A hybrid metaheuristic model consisting of genetic algorithm and artificial immune system, combined with simulated annealing, and great deluge algorithms respectively has been developed to tackle the problem of job shop rescheduling. The findings represent an improvement from those in the previous works. While the results did not show improvement in terms of the coverage of the antigen universe, they did improve the fitness of the antibodies produced in the population. This is important in order to find good search algorithm that could produce a range of good partial schedules to be used as replacement for certain jobs in the actual schedule when changes occur in the arrival dates of the jobs. Further work for this study is to investigate whether the model developed can be applied in examination timetabling problem.

Acknowledgment

This research is supported by the Fundamental Research Grant Scheme (FRGS). We thank the Ministry of Higher Education Malaysia and Universiti Utara Malaysia for the financial support and the administration of the study.

References:

[1] A., Bagheri, M., Zandieh, I., Mahdavi, and M., Yazdani, An Artificial Immune Algorithm for Flexible Job-Shop Scheduling Problem, *Future Generation Computer Systems*, vol. 26, no. 4, 2010, pp 533 – 541.

[2] A., Mohsen, and S., Hedieh, A Novel Artificial Immune Algorithm for Solving The Job Shop Scheduling Problem, *International Journal of Computer Applications* (0975 – 888), vol. 48, no. 14, 2012, pp 46 – 53.

[3] C., Bin, W., Shilong, and H., Haibo, Hybrid Artificial Immune System for Job Shop Scheduling Problem, *World Academy of Science, Engineering and Technology*, 59, 2011, pp 81 – 86.

[4] C.A., Coello, D.C. Rivera, and N.C., Cortes, Use of an Artificial Immune System for Job Shop Scheduling, *Proceedings of ICARIS 2003*, LNCS 2787, Timmis et al (Eds.), Springer-Verlag, 2003, pp 1 – 10.

[5] E., Hart, P., Ross, and J., Nelson, Producing Robust Schedules Via An Artificial Immune System, *Proceeding of the ICEC '98*, IEEE Press, 1998, pp 464-469.

[6] E., Hart, and P., Ross, An Immune System Approach to Scheduling in Changing Environments, *Proceeding of the GECCO 1999*, W.Banzhaf, J.Daida, A.E.Eiben, M.H.Garzon, V.Honavar, M.Jakiela and R.E.Smith (eds), Morgan Kaufmann, 1999, pp 1559-1565.

[7] E., Hart, and P., Ross, The Evolution and Analysis of a Potential Antibody Library for Job-Shop Scheduling, *New Ideas in Optimisation*. D. Corne, M.Dorigo & F. Glover (eds), McGraw-Hill, London, 1999, pp 185-202.

[8] E., Aarts, J., Korst, and W., Michiels, Chapter 7: Simulated Annealing, *Search Methodologies: Introductory Tutorials in Optomization and Decision Support*

Techniques, (Burke, E.K. and Kendall, G (eds)), Springer, 2005.

[9] E.K., Burke, Y., Bykov, J. P., Newall, S., Petrovic, A Time-Predefined Local Search Approach to Exam Timetabling Problem, *IIE Transactions*, 36(3), 2004, pp 509-528.

[10] G., Dueck, New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel, *Journal of Computational Physics*, 1993, pp 104, 86-92.

[11] H.-L., Fang, P., Ross, and D., Corne, A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling and Open-Shop Scheduling Problems, *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forrest (ed.). San Mateo, Morgan Kaufmann, 1993, pp 375-382.

[12] H.W., Ge, L., Sun, and Y.C., Liang, Solving Job-Shop Scheduling Problems by a Novel Artificial Immune Systems, *Proceedings of AI2005*, LNAI 3809, Zhang, S. and Jarvis, R. (Eds.), Springer-Verlag, 2005, pp 839 – 842.

[13] J., Grefenstette, Genesis: A System For Using Genetic Search Procedures, *Proceedings of a Conference on Intelligent Systems and Machines*, 1984, pp 161-165.

[14] J., Thompson, and K.A., Dowsland, General Cooling Schedules for a Simulated Annealing Based Timetabling Systems, *Practice and Theory of Automated Timetabling*, Lecture Notes in Computer Science 1153, (Eds. E Burke and P Ross), Springer-Verlag, 1996, pp 345 – 363.

[15] K.A., Dowsland, Chapter 2: Simulated Annealing, *In Modern Heuristic Techniques For Combinatorial Problems*, Reeves, C.R. (ed), McGraw Hill, 1995.

[16] K.A., Dowsland, Off-the-Peg or Made-to-Measure? Timetabling and Scheduling with SA and TS, *Practice and Theory of Automated Timetabling II 1997*, Lecture Notes in Computer Science 1408, (Eds. E. Burke and M. Carter), Springer-Verlag, 1998, pp 37 – 52.

[17] L., Sompayrac. *How the Immune System Works*, 2nd Edition, Blackwell Publishing, 2003.

[18] L. N., de Castro. and J, Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer-Verlag, 2002

[19] M., Chandrasekaran, P., Asokan, S., Kumanan, T., Balamurugan, and S., Nickolas, Solving Job Shop Scheduling Problems Using Artificial Immune System, *The International Journal of Advanced Manufacturing Technology*, Vol. 31, No. 5-6, Springer London, 2006, pp 580 – 593.

[20] M., Oprea, and F., Forrest, Simulated Evolution of Antibody Gene Libraries Under Pathogen Selection, *Proceedings of 1998 IEEE International Conference on Systems, Man and Cybernetics*, 1998.

[21] N., Metropolis, A.W., Rosenbluth, M.N., Rosenbluth, A.H. Teller, and E., Teller, Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics*, 21(6), 1953, pp 1087-1092.

[22] O., Engin, and A., Doyen, A New Approach to Solve Hybrid Flow Shop Scheduling Problems by Artificial Immune System, *Future Generation Computer Systems*, 20, Elsevier, 2004, pp 1083 – 1095.

[23] P., Spellward, and T., Kovacs, On the Contribution of Gene Libraries to Artificial Immune Systems, *Proceedings of 2005 Conference on Genetic and Evolutionary Computation (GECCO '05)*, ACM Press, 2005, pp 313 – 319.

[24] Q., Ren, and W., Yuping, A New Hybrid Genetic Algorithm for Job Shop Scheduling Problem, *Computers a& Operations Research*, vol.39, no.10, 2012, pp. 2291 – 2299.

[25] R.A., Goldsby, T.J., Kindt, and B.A., Osbourne, *Kuby Immunology*, 4th Edition, New York: W.H. Freeman, 2000.

[26] R.R., Hightower, S., Forrest, and A.S., Perelson, The Evolution of Emergent Organization in Immune System Gene Libraries, *Proceedings of Sixth Annual Conference on Genetic Algorithms*, Eshelman, L. (ed.), Morgan Kaufmann, 1995, pp 344 – 350.

[27] S., Abdullah, and E.K., Burke, A Multi-start Large Neighbourhood Search Approach with Local Search Methods for Examination Timetabling, *The Proceedings of International Conference on Automated Planning and Scheduling (ICAPS 2006)*, Editors: Derek Long, Stephen F. Smith, Daniel Borrajo and Lee McCluskey, Cumbria, UK, 2006, pp 334 – 337.

[28] T.E., Morton, and D.W., Pentico, *Heuristic Scheduling Systems*, John Wiley, 1993.

[29] Z., Yunli, and Q., Xueni, A Hybrid AIS-Based Algorithm For Solving Job Shop Scheduling Problem, *Proceedings of 2012 International Conference on Communication Systems and Network Technologies*, IEEE Computer Society, 2012, pp 498 – 502.