Minimizing the makespan for the problem of reactive production scheduling in a FMS with AGVs using a new structure of chromosome in a hybrid GA with TS

Carlos C.M.Tuma, Orides Morandin Jr, Vinicius F. Caridá
Department of Computer Science
Federal University of São Carlos
São Carlos, Brazil
(carlos_tuma, orides, vinicius_carida)@dc.ufscar.br

Abstract

Scheduling of Simultaneous production of machines and AGVs in Flexible Manufacturing Systems involves modeling and searching methodologies in a wide solution space. A criterion often used in the analysis of results for this type of problem is makespan. Many search methods are being used in an attempt to minimize the makespan. This paper has two research objectives. The first consists in verifying the hypothesis that hybridization of a global search (Genetic Algorithm (GA)) with a local search (Tabu Search (TS)) can achieve better results in the problem of reactive scheduling production. The second consists in investigation of a new structure of the GA chromosome for the algorithm has higher correlation with the variable of minimization of the problem, the makespan. This proposal involves modeling tasks in Timed Petri nets during the fitness evaluation, considering the input buffers of machines, AGVs, and control flags for these buffers. The proposal was tested in one scenario of FMS and validated by comparing its results.

1. Introduction

Currently, the manufacturing industries have undergone tremendous changes due to the increased use of process technologies, such as machine tools, computerized numerical control, industrial robots, handling systems and transport of materials [1].

This context indicates a change in focus that places the manufacture no longer as a simple support area, but as a member of the group of factors determining competitiveness. In this environment, the manufacture must be evaluated on its strategic dimensions, such as cost, quality, service and flexibility [2].

In Flexible Manufacturing Systems (FMS) a product usually has several routings and is manufactured in different machines, not sequential, geographically separated from each other, necessitating means of transport. This transport system usually consists of AGVs (Automated Guided Vehicle) [3].

The problem of production planning (PP) is the allocation of resources in time and in the correct sequence, so that the result is the completion of all tasks in the shortest time possible [2]. This problem is NP-hard, i.e., the computational cost grows exponentially with the size of the problem [4].

Among the possibilities to solve this problem there is the use of artificial intelligence techniques. Several approaches proposed for addressing the problem of scheduling indicate a high efficiency in the use of genetic algorithms (GA) [5] [6] [7] [8] [9]. Other approaches make use of Adaptive Genetic Algorithms (AGA) to avoid the problem of poor choice of parameters and premature convergence of a traditional genetic algorithm [10] [11] [12] [18].

This problem has been treated as isolated from the transportation problem, but has indicated the importance of the need to treat it as a problem of production scheduling simultaneously with the transport such as the approaches of [14] [15] [16] [1].

New approaches suggest using the genetic algorithm, which is a global search method in conjunction with a local search method [26] [27]. Hybridization of genetic algorithm with Tabu Search (TS) has shown promise in some studies [28] [29] [30] [31] [32].

To simplify the PP, it's common the simplification of scenarios in their factory modeling, ignoring or grouping elements such as setup time, input buffers, output buffers, AGVs, etc. So modeling usually does not include the input buffers of machines, AGVs or some kind of safeguard for deadlock. Another point to be consider is the time to get the solution. I.e. the PP problem will not be considered as planning task, but as a control one. This means that the problem of this work is, in fact, a reactive scheduling problem.

One of the motivations of this work is to bring the model problem closer to reality, by considering AGVs and input buffers in modeling, including also the use of control flags to avoid the deadlock.

Treatment interlocking within Timed Petri Nets (TPN), through the inclusion of AGVs and flag free buffers, should resolve the disputes of the AGV, thus avoiding deadlocks. The possibility to advance the carriage through the input buffer generates the expectation of greater performance in makespan.

Within this context, this paper proposes the use of the adaptive genetic algorithm with Tabu Search to problem of reactive production scheduling. Is also proposed a new form of structure of the genetic algorithm chromosome to adhere better to the problem and find better makespan results.

978-1-4799-0864-6/13/$31.00 ©2013 IEEE

2. Addressing the problem using genetic algorithm and timed petri net

The purpose of this study has been to develop a strategy for modeling production scheduling in flexible manufacturing systems using TPN considering input buffers, AGVs and free buffer flags, using:

- Genetic Algorithm, with parameters given in [18], with complete scheduling representation in one chromosome. More details in the next section;
- Modeling of Jobs and AGVs in TPN, using the input buffer and free buffer flag, example in [25] Figure 4;
- Allocation of AGVs by simple rules, which are: AGVs serving in the order of their numbering and transitions are met in the order of their creation in modeling.

The problem of scheduling production includes several products that can be produced through several different routes. What is sought is a series of products to be considered in producing such a way that corresponds to a desired performance goal. Several performance measures are used today, and in this paper it is used the minimum makespan objective. The results were compared with some articles with published results.

There are some assumptions in the proposed system, same used in comparisons, except the last, namely:

- Each machine operates only one product at a time and each product can only be operated by one machine at a time;
- An AGV carries only one product at a time;
- The times of production and transportation are deterministic and known in advance;
- The setup times are included in the time of production;
- A job starts transporting the raw material from the load sector to the input buffer of the first machine in your routing;
- A job ends up at the finished product transportation from the output buffer of the last machine to the unload sector;
- Machines and AGVs do not stop or are defective;
- 2 AGVs are considered;
- Every machine has an input buffer of size 1 and output buffer size large enough to support any claim.

The algorithm uses as fitness function the inverse of the number of turns required to go through the path in the tree cover for the PP provided.

3. Implementation

The system to validate the proposal will have six variants for comparison and verification of results.

The first and second implementation will be the genetic algorithm replicating the paper [1] and adaptive genetic algorithm replicating the paper [18]. The third and fourth implementation will be the genetic algorithm and adaptive genetic algorithm both with modified chromosome to prove the hypothesis that a chromosome with more adherences to the problem can provide better results. Details of how to work both methods are in section 3.1.

The fifth will be the implementation of the adaptive genetic algorithm of the paper [18] with the assistance of Tabu Search, to prove the hypothesis that a global search with local search provides better results for the problem. The sixth will be the implementation of the adaptive genetic algorithm with modified chromosome and with Tabu Search showing that even with a hybrid search the chromosome with more adherences to the problem gets better results. Some details of Tabu Search are in section 3.2.

3.1. Modified Chromosome

Are many papers in the literature that use genetic algorithm on the problem of production scheduling, as the work [18], using the configuration that each chromosome maps a possible solution, according to Figure 1, where it is seen that each pair of genes indicate a production of a product using a specific routing.

Note that in Figure 1 where P1 is the product 1, R11 is the routing 1for the production of P1, P2 is the product 2, R23 is the routing 3 for the production of P2, P3 is the product 3, R32 is the routing 2 for the production of P3.

P1	**R11**	**P2**	**R23**	**P3**	**R32**

Fig. 1. Chromosome mapping with one complete solution, adapted [18].

The new chromosome proposed in this paper is illustrated in Figure 2. This chromosome is divided into two parts, A and B. Part A indicates the routing for the production of product. The position vector indicates which product (illustrated in Position item), ie, the first three positions of the chromosome (Part A) show:

Position 1: Product 1 with routing 2;

Position 2: Product 2 with routing 1;

Position 3: Product 3 with routing 1.

The second part of the chromosome (part B, which goes from position 4 to position 12) indicates the sequence of operations that will be executed according to the scheduling, where:

Position 4: Product 1 in operation 1;

Position 5: Product 2 in operation 1;

…

Position 12: Product 3 in operation 3.

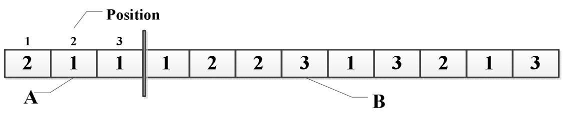

Fig. 2. Modified Chromosome.

The genetic parameters used in [18] are:

- Crossover: 80%, making the exchange of routes between 2 identical products on 2 chromosomes, as illustrated in Figure 3.
- Population of 20 individuals;
- Selection for new population: 4 by elitism and 16 per roulette;

For the new chromosome the parameters are the same used by the authors in [18], but the form of the crossover and mutation will be different.

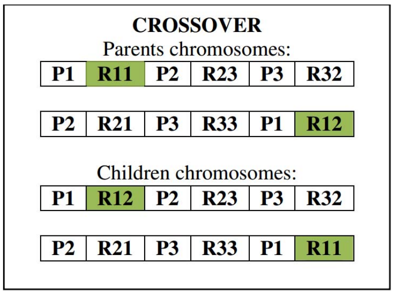

Fig. 3. Mechanism of Crossover, adapted [18].

In the new chromosome, the crossover happens only in the second half of part B, as illustrated in Figure 4. After the crossover the system checks if the number of operations consistent with the routing of Part A of chromosome. If there is any irregularity, the system makes an adjustment in the last position of the chromosome. In the second child chromosome in Figure 4, by crossing the last position would have the value 2 and 3 respectively, but the verification altered for 3 and 2 to maintain agreement with the routings.

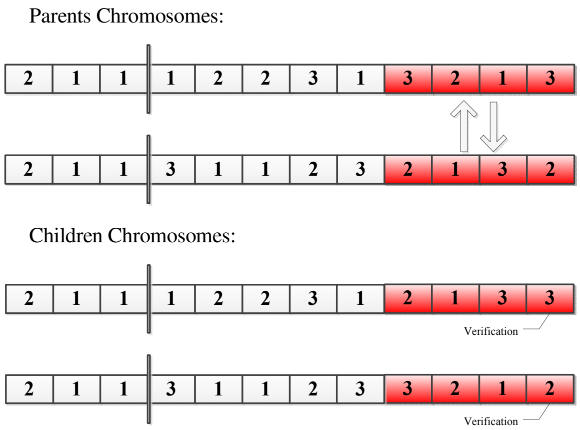

Fig. 4. Mechanism of Crossover of the Modified Chromosome.

Mutation: 5% (1 individual per generation), random exchange routings for a product by another valid routing, as illustrated in Figure 5.

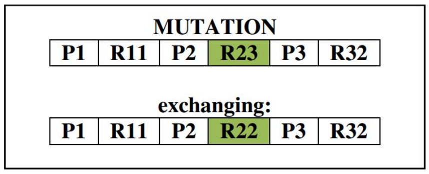

Fig. 5. Mechanism of mutation, adapted [18].

In the new chromosome, the mutation happens only in part A, as illustrated in Figure 6. After mutation the system checks if the new routing consistent with the number of operations on the part B of chromosome. If there is any irregularity, the system makes an adjustment in the last position of the chromosome.

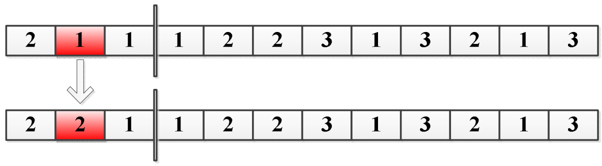

Fig. 6. Mechanism of Mutation of the Modified Chromosome.

The fitness of each chromosome used in genetic algorithm is obtained by the equation (1):

$$F(i) = \frac{1}{mkp_i} \quad (1)$$

Where: F(i) is the fitness function of chromosome i and mkpi is the makespan of the same chromosome.

Stopping criterion: Experiment - 50 generations (covering 0.0005% of the search space).

3.2. Tabu Search

Tabu search is a meta-heuristic method [33] [34] and many researches have used it to solve scheduling problems. One of its principal characteristics is a use of a memory of restrictions during a search, called Tabu memory.

In this work, these memories have a size of 5 restrictions and this restrictions comprehends the positions of swap operations in initial chromosome to create a new one. As operator in neighborhood it is used a swap operator, that exchanges 2 operations, randomly chosen, like shown in Fig. 7.

Stopping criterion: 50 iterations.

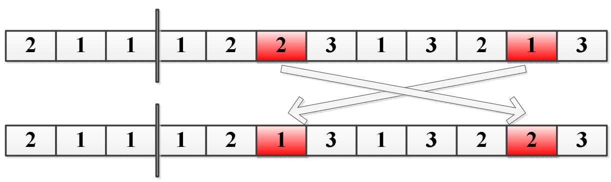

Fig. 7. Swap Operation.

4. Computer Experiments

Four batteries of tests were performed, each execution with 35 results. The experiment used a specific scenario, with 2 AGVs.

Best makespan and corresponding scheduling is presented as sequence of operations and as a Gantt chart. Best scheduling, as well as the sequence obtained from their performances are presented and a Gantt chart of the best scheduling in experiment is obtained from sequencing.

All results are in terms of turns required to complete all jobs (makespan). Table 4 presents all average of results obtained in experiments.

4.1. Experiments

Experiments consists of running 35 times of the proposed algorithm, within a scenario of 9 products, with 9 machines and two routings for each product. Tables Table 1, Table 2 and Table 3 present a detailed description of this scenario.

TABLE I. PRODUCTS AND ROUTES.

PRODUCTS	ROUTES	
P1	R11	M1 M2 M4 M5 M7 M9
	R12	M3 M4 M5 M6 M8 M9
P2	R21	M1 M2 M3 M4 M5 M6 M7
	R22	M2 M3 M5 M7 M8 M9
P3	R31	M4 M5 M6 M7 M8
	R32	M2 M3 M7 M8 M9
P4	R41	M2 M3 M4 M6 M7
	R42	M1 M5 M6 M8 M9
P5	R51	M4 M5 M7 M8 M9
	R52	M1 M2 M3 M5 M6
P6	R61	M2 M4 M5 M6 M7 M8 M9
	R62	M1 M3 M6 M7 M8 M9
P7	R71	M1 M2 M4 M5 M6 M9
	R72	M1 M2 M3 M7 M8 M9
P8	R81	M4 M5 M6 M7 M8 M9
	R82	M3 M4 M5 M7 M8 M9
P9	R91	M3 M5 M6 M7 M8 M9
	R92	M2 M4 M6 M7 M8 M9

TABLE II. TIMES OF OPERATION.

MAC.	PRODUCTS								
	P1	P2	P3	P4	P5	P6	P7	P8	P9
M1	428	439	453	403	481	446	414	491	458
M2	423	433	474	436	440	495	457	419	486
M3	459	487	417	410	477	474	452	435	416
M4	433	405	447	410	442	448	426	491	454
M5	467	447	486	400	450	469	493	495	452
M6	461	497	496	468	468	408	408	452	438
M7	464	495	459	489	436	454	457	477	484
M8	455	469	489	439	486	424	497	452	435
M9	418	439	480	457	435	482	445	408	416

TABLE III. TIME OF TRANSPORT IN AGVS.

	C	M1	M2	M3	M4	M5	M6	M7	M8	M9	D
C	0	14	5	10	12	14	10	13	11	9	13
M1	11	0	11	6	5	6	5	9	11	6	11
M2	6	9	0	7	5	6	10	6	9	11	13
M3	10	7	5	0	11	5	10	12	14	8	6
M4	7	13	8	9	0	10	11	6	14	7	9
M5	9	10	8	11	10	0	7	13	9	12	9
M6	14	11	13	7	14	6	0	12	13	14	8
M7	10	7	9	13	5	8	10	0	8	10	13
M8	11	6	9	6	8	10	12	13	0	9	7
M9	6	5	6	8	9	10	9	5	4	0	8
D	6	5	11	8	10	11	8	5	14	13	0

The average makespan found in the implementation I (GA using traditional chromosome [1]) was 5686 u.t..

The average makespan found in the implementation II (AGA using traditional chromosome [18]) was 5572 u.t..

The average makespan found in the implementation III (GA using new chromosome) was 5054.31 u.t., standard deviation of 170. The minimum value found was 4697 u.t. and the maximum was 5512 u.t..

The average makespan found in the implementation IV (AGA using new chromosome) was 5015.74 u.t., standard deviation of 152. The minimum value found was 4689 u.t. and the maximum was 5191 u.t..

The average makespan found in the implementation V (AGA with tabu search using traditional chromosome) was 5094.16 u.t., standard deviation of 155. The minimum value found was 4737 u.t. and the maximum was 5262 u.t..

The average makespan found in the implementation VI (AGA with tabu search using new chromosome) was 4914.97 u.t., standard deviation of 159. The minimum value found was 4645 u.t. and the maximum was 5177 u.t..

The best scheduling of 4 experiments was: routes (1, 1, 2, 1, 2, 2, 1, 1, 1) and scheduling of operations (3, 9, 7, 8, 9, 6, 4, 3, 2, 8, 9, 6, 7, 3, 1, 2, 7, 4, 9, 8, 3, 5, 1, 4, 2, 7, 6, 8, 9, 3, 1, 4, 5, 6, 8, 9, 2, 1, 5, 7, 4, 6, 8, 2, 7, 1, 5, 2, 6, 5, 2, 1) obtained in 4645 turns. The sequencing machine and AGV (prod (start-end)) is as follows in table 4:

TABLE IV. THE SEQUENCING MACHINE AND AGV

AGV1: 3(0-5), 7(5-25), 4(25-41), 9(426-438), 2(440-463), 3(479-497), 8(523-544), 6(885-901), 9(901-913), 3(914-933), 2(1324-1342), 9(1351-1373), 8(1385-1400), 6(1401-1418), 1(1752-1774), 4(1798-1816), 7(1816-1826), 8(1852-1871), 9(1876-1884), 3(1895-1904), 4(2226-2246), 6(2260-2272), 2(2308-2332), 9(2332-2355), 3(2384-2392), 1(2671-2691), 5(2702-2717), 4(2728-2750), 9(2800-2818), 8(2825-2848), 2(3076-3095), 7(3136-3157), 1(3158-3181), 5(3194-3212), 8(3256-3276), 4(3296-3314), 2(3605-3623), 7(3701-3723), 1(3760-3775), 5(4055-4072), 2(4120-4132), 6(4183-4201), 5(4588-4604), 2(4627-4645).
AGV2: 9(0-10), 8(10-32), 6(32-53), 7(480-491), 1(886-906), 4(915-933), 5(1325-1349), 7(1372-1388), 3(1392-1406), 2(1805-1821), 1(2228-2238), 5(2238-2262), 7(2319-2332), 8(2353-2373), 6(2807-2828), 6(3257-3266), 1(4601-4609).
Mac1: 7(25-439), 6(439-885), 2(885-1324), 1(1324-1752), 5(1752-2233).
Mac2: 3(5-479), 4(479-915), 7(915-1372), 2(1372-1805), 1(1805-2228), 5(2262-2702).
Mac3: 9(10-426), 3(497-914), 6(914-1388), 4(1388-1798), 2(1821-2308), 5(2717-3194).
Mac4: 8(32-523), 7(1388-1814), 4(1816-2226), 1(2238-2671), 2(2671-3076).
Mac5: 9(438-890), 8(890-1385), 7(1826-2319), 1(2691-3158), 2(3158-3605), 5(3605-4055).

Mac6: 9(913-1351), 8(1400-1852), 6(1852-2260), 4(2260-2728), 7(2728-3136), 2(3623-4120), 5(4120-4588).
Mac7: 3(933-1392), 9(1392-1876), 8(1876-2353), 6(2353-2807), 4(2807-3296), 1(3296-3760), 2(4132-4627).
Mac8: 3(1406-1895), 9(1895-2330), 8(2373-2825), 6(2828-3252).
Mac9: 3(1904-2384), 9(2384-2800), 8(2848-3256), 7(3256-3701), 6(3701-4183), 1(4183-4601).

Is shown in Figure 8 a comparison of the convergences of the experiments: GA with new chromosome, AGA with new chromosome and AGA + TS with new chromosome. In this figure is clear the advantage of AGA+TS over other approaches. The considered values were the averages of best makespan calculated by generation in 35 executions.

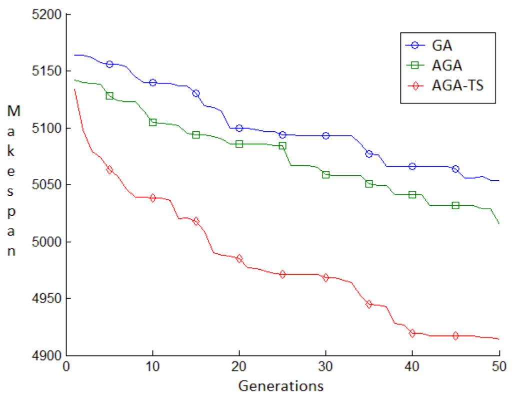

Fig. 8. Convergences of three experiments with new chromosome.

5. Analysis Of Results

The results are compared to those presented in [18] and [1], because these experiments have used the same scenarios and have searched the same objective (makespan).

Characteristics of these genetic algorithms:

- approach proposed by [18] - an adaptive genetic algorithm (AGA), with a population of 30, crossover starting at 0.6 and ranging from 0.5 to 1, mutation starting at 0,05 and ranging from 0.05 to 0.1, running on 100 generations or until convergence;
- approach of [1] - a genetic algorithm (GA), with a population of 200, crossover of 0.8, mutation of 0.4, running in 100 generations or until convergence.
- In all approaches apply the same operations of crossover and mutation, and were considered the same scenarios with 2 AGVs.

Table 5 presents the average makespan obtained in the proposal [1] and [18], well as of the fourth experiments conducted that are part of proposal this paper.

For more reliability of the results it was used one statistical test. Applying the Wilcoxon test to the sets of results, it can be stated that:

- GA [1] obtained the worst result;
- The combination of global search with local search (AGA+TS) obtained better results than the global search alone (GA or AGA), in both cases: traditional chromosome or new chromosome;
- GA and AGA both with new chromosome, were statistically tied;
- Comparing the same algorithms, but only changing the type of chromosome, in the three cases, the algorithms with new chromosome was better;
- Wilcoxon test confirm that AGA + TS with new chromosome is the best of all experiments.

TABLE V. AVERAGE OF MAKESPAN IN EXPERIMENTS.

Algorithms with traditional chromosome	**Makespan**	**Algorithms with new chromosome**	**Makespan**
GA	5686	GA	5054
AGA	5572	AGA	5015
AGA+TS	5094	AGA+TS	4914

Analyzing the Figure 8 it is possible to compare the convergence of the three search algorithms (GA, AGA and AGA + TS) using the new chromosome. Clearly hybridization of global search with local search has a better convergence over the generations.

Another point that has been observed is that the Tabu search with the traditional chromosome works as a global search, because to alter a gene of traditional chromosome, which equals a full route of the product, causes the new individual is not necessarily neighbor of the preceding. Other side in the new chromosome, each gene equivalent to only one product operation, so after the genetic operations, the new individual belongs to the neighborhood.

6. Conclusion And Future Work

This paper presents a proposal for production scheduling in manufacturing systems.

This paper is supported in two hypotheses. The first is the proposal of a new form of chromosome for the GA that makes the algorithm have greater adherence to the problem. The second is the combination of a global search (GA/AGA) with a local search (TS) to minimize the makespan.

Considering the results of average makespan found in experiments and presented in Table 4 it is possible to verify that the first hypothesis is valid. The second hypothesis also supported by the results achieved by the algorithms with new chromosome, which has been better in all tests. The global search with local search (AGA + TS) gets better results than global search alone (GA and AGA).

The proposal models every possible solution in production scheduling using a TPN considering the AGV and the input buffers of the machines with a free buffer flag to avoid interlock (deadlock).

This proposal also appears effective at treating the problem of production scheduling as reactive, in which the result is reached in time average 258.63 seconds in each experiment.

References

[1] B.S.P. Reddy, C.S.P. Rao, "A Hybrid Multi-Objective GA for Simultaneous Scheduling of Machines and AGVs in FMS", International Journal Advanced Manufacturing Technology, v.31, p. 601-613, 2006.

[2] J.A.B. Montevechi, O. Morandin Jr., P.E. Miyagi, Sistemas de Manufatura In: AGUIRRE, L.A. Enciclopédia de Automática, Controle & Automação. 1ed. São Paulo: Editora Blucher, 2007, p. 247-287.

[3] N. Slack, S. Chambers, R. Johnston, Administração de Produção, São Paulo, Atlas, 2002.

[4] F.T.S. Chan, S.H. Chung, P.L.Y. Chan, G. Finke, M.K. Tiwari, "Solving Distributed FMS Scheduling Problems Subject to Maintenance: Genetic Algorithms Approach", Robotics and Computer-Integrated Manufacturing, v 22, p. 493-504, 2006.

[5] S. Cavalieri, "Petri Nets and Genetic Algorithms to Increase Productivity in FMS". In: Second International Conference on Knowledge-Based Intelligent Electronic Systems, Adelaide, Australia, 1998. p. 134-142.

[6] P. Poncharoen, C. Hicks, P.M. Braiden, "The Development of Genetic Algorithms for the Finite Capacity Scheduling of Complex Products, with Multiple Levels of Product Structure. European Journal of Operational Research. n. 152, p. 215–225, 2004.

[7] S.J. Jeong, S.J. Lim, K.S. Kim, "Hybrid Approach to Production Scheduling using Genetic Algorithm and Simulation". International Journal of Advanced Manufacturing Technology, London, v. 28, p. 129 – 136, 2006.

[8] A.C. Deriz, "Um Método de Busca Usando Algoritmo Genético para Programação Reativa da Produção de Sistemas de Manufatura com Recursos Compartilhados", Masters Degree Dissertation, Federal University of São Carlos, 2007.

[9] O. Morandin Jr., A.C. Deriz, D.S. Sanches, E.R.R. Kato, "A Search Method using Genetic Algorithm for Production Reactive Scheduling Manufacturing Systems", IEEE International Symposium on Industrial Electronics - ISIE2008, 2008, Cambridge. Proceedings of the ISIE2008, 2008.

[10] Y. Yin, J. Yiu, Z. Cheng, "A Genetic Algorithm Based Approach to Flowshop Scheduling", Proceedings of the 5" World Congress on Intelligent Control and Automation, Hangzhou, P.R. China, June, 2004.

[11] L. Zhang, L. Wang, D. Zheng, "An Adaptive Genetic Algorithm with Multiple Operators for Flowshop Scheduling", International Journal Advanced Manufacturing Technology, v. 27, p. 580-587, 2006.

[12] F.T.S. Chan, S.H. Chung, P.L.Y. Chan, "An Adaptive Genetic Algorithm with Dominated Genes for Distributed Scheduling Problems". Expert Systems with Applications, v 29, p. 364-371, 2005.

[13] F. Taghaboni, "Scheduling and control of manufacturing systems with critical material handling". Ph. D. Thesis – Purdue University, West Lafayette, Indiana, 1989

[14] G. Ulusoy, F. Sivrikaya, U. Bilge, "A Genetic Algorithm Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles". Computers & Operations Research, vol. 24, pp. 335-351, 1997.

[15] J. Jerald, P. Asokan, R. Saravanan, A.D.C. Rani, " Simultaneous Scheduling of Parts and Automated Guided Vehicles in an FMS Environment using Adaptive Genetic Algorithm". International Journal Advanced Manufacturing Technology, v.29, p. 584-589, 2006.

[16] S.S. Sankar, S.G. Ponnambalam, R. Rajkumar, M. Gurumarimuthu, " An Intelligent Integrated Scheduling Model for Flexible Manufacturing System", Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, December, 2004, p.1095-1100.

[17] E.G.R. Maggio, "Uma Heurística Para a Programação da Produção de Sistemas Flexíveis de Manufatura Usando Modelagem em Redes de Petri", Masters Degree Dissertation, Federal University of São Carlos, 2005.

[18] O. Morandin Jr.; E.R.R. Kato; D.S. Sanches; B.D. Muniz, "Modeling strategy by adaptive genetic algorithm for production reactive scheduling with simultaneous use of machines and AGVs," IEEE International Conference on Systems, Man and Cybernetics, pp.704-709, 11-14, 2009.

[19] J. Kennedy; R. Eberhart, "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks.IV. pp. 1942–1948, 1995.

[20] P. Damodaran; D.A. Diyadawagamage; O. Ghrayeb; M.C. Vélez-Gallego, "A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines "International Journal of Advanced Manufacturing Technology,v 58, n 9-12, p 1131-1140, 2012.

[21] H. Samarghandi; T.Y. ElMekkawy, "A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion" International Journal of Advanced Manufacturing Technology, p 1-14, 2011.

[22] J. Li; Q. Pan; P.N. Suganthan; M.F. Tasgetiren, "Solving fuzzy job-shop scheduling problem by a hybrid PSO algorithm" Lecture Notes in Computer Science, v 7269 LNCS,p 275-282, 2012.

[23] D. Li; N. Deng, "Solving Permutation Flow Shop Scheduling Problem with a cooperative multi-swarm PSO algorithm" Journal of Information and Computational Science,v 9, n 4,p 977-987, 2012

[24] P. Damodaran; A.G. Rao; S. Mestry, "Particle swarm optimization for scheduling batch processing machines in a permutation flowshop" International Journal of Advanced Manufacturing Technology,p 1-12,2012

[25] O. Morandin Jr.; E.R.R. Kato; C.C.M. Tuma, "A strategy of production scheduling with the fitness function of genetic algorithm using Timed Petri net and considering AGV and the input buffer", IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, pp.1311-1316, 7-10, 2010.

[26] A. Sinha; D.E. Goldberg, "A Survey of Hybrid Genetic and eEvolutionary Algorithms", Illinois Genetic Algorithms Laboratory, 22p, 2003.

[27] A. Sinha; D.E. Goldberg, "Verification and extension of the theory of global-local hybrids", Proceedings of the Genetic and Evolutionary Computation Conference, 592-598, 2001.

[28] G. Vilcot; J.C. Billaut, "A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem", European Journal of Operational Research, Elsevier, v. 127, n. 2, p. 398-411, 2008.

[29] M.A. González; C.R. Vela; M. Sierra; R. Varela, "Tabu search and genetic algorithm for scheduling with total flow time minimization" Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems, p 33-41, 2010

[30] Y. Huang; X. Yao; D. Ge; Y.S. Li; "Entropy-enhanced genetic algorithm with tabu search for job shop scheduling problems" Advanced Materials Research, Mechatronic Systems and Materials Application, v 590, p 557-562, 2012.

[31] S. Meeran; M.S. Morshed, "A hybrid genetic tabu search algorithm for solving job shop scheduling problems: A case study" Journal of Intelligent Manufacturing, v 23, n 4, p 1063-1078, 2012.

[32] Q. Zhang; H. Manier; M.A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times" Computers and Operations Research, v 39, n 7, p 1713-1723, 2012.

[33] F. Glover, "Tabu Search—Part I" ORSA Journal on Computing Summer, 1:190-206, 1989.

[34] F. Glover, "Tabu Search—Part II" ORSA Journal on Computing Winter, 2:4-32, 1990.