Job Shop Rescheduling Using A Hybrid Artificial Immune Systems and Genetic Algorithm Model

Aniza Mohamed Din
School of Computing, UUM College of Arts and Sciences
Universiti Utara Malaysia, Malaysia
anizamd@uum.edu.my

Ku Ruhana Ku-Mahamud
School of Computing, UUM College of Arts and Sciences
Universiti Utara Malaysia, Malaysia
ruhana@uum.edu.my

Yuhanis Yusof
School of Computing, UUM College of Arts and Sciences
Universiti Utara Malaysia, Malaysia
yuhanis@uum.edu.my

Massudi Mahmuddin
School of Computing, UUM College of Arts and Sciences
Universiti Utara Malaysia, Malaysia
ady@uum.edu.my

***Abstract*—This paper discusses on developing a hybrid model to tackle the problem of changing environment in the job shop scheduling problem. The main idea is to develop building blocks of partial schedules using the model developed that can be used to provide backup solutions when disturbances occur during production. This model hybridizes genetic algorithm (GA) with artificial immune systems (AIS) techniques to generate these partial schedules. Each partial schedule, also known as antibody, is assigned a fitness value for the selection of final population of best partial schedules. The results of the analysis are compared with previous research. Future works on this study are also discussed.**

Keywords-component; artificial immune systems; genetic algorithm; job shop scheduling

I. INTRODUCTION

Job shop scheduling problems are well studied problems which 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. Fang [7], and Jensen and Hansen [15] employed a genetic algorithm to produce robust schedules for scheduling problems, where Fang also addressed job shop rescheduling problem. This study is specifically trying to tackle the problem of changing 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 paper is motivated by the aim of generating a range of partial schedules that could be used to produce backup schedules in the event of changes in a job shop environment to keep a smooth flow of manufacturing process with less interruption. In this paper, genetic algorithm and artificial immune system techniques are used to build these partial schedules. Past, complete schedules (later known as the antigen universe) are used to build this collection of partial schedules. This data stem from [13] where the number of jobs used is 15 assigned to five machines. These processes will be explained in Section II. Section III will discuss on the findings from the experiments.

II. A HYBRID MODEL

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

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. A biological antibody evolves to enable it to adapt with new antigens in addition to the ones that are already known. In [5], de Castro and Timmis discussed the classification of systems as artificial immune system. The system developed has to incorporate a basic model of an immune component and has to be designed by drawing upon theoretical or experimental ideas from immunology.

Previous works on scheduling has shown that AIS and GA can be used in a manufacturing environment. Different scheduling problems have been addressed including the job shop scheduling problem [2,3,4,8,11,16,19], flexible job-shop scheduling [1], the hybrid flow shop scheduling problem [6] and the job shop rescheduling problem [11,12,13], which is the main concern of this study. In [12], Hart and Ross built a block of partial schedules to tackle the job shop rescheduling problem. There are many definitions given to the antibody and the antigen for the problem. This study employs the definition given by Hart and Ross in [12]. The key definitions used are described below:

- An **antigen** is defined as "*the sequence of jobs on a particular machine given a particular scenario*" [12], 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*" [12], 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.
- **Gene libraries** consist of genotypes [14,18]. The gene libraries in this study are constructed from all the antigens in the antigen universe.
- A **final population** consists of a collection of best antibodies.
- **Fitness** represents the value assigned to each antibody in the antibody population to evaluate the coverage of an antibody over the antigens. It is used to determine which antibody can be considered as one of the best antibodies to be selected for the final population. Fitness can be calculated by matching the antibodies with the antigens. The higher the fitness, the better an antibody will be.

The study is divided into three phases. In the first phase, an AIS model is used to generate the antibody population, with l = 5 where l is the length of an antibody before we can evaluate each antibody to be selected into the collection of best antibodies (partial schedules). A genetic algorithm is then used to evolve the antibodies in the second phase. The idea is that only the antibodies with the highest fitness will be kept in the final population. The goal is to investigate if the fitness of the antibodies developed in the final population can be improved. When a final population is developed, in the third phase, three methods are applied to select the partial schedules, which are simple recombination, somatic recombination and single job addition [12]. The partial schedules developed are then recombined with incomplete schedules. This paper mainly focuses on the first two phases. It is also important to note here that the study applied the genetic algorithm as used in [12,13] and the algorithm is modified with the aim of improving the results.

A. Phase 1: Generate Antibody Population

Before antibody populations can be generated, an antigen universe must be created. The antigen universe for this study is the same used by Hart and Ross [13], which is based on a benchmark problem by Morton and Pentico [17]. 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 [7] 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.

The next step in this phase is to generate an antibody population from gene libraries [4,12,13,21]. 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 [9,14,18,20] 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

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

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. From the example, we can see a combination of three jobs from the first component and two jobs from the second component. We can get other combinations from these two components using (1) above to generate an antibody population. 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 checked and antibodies with recurring jobs are eliminated. The process continues until a population of antibodies is generated. By doing this, a level of antibody diversity can be developed.

B. Phase 2: Evolving the Population

A genetic algorithm based on GENESIS [10] is implemented and is used to evolve the antibody population. The crossover operator used is order-based crossover operator, 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 [12].

A matching function is used as the evaluation function within the genetic algorithm to calculate the fitness of each antibody in the antibody population. 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.

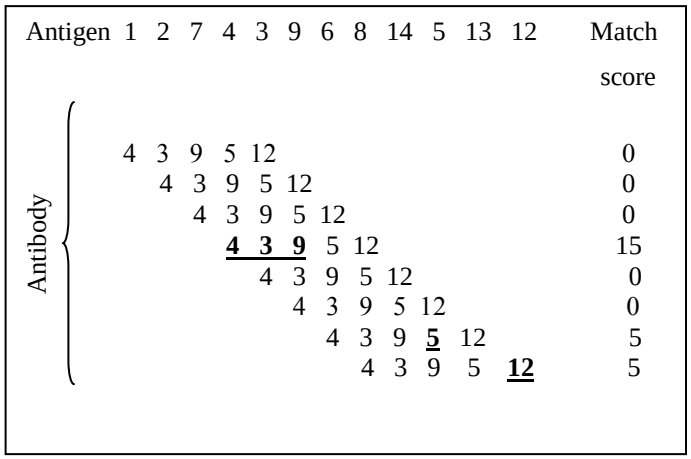

Figure 1. The process of matching an antibody with an antigen by aligning the antibody at every possible alignment position

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 13, 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 [12] 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. It is also important to note that for the preliminary experiments, any wildcard genes are not included in any antibody. This way the exact fitness of the antibodies can be seen when they are matched with the antigens. In [12], the authors allow a wild card match between the antibody and the antigen. A wild card is used as a substitute to any job.

III. EXPERIMENTAL RESULTS

As described in the previous section, Hart and Ross created ten test scenarios from a base problem, *jb11*, taken from Morton and Pentico [13,17] and the schedules generated from the problem became the antigen universe for this study. The antigen universe generates three types of antibody populations:

1) Population with antibody duplication (there are several similar antibodies in one population) – Type A (4514 antibodies)

2) Population with no antibody duplication regardless of the source gene libraries (no similar antibodies in one population) – Type B (2416 antibodies)

3) Population with antibody duplication (only when the antibodies are constructed from different source libraries) – Type C (2839 antibodies)

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 as used in [12]. We used two mutation rates in the experiments. A mutation rate of 0.2 is used as it is the same parameter used in [12] and therefore 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.

Table 1. Average number of antigens (out of a possible 10) not matched by any antibody as generated by Hart and Ross[12]

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

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, t_m, 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 [12]. 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 [12]. Table 2

shows findings from this study performed on final populations generated from the antibody population Type A, Type B and Type C, respectively (Phase I) with a mutation rate of 0.2.

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 in [12] 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. Therefore with this study, the partial schedules produced can be used as replacement to an actual schedule when disturbances occur.

IV. CONCLUSION

A hybrid model of AIS and GA has been developed to tackle the problem of job shop rescheduling. The findings represent an improvement upon those in the previous works. While the results did not yield 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, as we need 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 we have changes in the arrival dates of the jobs.

Further work for this study is to investigate the possibilities of hybridizing the current model developed with local search algorithms to improve the performance of the model.

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. The authors would also like to thank Professor Edmund Burke, Professor Uwe Aickelin and Dr. Rong Qu from the University of Stirling and University of Nottingham, UK for their technical support.

REFERENCES

[1] Bagheri, A., Zandieh, M., Mahdavi, I., and Yazdani, M., "An artificial immune algorithm for flexible job-shop scheduling problem," Future Generation Computer Systems, vol. 26, no. 4, pp 533 – 541, 2010.

[2] Bin, C., Shilong, W., and Haibo, H. "Hybrid artificial immune system for job shopscheduling problem," World Academy of Science, Engineering and Technology, 59, pp 81 – 86, 2011.

[3] Chandrasekaran, M., Asokan, P., Kumanan, S., Balamurugan, T. and Nickolas, S., "Solving job shop scheduling problems using artificial immune system," The International Journal of Advanced Manufacturing Technology, vol. 31, no. 5-6, Springer London, 580 – 593, 2006.

[4] Coello, C.A., Rivera, D.C. and Cortes, N.C., "Use of an artificial immune system for job shop scheduling," in Proceedings of ICARIS 2003, LNCS 2787, Timmis et al (Eds.), Springer-Verlag, 1 – 10, 2003.

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

[6] Engin, O. and Doyen, A., "A new approach to solve hybrid flow shop scheduling problems by artificial immune system," Future Generation Computer Systems, 20, Elsevier, 1083 – 1095, 2004.

[7] Fang, H.-L., Ross, P., and Corne, D., "A promising genetic algorithm approach to job-shop scheduling, rescheduling and open-Shop scheduling problems," in Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (ed.). San Mateo, Morgan Kaufmann, 375-382, 1993.

[8] Ge, H.W., Sun, L. and Liang, Y.C., "Solving job-shop scheduling problems by a novel artificial immune systems," in Proceedings of AI2005, LNAI 3809, Zhang, S. and Jarvis, R. (Eds.), 839 – 842, Springer-Verlag, 2005.

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

[10] Grefenstette, J., "Genesis: A system for using genetic search procedures," in Proceedings of a Conference on Intelligent Systems and Machines, 161-165, 1984.

[11] Hart, E., Ross, P., and Nelson, J., "Producing robust schedules via an artificial immune system," in Proceeding of the ICEC '98, IEEE Press, 464-469, 1998.

[12] Hart, E., and Ross, P., "An immune system approach to scheduling in changing environments," in 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, 1559-1565, 1999.

[13] Hart, E., and Ross, P., "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, 185-202, 1999.

[14] Hightower, R. R., Forrest, S., and Perelson, A. S., "The evolution of emergent organization in immune system gene libraries," in Proceedings of Sixth Annual Conference on Genetic Algorithms, Eshelman, L. (ed.), Morgan Kaufmann, 344 –350, 1995.

[15] Jensen, M. T., and Hansen, T. K., "Robust solutions to job shop problems," in the Proceedings of the 1999 Congress on Evolutionary Computing (CEC99), 1138 – 1144, 1999.

[16] Mohsen, A. and Hedieh, S., "A novel artificial immune algorithm for solving the job shop scheduling problem," International Journal of Computer Applications (0975 – 888), vol. 48, no. 14, pp 46 – 53, 2012.

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

[18] Oprea, M., and Forrest, F., "Simulated evolution of antibody gene libraries under pathogen selection," in Proceedings of 1998 IEEE International Conference on Systems, Man and Cybernetics, 1998.

[19] Ren, Q., and Yuping, W., "A new hybrid genetic algorithm for job shop scheduling problem," Computers a& Operations Research, vol.39, no.10, pp. 2291 – 2299, 2012.

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

[21] Spellward, P. and Kovacs, T., "On the contribution of gene libraries to artificial immune systems," in Proceedings of 2005 Conference on Genetic and Evolutionary Computation (GECCO '05), ACM Press, 313 – 319, 2005.

[22] Yunli, Z., and Xueni, Q., "A hybrid AIS-based algorithm for solving job shop scheduling problem," in proceedings of 2012 International Conference on Communication Systems and Network Technologies, IEEE Computer Society, pp 498 – 502, 2012.