A Hybrid Differential Evolution Approach for Simultaneous Scheduling Problems in a Flexible Manufacturing Environment

M.V. Satish Kumar, KITS, Singapur, A.P, INDIA

C.S.P Rao, NIT, Warangal, A.P, INDIA

G.Rangajanardhan, JNTU, Kakinada, A.P, INDIA

Abstract

Scheduling of machines and transportation devices like Automated Guided Vehicles (AGVs) in a Flexible Manufacturing System(FMS) is a typical N-P hard problem. Even though several algorithms were employed to solve this combinatorial optimization problem, most of the work concentrated on solving the problems of machines and material handling independently. In this paper the authors have attempted to schedule both the machines and AGVs simultaneously, with makespan minimization as objective, for which Differential Evolution (DE) is applied. Operations based coding is employed to represent the solution vector, which is further modified to suit the DE application. The authors have proposed two new strategies of DE in this paper which better suits the problem. We have developed a separate heuristic for assigning the vehicles and this is integrated with the traditional DE approach. The hybridized approach is tested on a number of benchmark problems whose results outperformed those available in the literature.

Keywords

Flexible Manufacturing System, Evolutionary Algorithms, Integrated scheduling, Genetic Algorithms, Differential Evolution.

Introduction

Flexible Manufacturing System can be treated as the promising technology which has the capability not only to produce parts at lower cost but also to face challenge of demand for variety of products in mid volumes, with shorter lead times. A number of advantages like increased utilization of resources, reduction in work in- progress inventory, improvement in productivity, better utilization of floor space are possible in an FMS. To achieve all these goals and to perform well FMS should be properly designed and operated. One of the important aspects in FMS operation is its scheduling policy, by which its resources like machines and automated material handling system can be utilized properly. Material handling is among the crucial elements of an FMS and for successful implementation of FMS, better coordination between machines and material handling is very important. Hence there is a need for scheduling of material handling system along with the machines, which makes the scheduling of FMS more complex than the general job shop scheduling.

Among the alternatives available for automated material handling in autonomous manufacturing environments, Automated Guided Vehicles system (AGVs) is finding increasing applications, because of its capability to transport variety of part types from point to point without human intervention. The shop floor can be controlled easily through an intelligent computer system, because of the flexibility of AGVs, that they can be integrated with the other computer control production and storage equipment. In this paper the authors have applied a new evolutionary algorithm, Differential Evolution (DE) to schedule the FMS under consideration. The heuristic developed for vehicle assignment is integrated with the DE, for simultaneous scheduling of machines and automated guided vehicles. Different DE strategies are available in the literature and each one performs better in a particular problem. The existing strategies were tested for the problems considered and two new strategies are proposed by modifying them slightly to achieve better performance.

Literature Survey

Several researchers have laid thrust on the importance of the material handling system (Raman 1986, Green Wood 1988, Bozer 1989, Hom and Mc Ginnis 1989, Kouvelis.P 1992).

Some of the researchers reported material handling system scheduling as a comparison of a set of vehicle dispatching rules in relation to a pre specified schedule and on a particular layout (Egbelu and Tanchoco 1984). Equal importance was attributed for making scheduling of AGVs an integral part of the overall scheduling activity. The outcome of the co-ordination of machines and material handling system during the machine scheduling phase would be expected to raise the performance of the FMS.

Simultaneous scheduling of machines and AGVs was attempted in different ways like developing on line dispatching and control rules (Wu and Wysk, 1989, Ro and Kim 1990, Sabuncuoglu and Hommertz Heim, 1992, Sawik 1993). A beam search based algorithm was developed for the simultaneous scheduling of machines and AGVs (Karabuk and SabunCuoglu, 1993) Umit Bilge and Gunduz Ulusoy,(1995) have applied sliding time windows approach to this problem. They formulated the problem as two sub problems one for machine scheduling and the other for vehicle scheduling which interact through a set of time window constraint for the material handling trip starting times. They have developed an iterative procedure, in which a new machine schedule is generated by a heuristic procedure and the operation completion times obtained from this schedules were used to construct time windows for the trips and a feasible solution was searched for the second sub problem, which was again handled as a sliding time window problem. The same authors have applied genetic algorithms for solving the simultaneous scheduling problem. They have used a suitable coding scheme, in which chromosomes are fixed length strings containing two consecutive locations to represent both dimensions of the search space: Operation sequencing and AGV assignment. Smith et al.,(1999) studied the job shop scheduling problem ,while considering the loading and unloading processes they emphasized on the importance of addressing material handling. Tamer.F. Abdelmaguid et al., (2004) have developed a hybrid GA procedure, which uses operation based coding for scheduling machines. They have also developed a heuristic to solve the vehicle scheduling, because of which they have reduced the length of the chromosome to half that created by Ulusoy. Lacomme et al., (2005) addressed the job input sequencing and vehicle dispatching in a single vehicle automated guided vehicle system. They have coupled the heuristic branch and bound approach with discrete event simulation model. Siva P.Reddy et al., (2006) have attempted the same problem set as that of Ulusoy and Tamer with a modified GA approach. Jerald et al.,(2006) have used an adoptive genetic algorithm for solving the simultaneous scheduling of parts and AGVs problem.

Application of Differential Evolution algorithm to the multiple machine flow-shop scheduling problem, the single machine total weighted tardiness problem, and the single machine common due date scheduling problem was investigated Andreas C. Nearchou et al.,(2006). Their intention was to show how DE can address a wide range of combinatorial problems with discrete decision variables. Quan-Ke Pan et al.,(2007) have presented a new novel discrete differential evolution (DDE) algorithm with a local search to solve the permutation flow shop scheduling problem with the make span criterion. Jihui Zhang et al.,(2007), have proposed a stochastic method based on the differential evolution (DE) to address a wide range of discontinuous optimization problems such as scheduling and multi-item inventory control.

Differential Evolution

Evolutionary Algorithms (EA) which simulates evolution process in computer have created lot of interest among the researchers, which led to their application in a variety of fields. Genetic Algorithms, GA (Holland, J.H., 1975) are popular among the EAs and they were used to address scheduling problems by many researchers. Even though GA can be considered as a better searching algorithm and many versions of GA were developed by several researchers, still it is considered as an algorithm with less convergence speed (Gopalakrishna. A. 2006). To overcome this difficulty Differential Evolution (DE) was proposed recently, which can be considered as an improved version of GA and it has comparatively good convergence speed.

Kenneth Price got the idea of using vector differences for perturbing the vector population, when he was attempting to solve Chebychev Polynomial fitting Problem, which was posed to him by Rainer Storn (Rainer Storn, Kenneth Price, 1995). Since its inception it has undergone a number of experimentations and computer simulations and finally yielded as a versatile and robust tool, as it is today. DE, which can be considered as a new member in the family of evolutionary algorithms, also uses operators like, crossover and mutation (Dervis Karboga, Selcuka Okdem.,2004). Very similar to genetic algorithm, it also starts with an initial population of solutions but here all of them will be represented by vectors. Here in this population based algorithm, all the "NP" randomly generated initial vectors will be replaced with the better ones (generated by the specific variation operators)in every generation and this process repeats till the termination criteria is satisfied. Once an initial population of "NP" vectors is created, their fitness will be calculated based on the objective criteria under consideration. Even though the remaining steps in DE, like crossover and mutation are same as GA, their operation is different here, which are described below.

Mutation

Unlike GA, where mutation follows crossover, in DE mutation will be performed first. Three vectors x_{r1}, x_{r2} , x_{r3} which are different from the current vector will be randomly selected and the weighted difference of two vectors in the population is added to a third vector to get the resultant vector known as mutant vector ($v_{i, g+1}$), as given below

$$V_{i,g+1} = x_{r1,g} + F(x_{r2,g} - x_{r3,g})$$

Where F > 0 is scaling factor, which controls the magnitude of the differential variation of $(x_{r2,g} - x_{r3,g})$.

Crossover

Mutant vector obtained in the previous step and target vector under consideration are subjected to crossover, to generate trial vector ($U_{i,\,g+1}$). This operator controls the amount of diversity of mutant vectors.

$$U_{ji,g+1} = \begin{cases} V_{ji,g+1} & \text{if } rn_j \le C_r \\ X_{ji,g} & \text{otherwise} \end{cases}$$

Where j= 1, 2, ----- D

$C_r \in [0,1]$ is the crossover constant which represents the probability of trail vector that inherits parameter values from the mutant vector and D represents the number of dimensions of a vector.

Boundary check

In boundary-constrained problems, the parameter values of the trial vectors need to be checked whether they lie with in the range or not. In case they violate the boundary constraint they should be adjusted.

Selection

The trial vector generated in the crossover step and the target vector under consideration are compared to select the member for the next generation. The selection is based on the factor that whether the target vector is producing the smaller function value or the trial vector and accordingly it will be selected, as given below:

If $f(u_{i,\,g+1}) \leq f(x_{i,g})$, set $x_{i,\,g+1} = u_{i,\,g+1}$

Otherwise $x_{i,\,g+1} = x_{i,\,g}$

Termination

Termination of the above steps can be made in two ways. One is predefining the number of generations for which this procedure has to be repeated and the other is, stopping the procedure when there is no further improvement in the obtained result.

Even though the steps involved in DE and GA are similar, in GA crossover plays a major role where as in DE mutation is crucial. Only fittest parents will get chance to participate in mating, in case of GA, where as in DE all the parents will be given equal chance. In GA there is a possibility that inferior off-springs may replace the better parents but this does not arise in DE as all the children have to compete with their parents to get place in the next generation, which makes it to perform better (Gopalakrishna. A. 2006). In addition, DE is a fast converging algorithm using few control parameters and yet yields a global optimum solution, which created interest among many researchers and in many engineering fields it has been applied successfully to solve optimization problems.

Problem Statement

In this paper the authors have considered an FMS environment and attempted to schedule both machines and AGVs, using Differential Evolution approach. Given a set of jobs, each containing a number of operations, which have to be performed on predefined machines, and the jobs are to be moved in between machines using automated guided vehicles, it is aimed to schedule both the machines as well as vehicles. The objective criteria considered here is minimization of makespan, which can be defined as time interval between the starting of the first operation and the finishing of the last operation.

While solving the problem, the following assumptions were made:

The number of jobs and the number of operations in each job and also their corresponding machines as well as their processing times are known. Each machine is capable of performing more than one type of operation, but once an operation is started it can not be interrupted. One important constraint considered in this problem is precedence constraint, according to which an operation of a particular job can not be started unless its preceding operation of the same job is completed. A predefined number of AGVs are used to carry parts in between machines. It is assumed that all the vehicles can carry unit load at a time and all of them moves with equal speed and issues like traffic congestion, control, machine failure and down time will be considered during the real time control.

The FMS environment under consideration is having a Load/Unload station where all the jobs and vehicles are present before the beginning of operations. Both the L/U station as well as machines is assumed to have sufficient buffer space. Initially the AGVs are at L/U station and later moves in between L/U to machine or machine to machine depending upon the operation to be performed at that instance. AGV may be moving empty or carrying a weight and accordingly the trips are known as deadheading trips or loaded trips. We assume that all the design and setup issues within the hierarchy of OR/MS problems in an FMS as suggested by Steeke (1985) have already been resolved. It is also assumed that the allocation of tools, pallets and other necessary equipment are completed. The routing of each part type is known before making decisions on scheduling.

Even though the FMS environment considered here and the one considered by Raman, Talbot (1986) and Rachmadugu (1990) are similar, there is a slight difference in operating policy. According to them, AGV always returns to the L/U station after completion of every task where as in our case AGV moves to L/U station only for picking up jobs, whose first operation is to be performed.

In this paper, the authors have considered a set of 82 problems, which were earlier studied by Ulusoy, Tamer and Siva P. Reddy. Ten different job sets each with a specific number of operations are considered in combination with four layouts having separate traveling timings for AGV movements. The problems are grouped into two sets, one with relatively high t/p ratio, whose value is >0.50 and the other set comprising of problems with relatively lower value of t/p, generally <0.25 (i.e. those problems having very less travel times when compared to process times) The first group of problems are generally denoted by 1.1, 1.2,…..10.4, indicating first job set with first layout to tenth job set with fourth layout. The second group of problems are formulated by making the processing times double and traveling times half, that of the problems considered in first group, which are denoted by 1.10, 1.20, ……10.40. In the second group it self, another sub group of problems were formed by taking the processing times, triple to those in first group and making the travel times half of those in first group, which are denoted as 2.41, 3.41,…….7.41.

Proposed Methodology

In this paper the authors have applied the Differential Evolution algorithm for the above problems and developed “C” code to implement all the steps of DE. In the proposed approach every possible schedule is represented by a chromosome, in which each gene represents an operation to be performed. Operations based coding (Gen and Cheng, 1997), is implemented in this paper, where a two digit number is used to represent a gene. The first digit indicates the job number and the second digit indicates the operation number in that particular job. For example, 23 indicates third operation in second job. Once an initial population of required size (which is generally 10 times the total number of operations in that particular job) is created, each chromosome is checked for precedence constraints and repaired if necessary. For repairing the invalid sequences separate module has been developed.

While finding the mutant vector from the three randomly selected solution vectors, the actual positions of the genes in the chromosomes are used instead of two digit numbers, which are obtained by using a conversion module. This module converts the gene under consideration to its position in the chromosome and once mutation operation is carried out, it converts back the new position obtained to its corresponding genes. Once mutant vectors are obtained in terms of their positions, boundary check will be done and a correction module is developed to adjust the position obtained, so that it falls with in the boundary. The next step is to perform crossover on target vector and mutant vector, to get trial vector and finally target vector and trial vector are compared to select the member for the next generation, based on the minimum makespan criteria.

A separate heuristic is developed for assigning the vehicle at every operation. This heuristic checks which vehicle can reach the desired machine at the earliest and accordingly assigns the vehicle for that operation. This vehicle assignment algorithm is incorporated in the conventional DE approach and the resulting hybrid DE is applied to the problem under consideration. In this way the machines or in other words operations as well as vehicles scheduling is integrated and the best sequence of operations is obtained.

Even though different strategies are available in the literature, a particular strategy may work well for a particular problem and hence in this paper the authors have attempted the problems using 5 different strategies available in the literature to test the results. In addition the authors have also proposed two strategies and applied them to same set of problems. The best results obtained among all the strategies are compared with the results available in the literature for the set of 82 problems.

The following strategies are proposed by the authors, by slightly modifying the strategies available in the literature.

Proposed strategy 1: $x_{r1.g} + F1(x_{r2.g} - x_{r3.g}) + F2(x_{r4.g} - x_{r5.g})$

Proposed strategy 2: $x_{i.g} + F1\,(x_{best.g} - x_{r1.g}) + F2(x_{r2.g} - x_{r3.g})$

In the above proposed strategies, the terms mentioned are considered with the same meaning as and when they are used in the existing strategies i.e here also $x_{r1.g}$, $x_{r2.g}$, $x_{r3.g}$, $x_{r4.g}$ and x_{r5g} are randomly selected vectors where as $x_{i.g}$ is the current vector and $x_{best.g}$ is the vector with minimum makespan in that particular generation. Here also F1&F2 are scaling factors.

Tuning the parameters

While applying to the problem under consideration, the DE parameters are set as follows:

The value of NP was varied from two to ten times total the number of operations in the job. It is observed that the algorithm gave good results when NP is eight times the length of the chromosome.

Similarly the crossover point is varied between 0.1 and 0.9 and finally found that, at the value of 0.5 it gave good results. The other important parameter F is also tried with different values and finally got good results when F is taken as 0.8. Whenever F1 & F2 are involved in a particular strategy their values are taken as 0.8 & 0.6 respectively. Another important criterion to be considered is termination, which decides the number of generations. In this paper the authors have run the algorithm a number of times changing the generations from 100 to 3000, from which it is observed that the algorithm yielded good results when the number of generations are in between 1500 and 2000.

Results and Discussions

The authors have applied hybrid DE (HDE) to the above said 82 problems which were already solved by Ulusoy, Tamer and Siva P Reddy. The results obtained are compared with those published by Ulusoy, who solved the problems using sliding time windows (STW) as well as genetic algorithms (UGA). The results are also compared with the hybrid GA proposed by Tamer (TGA) as well as S.P.Reddy (SGA).

It is found that, DE outperformed other results in 8 problems and gave equally good results in 23 problems among the first group 40 problems, where t/p ratio is >0.5. When it comes to second group of 42 problems having t/p ratio <0.25, the proposed algorithm outperformed in 24 problems and is on par with the best known results in 15 problems. The problems where HDE performed better when compared to other algorithms are shown in the graphs below. The proposed hybrid DE appears to be better in problems with less t/p ratio (<0.25) which is shown clearly in the graphs.

Problem No.	T/P Ratio	STW	UGA	TGA	SGA	HDE
1.1	0.59	96	96	96	96	96
1.2	0.47	82	82	82	82	82
1.3	0.52	84	84	84	84	84
1.4	0.74	108	103	103	103	103
2.1	0.61	105	104	102	100	100
2.2	0.49	80	76	76	76	76
2.3	0.54	86	86	86	86	86
2.4	0.77	116	113	108	108	108
3.1	0.59	105	105	99	99	99
3.2	0.47	88	85	85	85	85
3.3	0.51	86	86	86	86	86
3.4	0.74	116	113	111	111	111
4.1	0.91	118	116	112	112*	115
4.2	0.73	93	88	88	87	85*
4.3	0.80	95	91	89	89*	94
4.4	1.14	126	126	126	126	126
5.1	0.85	89	87	87	87	87
5.2	0.68	69	69	69	69	69
5.3	0.74	76	75	74	74	74
5.4	1.06	99	97	96	96*	104
6.1	0.7	1	1	118	118	1

	8	20	21			15*
6.2	0.54	100	98	98	98*	100
6.3	0.54	104	104	104	103	103
6.4	0.78	120	123	120	120*	124
7.1	0.78	119	118	115	111*	112
7.2	0.62	90	85	79	79	79
7.3	0.68	91	88	86	83	83
7.4	0.97	136	128	127	126	126
8.1	0.58	161	152	161	161	161
8.2	0.46	151	142	151	151	143*
8.3	0.50	153	143	153	153	132*
8.4	0.72	163	163	163	163	163
9.1	0.61	120	117	118	116	104*
9.2	0.49	104	102	104	102	94*
9.3	0.53	110	105	106	105	100*
9.4	0.76	125	123	122	122	113*
10.1	0.55	153	150	147	147*	149
10.2	0.44	139	137	136	135*	137
10.3	0.49	143	143	141	139	139
10.4	0.69	171	164	159	158*	164

Table 1: comparison of test results for problems with t/p ratio >0.5

Problem No.	**T/P Ratio**	**STW**	**UGA**	**TGA**	**SGA**	**HDE**
1.10	0.15	126	126	126	126	125*

1.20	0.12	123	123	123	123	122*
1.30	0.13	122	122	122	122	121*
1.40	0.18	124	124	124	124	124
2.10	0.15	148	148	148	148	147*
2.20	0.12	143	143	143	143	142*
2.30	0.13	146	146	146	146	143*
2.41	0.13	217	217	217	217	217
3.10	0.15	150	148	150	150	147*
3.20	0.12	148	145	145	145	143*
3.30	0.13	149	146	146	146	144*
3.40	0.18	151	151	151	151	149*
3.41	0.12	222	221	221	221	221
4.10	0.15	121	119	119	119	119
4.20	0.12	116	114	114	114	114
4.30	0.13	116	114	114	114	114
4.41	0.19	179	172	172	172	172
5.10	0.21	102	102	102	102	102
5.20	0.17	100	100	100	100	99*
5.30	0.18	99	99	99	99	99
5.41	0.18	154	148	148	148	148
6.10	0.16	186	186	186	186*	192
6.20	0.12	183	181	181	181*	190
6.30	0.24	1	1	1	182	1

		84	82	82		82
6.40	0.19	1 85	1 84	1 84	184	1 84
7.10	0.19	1 37	1 37	1 37	137	1 05*
7.20	0.15	1 36	1 36	1 36	136	1 33*
7.30	0.17	1 37	1 37	1 37	137	1 28*
7.40	0.24	1 38	1 37	1 37	137	1 31*
7.41	0.16	2 03	2 03	2 03	203	2 03
8.10	0.14	2 92	2 71	2 92	292	2 43*
8.20	0.11	2 87	2 68	2 87	287	2 46*
8.30	0.13	2 88	2 70	2 88	288	2 47*
8.40	0.18	2 93	2 73	2 93	293	2 65*
9.10	0.15	1 76	1 76	1 76	176	1 68*
9.20	0.12	1 74	1 73	1 73	173	1 65*
9.30	0.13	1 76	1 74	1 74	174	1 66*
9.40	0.19	1 77	1 75	1 75	175	1 67*
10.10	0.14	2 38	2 36	2 38	238	2 38
10.20	0.11	2 36	2 38	2 36	236	2 25*
10.30	0.12	2 37	2 41	2 37	237	2 37
10.40	0.17	2 40	2 44	2 40	240 *	2 47

Table 2 : comparison of test results for problems with t/p ratio < 0.25

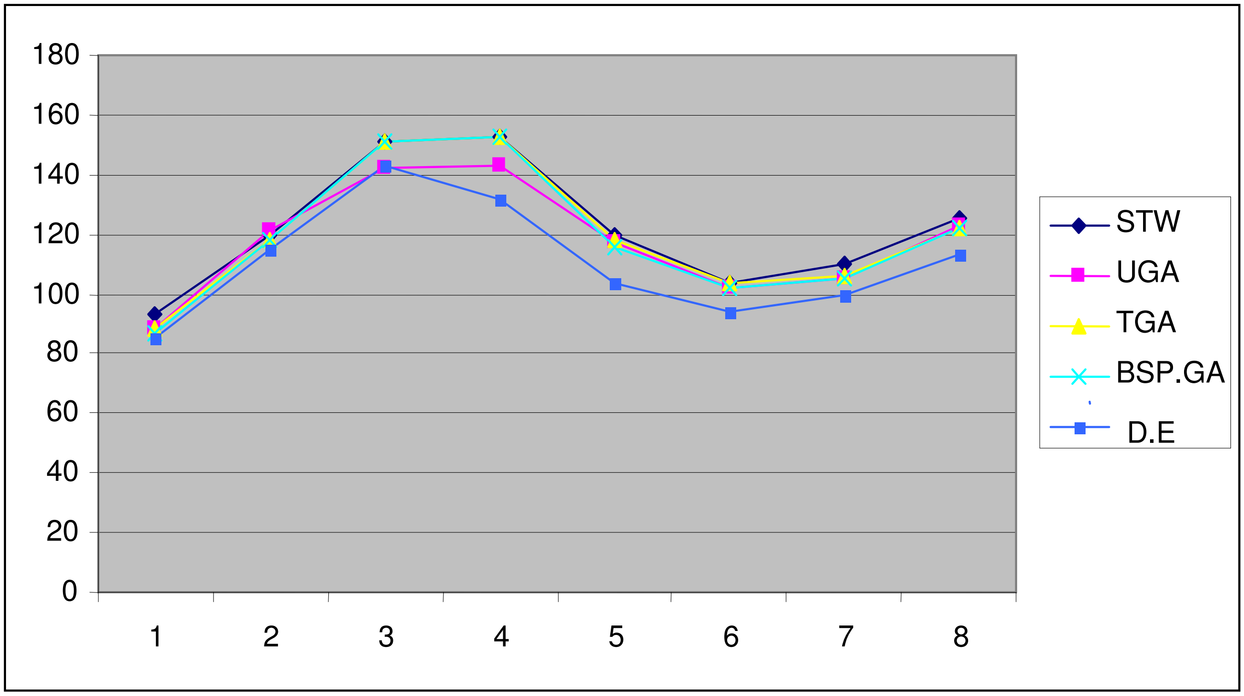


Figure 1: Graph showing the performance of DE compared with other methods (for problems with t/p ratio > 0.5).

Figure 2: Graph showing the performance of DE compared with other methods (for problems with t/p ratio < 0.25.

Conclusions

In today's globally competitive environment every organization must concentrate on effective utilization of resources rather than simply expanding the infrastructure, for which reason scheduling is playing a vital role. To satisfy the customer, flexibility in manufacturing is inevitable now-a-days, which can be achieved by FMS. Even though many of the researchers concentrated on the scheduling of an FMS, most of them dealt the problem as two sub problems i.e. scheduling of machines and scheduling of material handling system. In this paper we have made an attempt to integrate scheduling of both machines and AGVs as a single problem and solved it using Differential Evolution (DE). The proposed algorithm has been applied on a plethora of bench mark problems, which gave reasonably good results. It is also observed that the algorithm even outperformed the other existing methods. The results reveal that, especially for problems having t/p ratio less than 0.25, the proposed algorithm yielded better results.

References

Andreas C. Nearchou, Sotiris L. Omirou. (2006) "*Differential evolution for sequencing and scheduling optimization*", J Heuristics 12: 395–411.

Bozer, Y.A.(1989). *"Guided vehicle systems: information/control system implication of alternative design and operation strategies"*. In Advanced Information Technologies for Industrial Material Flow Systems, pp. 417--436. Springer-Verlag, Berlin.

Dervis Karboga, Selcuka Okdem.(2004) "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm", Turk J Elec Engin, Vol.12, No.1

Egbelu, P.J., and Tanchoco. J.M.A. (1984), *"Characterization of Automated Guided Vehicle Dispatching Rules"*, International Journal of Production Research, 22 (3), 359-374

Gen, M and Cheng, R. (1997) *"Genetic Algorithms and Engineering Design",* (New York: Wiley)

Gopalakrishna. A. (2006). *"Optimization of surface grinding operations using a differential evolution approach",* Journal of Materials Processing Technology , pp 1-8.

Greenwood, N. R. (1988) "*Implementing Flexible Manufacturing Systems"*. Macmillan, London.

Gundiz Ulusoy, Funda Srvnkaya – Serifogulg, Umit Bilge(1997) "A Genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles", Computers & Operations Research, Vol.24, No.4, pp.335-351

Han, M-H., and McGinnis, L.F.(1989*) "Control of Material Handling Transporter in Automated Manufacturing"*, IIE Transactions, 21 (2), 184-190.

Holland, J.H.(1975). "*Adaptation in Natural and Artificial Systems",* The University of Michigan Press, Ann Arbor, Michigan.

Jerald , P. Asokan , R. Saravanan , A. Delphin Carolina Rani, (2006) *"Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm",* Int J Adv Manuf Technol 29: 584-589.

Jihui Zhang and Junqin Xu(2007) *"A New Differential Evolution for Discontinuous Optimization Problems"*. Third International Conference on Natural Computation (ICNC), IEEE, pp 5-9.

Karabuk. S. and SabunCuoglu.I(1993). "*A beam search based algorithm for scheduling machines and AGVs in an FMS",* Second Industrial. Engg. Res. Conf., pp 308-312.

Kouvelis, P. (1992) *"Design and planning problems in flexible manufacturing systems: a critical review",* J. Intelligent Manufacturing , 3, pp75-99.

Lacomme .P, Moukrim.A, Tchernev. N (2005) "Simultaneous job input sequencing and vehicle dispatching in a single-vehicle automated guided vehicle system: a heuristic branch-and-bound approach coupled with a discrete events simulation model" International Journal of Production Research, Vol.43, No.9, pp1911-1942.

N.Raman, F.B Talbolt, R.V.Rachamadugu,(1986)"Simultaneous scheduling machines and material handling devices in automated manufacturing",Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems.

Quan-Ke Pan, M. Fatih Tasgetiren and Yun-Chia Liang,(2007) "A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem" GECCO'07, London, England, United Kingdom, pp. 126-133.

Rachamadugu, R. and Schriber, T. J. (1990) *"Performance of dispatch rules under perfect sequence flexibility",* In Proc. of the Winter Simulation Conf., pp. 653-658.

Rainer Storn, Kenneth Price, (1995) *"Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces",* Technical Report TR- 95012,International Computer Science Institute, Berkley.

Ro, I. and Kim, J. (1990) *"Multi-criteria operational control rules in flexible manufacturing systems"*. Int. J. Prod. Res., 28, 47-63.

Sabuncuoglu, I. and Hommertzheim, D. L.(1992) "Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system". Int. J, Prod. Res., 30, 1059-1079.

Sawik, T. (1993) *"Scheduling of machines and vehicles in an FMS: single-level versus multi-level approach",* Automatyka *,* 64, 541-568.

Siva P. Reddy. B, C.S.P. Rao (2006). *"A Modified G A approach for simultaneous scheduling of machines and AGVs in FMS",* International Journal of Manufacturing Science and Technology, Vol.8, No.1, pp 48-58 .

Smith, J.S., Peters, B.A. and Srinivasan, A.(1999*). "Job shop scheduling considering material handling"*, Int. J, Prod. Res., 37, 1541-1560.

Stecke, K.E.(1985), *"Design, Planning, Scheduling and control problems of Flexible Manufacturing Systems",* Annals of Operations Research, 3, 3-12.

Tamer F.Abdelmaguid, Ashraf. O.Nasef, Badawia. A.Kamal &Mohamed.F.Hassan, (2004) "A hybrid GA / heuristic approach to the simultaneous scheduling of machines and automated guided vehicles", International Journal of Production Research ,Vol.42, No.2, pp267-281.

Umit Bilge and Gunduz Ulusoy(1995). "A time window approach to simultaneous scheduling of machines and material handling system in an FMS", Journal. Of Operations Research, Vol.63, No.6, November-December.

WU, S.Y.D. and WYSK, R. A. (1989). "An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing", International Journal of Production Research, 27, 1603–1623.

Appendix - I

Data for the job sets considered.

Job Set 1:

Job1: M1(8); M2(16); M4(12)

Job2: M1(20); M3(10); M2(18)

Job3: M3(12);M4(8);M1(15)

Job4: M4(14);M2(18)

Job5: M3(10); M1(15)

Job Set 2:

Job1: M1(10);M4(18);

Job2: M2(10); M4(18)

Job3: M1(10); M3(20)

Job4: M2(10);M3(15);M4(12)

Job5: M1(10);M2(15);M4(12)

Job6: M1(10);M2(15);M3(12)

Job Set 3:

Job1: M1(16);M3(15)

Job2: M2(18);M4(15)

Job3: M1(20);M2(10)

Job4: M3(15):M4(10)

Job5: M1(18);M2(10); M3(15);M4(17)

Job6:M2(10);M3(15);M4(8);M1(15);

Job Set 4:

Job1: M4(11);M1(10);M2(7)

Job2: M3(12);M2(10); M4(8)

Job3: M2(7);M3(10);M1(12):M3(8)

Job4: M2(7);M4(8);M1(12);M2(6)

Job5: M1(9);M2(7);M4(8);M2(10;M3(8)

Job Set 5:

Job1: M1(6); M2(12); M4(9)

Job2: M1(18); M3(6); M2(15)

Job3: M3(9);M4(3);M1(12)

Job4: M4(6);M2(15)

Job5: M3(3); M1(9)

Job Set 6:

Job1: M1(19);M2(11):M4(7)	Job2: M1(19);M2(20); M4(13)
Job3: M2(14);M3(20);M4(9)	Job4: M2(14);M3(20);M4(9)
Job5: M1(11);M3(16);M4(8)	Job6 : M1(10);M3(12);M4(10)

Job Set 7:

Job1: M1(6);M4(6)	Job2: M2(11);M4(9)
Job3: M2(9);M4(7)	Job4: M3(16);M4(7)
Job5: M1(9);M3(18)	Job6: M2(13); M3(19); M4(6)
Job7: M1(10); M2(9); M3(13)	Job8: M1(11); M2(9); M4(8)

Job Set 8:

Job1: M2(12); M3(21); M4(11)	Job2: M2(12); M3(21); M4(11)
Job3: M2(12); M3(21); M4(11)	Job4: M2(12); M3(21); M4(11)
Job5: M1(10); M2(14); M3(18); M4(9)	Job6: M1(10); M2(14); M3(18); M4(9)

Job Set 9:

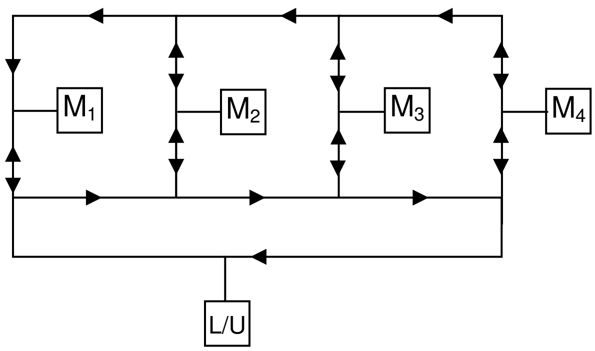
Job1: M3(9);M1(12);M2(9);M4(6)	Job2: M3(16);M2(11);M4(9)
Job3: M1(21);M2(18):M4(7)	Job4: M2(20);M3(22);M4(11)
Job5: M3(14);M1(16);M2(13);M4(9)	

Job Set 10:

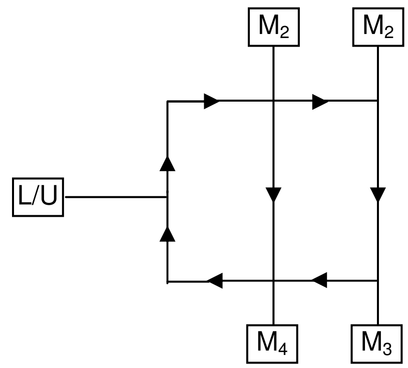
Job1: M1(11);M3(19);M2(16);M4(13)	J0b2: M2(21);M3(16);M4(14)
Job3: M3(8);M2(10);M1(14);M4(9)	Job4: M1(13);M3(20);M4(10)
Job5: M1(9);M3(16); M4(18)	Job6: M2(19); M1(21); M3(11) M4(15)

Appendix - II

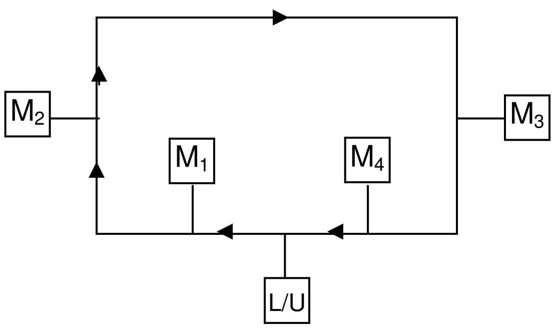
Layout Configurations



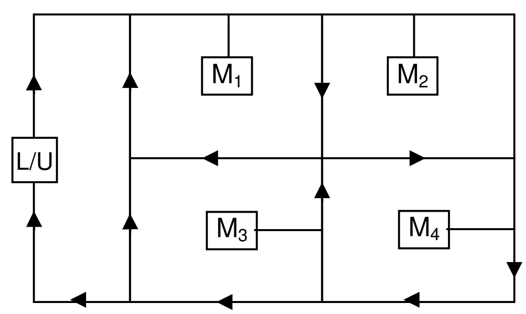
Layout – 1



Layout – 2



Layout – 3



Layout – 4

	L/U	M_1	M_2	M_3	M_4
L/U	0	2	4	10	12
M_1	12	0	2	8	10
M_2	10	12	0	6	8
M_3	4	6	8	0	2
M_4	2	4	6	12	0

Layout-3: Travel Time Matrix

	L/U	M_1	M_2	M_3	M_4
L/U	0	4	8	10	14
M_1	18	0	4	6	10
M_2	20	14	0	5	6
M_3	12	8	6	0	6
M_4	14	14	12	6	0

Layout-4: Travel Time Matrix