PSO in 2D-space to Solve Reactive Scheduling Problems in FMS to Reduce the Makespan

Orides Morandin Jr.[1], Carlos C.M.Tuma[1], Vinicius F. Caridá[1], Emerson C. Pedrino[1], Roberto H. Tsunaki[2]
[1]Department of Computer Science - Federal University of São Carlos
(orides, carlos_tuma, vinicius_carida, emerson)@dc.ufscar.br
[2] Department of Mechanical Engineering - EESC - University of São Paulo
rtsunaki@sc.usp.br

Abstract—**Production reactive scheduling considering simultaneous use of machines and AGVs in Flexible Manufacturing Systems (FMS) involves modeling and searching methodology in a wide solution space. Several researches have been using the Genetic Algorithms (GA) as a search method to solve this problem since these algorithms have the capacity of globally exploring the search space and find good solutions quickly. Due to the characteristics of the problem, this paper proposes the use of Particle Swarm Optimization (PSO) to solve reactive scheduling problems with the minimization of makespan in FMS. The proposal was tested in two factory scenarios and validated by comparing its results with those obtained by techniques based on genetic algorithm and adaptive genetic algorithm.**

***Keywords*—PSO, Genetic Algorithms, Production Scheduling, FMS.**

I. 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].

Flexibility means the ability to change the operation, i.e., can change what the operation does, how it does or when it does [3].

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 [5] [6] [7] [8] [9].

Other approaches make use of Adaptive Genetic Algorithms 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].

Based on the characteristics of the problem, other metaheuristic can provide good results, like PSO. PSO is originally attributed to Kennedy, Eberhart [19] and makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. Also PSO has been proposed at several papers and obtained positive results [20] [21] [22] [23] [24].

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 a methodology for finding solutions to reactive production scheduling in FMS with AGVs simultaneous modeling using TPN with input buffer, AGVs and control flag, and the programming of the output of PSO Algorithm and using order of AGVs by simple rules.

II. PARTICLE SWARM OPTIMIZATION

The PSO algorithm mimics the behavior of flying birds, called particles, and each bird means a solution possible. The initial population consists of random solutions, and the search occurs by updating generations.

The original PSO works with real numbers, and for scheduling problems, real numbers must change to positive integers.

Each particle s has many informations: position of particle in space p; vector of velocity $\vec{v}$; fitness of particle f; memory of best local solution encountered pb, fb (pos, fitness).

For each iteration of algorithm, all particles were recalculated, using the follows equations:

$$v(t+1) = \alpha 1 v(t) + \beta 1\big(pb - p(t)\big) + \beta 2\big(gb - p(t)\big) \qquad (1)$$

Where:

- $\alpha 1, \beta 1, \beta 2$ are random numbers in (0,1);
- gb, is the position of global best fitness in that iteration.

$$p(t+1) = p(t) + v(t+1) \qquad (2)$$

Convergence is the natural way for PSO algorithm, because each particle flies in the dimensional problem space with a velocity and direction adjusted according to the fitness seen in the all swarm.

III. ADDRESSING THE PROBLEM USING PSO AND 2D PROBLEM MAPPING

The purpose of this study was to develop a strategy for modeling production scheduling in FMS using TPN considering input buffers, AGVs and free buffer flags, using:

- Particle Swarm Optimization with complete scheduling in one particle, address as coordinates in 2D space;
- Modeling of Jobs and AGVs in TPN, using the input buffer and free buffer flags, see [25];
- 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. See [25] for details.

FMS have a finite number of solutions, however very large the extent to which the problem grows.

This number can be calculated according to the expression below:

$$ns = np * nc \qquad (03)$$

Where:

- ns is a number of possible solutions;
- np is a number of permutations on jobs;
- nc is a number of combinations between all routes of each job.

These two numbers can be easily interpreted as the X and Y axes of the two-dimensional space of solutions, and by means of an efficient encoding, each point in this space indicates a single viable solution. Both axes are limited to begin at 1 and finish in the maximum values of their indicators.

The population of particles of PSO was created randomly, with the generation of two random numbers, one indicating the permutation of the solution and the other indicating the combination of the result. A routine generates the respective sequencing. The values of positions generated in iterations in PSO are real numbers, which are truncated to integers.

During the iterations of PSO, the points of the Cartesian plane will be moving toward a global optimum, and is possible to verify this migration in 2d graphics.

In Figure 1 can be checked the convergence form of PSO, which begins a global search with random starting points (red points of the extremities) and will converge to better solutions (step 2 - green points and step 3 - blue points).

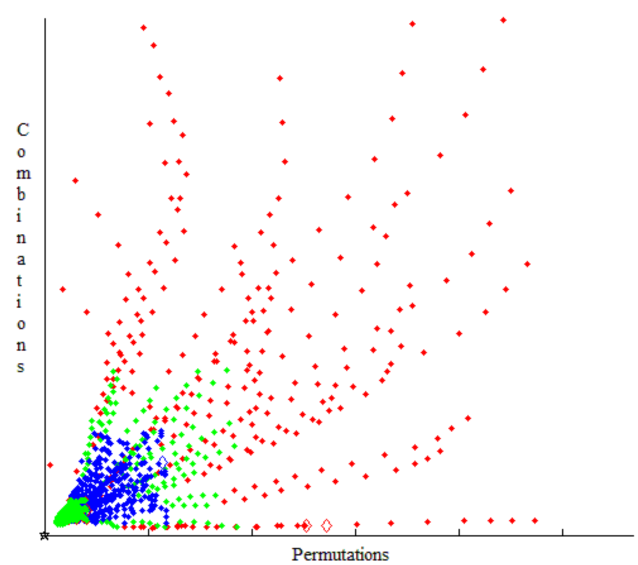

Figure 1 – Convergence of PSO

In figure 2 is shown a convergence with fewer solution points for better visualization. Can be seen that the search begins with the points (solutions) generating a trail in blue toward the point marked 1. Then starts a new convergence towards point 2, with trace green and finally in the last converging in red towards the point 3.

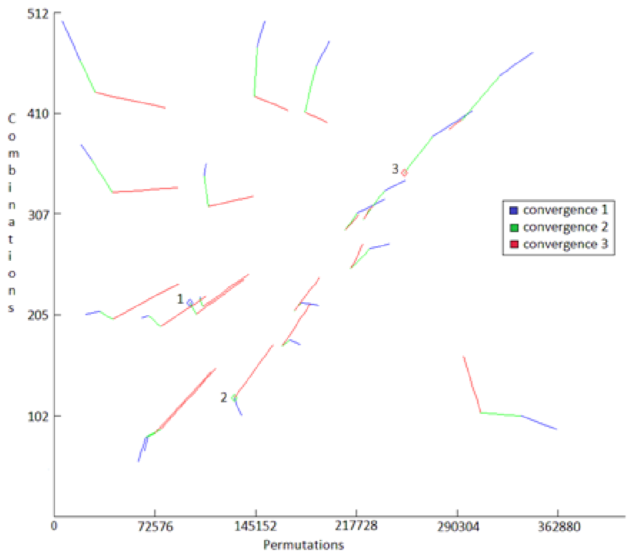

Figure 2 – Representation of a search with PSO in 2D-space

The other structure necessary for this research is described in the paper [25].

The problem of reactive scheduling production includes several products that can be produced through several different routes. What is sought is a series of products to be followed in production such a way that corresponds to a desired performance goal. Several performance measures are used today, and this paper uses the minimum makespan objective. The results were compared with some papers 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 discharge 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 production of Scheduling provided.

IV. IMPLEMENTATION

A system to evaluate the proposal was performed using the following steps:

Step 1 - Read the production scenario through data files, as shown in tables: Table 1 (Products & Routings) where Mx is the machine x, Table 2 (Operating time) and table 3 (Times of AGVs in the transportation). These tables describe the scenario to be used in a test of the proposed system (Experiment 1, section V.1);

Step 2 - The search for possible solutions is made through the use of PSO algorithm. The modeling of the particle indicates permutation, combination and best fitness.

The PSO parameters used are:

- Population of 20 particles;
- Creation for new population: using expression in 3;
- The fitness of each particle used in PSO algorithm is obtained by the equation:

$$F(i) = 1/mkp\,; \qquad (04)$$

Where:

- $F(i)$ is the fitness function of particle;
- i and mkp is the makespan of the same particle;
- Stopping criterion:
    - Experiment 1: 5 iterations (covering 92.5% of the search space);
    - Experiment 2: 50 iterations (covering 0.0005% of the search space).

Step 3 - The solutions are dynamically mapped in TPN. It is possible to see an example of mapping production with the use of AGV in Figure 4 in [25].

The fitness is calculated through the construction of a possible path in the tree coverage, following certain rules, i.e., the production scheduling and allocation rules of AGV.

Step 4 - From the mapping of all jobs in the TPN solution, it's created the marking vector and vector transition in which the facility is based on production to run the program and see how many turns each solution takes to reach its conclusion. This number of turns is the minimum makespan.

Step 5 - During the execution of scheduling, the rules of order of the AGVs follow criteria: always choose the AGV with the lowest number index, and always choose requisitions (transitions) with the lowest number index during the modeling.

Step 6 - At the end of PSO algorithm, the final solution is presented as the particle and its implementation in terms of lists of performed operations and in the form of Gantt chart.

V. COMPUTER EXPERIMENTS

Two batteries of tests were performed, each with 35 results. Each experiment used a specific scenario, both with 2 AGVs. These results are presented in Table 7.

Best schedulings, as well as the sequence obtained from their performances are presented and a Gantt chart of the best scheduling in Experiment 1 is obtained from sequencing.

All results are in terms of turns required to complete all jobs (makespan). Table 7 presents all results obtained in both experiments.

A. *EXPERIMENT 1*

The Experiment 1 consists of running 35 times of the proposed algorithm, within a scenario of 3 products, with 6 machines and several routings for each product.Table I, Table II and Table III present a detailed description of this scenario.

TABLE I. PRODUCTS AND ROUTES - EXPERIMENT 1

PRODUCTS	ROUTES	
P1	R11	M1,M2,M3,M4,M5
	R12	M1,M2,M3,M6
P2	R21	M1,M4,M5,M6
	R22	M2,M4,M5,M6
	R23	M3,M4,M5,M6
P3	R31	M1,M5,M6
	R32	M2,M5,M6
	R33	M3,M4,M5,M6

TABLE II. TIMES OF OPERATION - EXPERIMENT 1.

MACHINES	PRODUCTS		
	P1	P2	P3
M1	434	458	472
M2	452	443	465
M3	400	405	469
M4	472	485	459
M5	460	402	432
M6	421	435	445

TABLE III. TIMES OF TRANSPORT IN AGVS - EXPERIMENT 1

	C	M1	M2	M3	M4	M5	M6	D
C	0	4	6	8	14	12	10	6
M1	10	0	3	5	11	9	7	14
M2	12	15	0	3	9	7	9	5
M3	14	17	15	0	7	9	11	8
M4	8	11	9	7	0	3	5	10
M5	6	9	7	9	15	0	3	7
M6	4	7	9	11	17	15	0	14
D	4	7	9	11	17	15	0	0

The average makespan found for Experiment 1 was 2235.11 u.t., standard deviation of 13.16. The minimum value found was 2228 u.t. and the maximum was 2297 u.t..

The best scheduling of Experiment 1 was: 3-2, 2-3, 1-2 obtained in 2228 turns, and generating the following sequence machine and AGV (prod (start-end)):

Below is the best scheduling obtained for this problem:

- AGV1: 3(0-6), 1(6-22), 2(413-425), 1(456-470), 3(471-478), 3(910-913), 1(923-935), 2(1322 -1334), 3(1358 -1372), 2(1793-1807), 1(2214 -2228).
- AGV2: 2(0-8), 2(910-920), 1(1359-1379).
- Mac1: 1(22-456).
- Mac2: 3(6-471), 1(471-923).
- Mac3: 2(8-413), 1(935-1335).
- Mac4: 2(425-910).
- Mac5: 3(478- 910), 2(920-1322).
- Mac6: 3(913-1358), 2(1358-1793), 1(1793- 2214).

Note that the Product1 delivery is anticipated in the input buffer of the machine 6, advanced its production, it does not require waiting for transport when the current operation finishes (Figure 7, detail).

It is noticed that the machine 2 and 6 don't have transport intervals, from the moment they receive their first shipment. This shows that its vehicles were always early, leaving a raw material in the input buffer.

These sequences led to the Gantt chart shown in Figure 3.

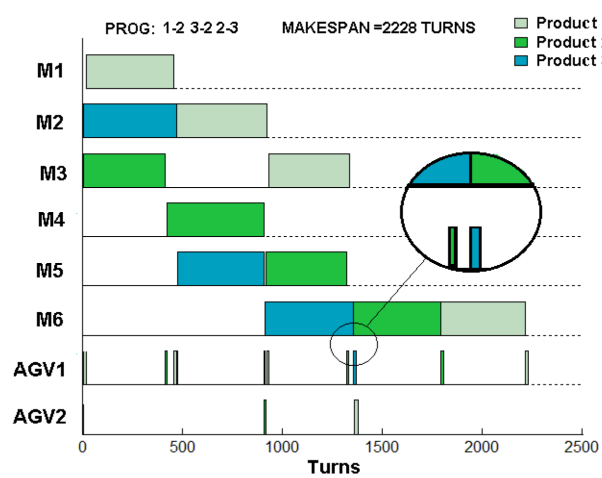

Figure 3 - Gantt chart of production

B. *EXPERIMENT 2*

The Experiment 2 consists of running 35 times of the proposed algorithm, within a scenario of 9 products, with 9 machines and several routings for each product. Table IV, Table V and Table VI present a detailed description of this scenario.

TABLE IV. PRODUCTS AND ROUTES - EXPERIMENT 2

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 V. TIMES OF OPERATION - EXPERIMENT 2.

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 VI. TIME OF TRANSPORT IN AGVS - EXPERIMENT 2

	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 for Experiment 2 was 4847.60 u.t., standard deviation of 147.17 . The minimum value found was 4650 u.t. and the maximum was 5087 u.t..

The best scheduling of Experiment 2 was: 7-1, 6-2, 3-2, 9-1, 2-1, 1-1, 5-2, 4-1, 8-1 obtained in 4650 turns. The sequencing machine and AGV (prod (start-end)) is as follows:

- AGV1: 7(0-14), 4(14-30), 8(30-48), 2(429-450), 3(479-497), 8(539-560), 6(874-890), 9(900-912), 3(914-933), 2(1313-1331), 9(1350-1372), 7(1372-1386), 6(1388-1407), 8(1408-1421), 1(1741-1763), 4(1798-1816), 7(1816-1826), 9(1876-1897), 3(1898-1907), 5(2222-2238), 1(2238-2243), 8(2267-2290), 2(2308-2332), 6(2332-2346), 3(2387-2404), 1(2676-2696), 4(2735-2754), 6(2770-2787), 9(2803-2811), 2(3081-3101), 7(3143-3164), 5(3168-3181), 6(3285-3305), 2(3610-3628), 7(3730-3752), 1(3760-3775), 5(4060-4077), 2(4125-4137), 8(4138-4156), 1(4556-4577), 5(4593-4609), 2(4632-4650).
- AGV2: 3(0-5), 9(5-21), 6(21-45), 9(437-448), 7(480-501), 1(875-895), 4(915-933), 5(1314-1338), 3(1392-1409), 2(1805-1821), 6(1821-1843), 4(2226-2242), 7(2319-2332), 9(2333-2355), 5(2678-2691), 8(2807-2827), 1(3163-3186), 8(3286-3303), 4(3303-3321).
- Mac1: 7(14-428), 6(428-874), 2(874-1313), 1(1313-1741), 5(1741-2222).
- Mac2: 3(5-479), 4(479-915), 7(915-1372), 2(1372-1805), 1(1805-2228), 5(2238-2678).
- Mac3: 9(21-437), 3(497-914), 6(914-1388), 4(1388-1798), 2(1821-2308), 5(2691-3168).
- Mac4: 8(48-539), 7(1386-1812), 4(1816-2226), 1(2243-2676), 2(2676-3081).
- Mac5: 9(448-900), 8(900-1395), 7(1826-2319), 1(2696-3163), 2(3163-3610), 5(3610-4060).
- Mac6: 9(912-1350), 6(1407-1815), 8(1815-2267), 4(2267-2735), 7(2735-3143), 2(3628-4125), 5(4125-4593).
- Mac7: 3(933-1392), 9(1392-1876), 6(1876-2330), 8(2330-2807), 4(2807-3296), 1(3296-3760), 2(4137-4632).
- Mac8: 3(1409-1898), 9(1898-2333), 6(2346-2770), 8(2827-3279).
- Mac9: 3(1907-2387), 9(2387-2803), 6(2803-3285), 7(3285-3730), 8(3730-4138), 1(4138-4556).

Note the progress of the job 6 at mac1: raw material transported by AGV2 and delivered on time 45 and starting the processing in mac1 in time 428.

It is noticed that the machine 1, 2 and 9 do not have transportation waiting intervals, from the moment it receives its first load. This shows that its vehicles were always early, leaving a raw material in the input buffer.

C. *ANALYSIS OF RESULTS*

The results of this paper are compared with the results presented in [18], [1] and [25], because these experiments have used the same scenarios and have searched the same objective (makespan).

Characteristics of genetic algorithms presented in papers:

- 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 0.005 and ranging from 0.0 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 0.40, running in 100 generations or until convergence;
- Approach of [25] - a genetic algorithm (GA), with a population of 20, crossover of 0.8, mutation of 0.05, running in 5/100 generations or until convergence.
- In all approaches apply the same operations of crossover and mutation, and were considered the same scenarios with 2 AGVs.

In Table VII there are the average makespan found in the 3 approaches using GA and PSO approach in this paper using two scenarios.

TABLE VII. - AVERAGE OF MAKESPAN IN EXPERIMENT I AND EXPERIMENT II FOR THE METHODS COMPARED

Experiment	PSO	AG [25]	AGA[18]	AG [1]
Experiment I	2235	2233	2354	2373
Experiment II	4847	4948	5572	5686

Applying the Wilcoxon test to the sets of results, it can be stated that the proposed method showed better results and statistically different than the results presented in [25], [1] and [18] with 5% significance level.

The processing times consumed in calculating the experiments I and II are very close to the ones obtained in the other approaches.

VI. CONCLUSION

This paper presented a proposal for reactive production scheduling in manufacturing systems. 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 261.32 seconds in experiment 2.

Analyzing the final results, it is possible to see that this proposal and the proposal [25] are better. The two proposals reached the best makespan, but in average and statistically, the PSO is better than GA. Another important fact is the logical coverage of the PSO, while the GA is totally random.

ACKNOWLEDGMENT

The authors thank the CAPES and CNPq for providing scholarships and the project INCT SEC for the support and investment in the research.

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.