A Comparative Representation Approach To Modern Heuristic Search Methods In A Job Shop

P.D.D.Dominic, Universiti Teknologi PETRONAS, Malaysia

Ahmad Kamil, Universiti Teknologi PETRONAS, Malaysia

P.Parthiban, National Institute of Technology, Tiruchirappalli, India
M.Punniyamoorthy, National Institute of Technology, Tiruchirappalli, India

Abstract

The job shop problem is among the class of NP- hard combinatorial problems. This Research paper addresses the problem of static job shop scheduling on the job-based representation and the rule based representations. The popular search techniques like the genetic algorithm and simulated annealing are used for the determination of the objectives like minimizations of the makespan time and mean flow time. Various rules like the SPT, LPT, MWKR, and LWKR are used for the objective function to attain the results. The summary of results from this paper gives a conclusion that the genetic algorithm gives better results in the makespan time determination on both the job based representation and the rule based representation and the simulated annealing algorithm gives the better results in the mean flow time in both the representations.

Keywords

Introduction

A general job shop scheduling problem (JSSP) can be defined as n job, m

machine problem, where the aim is to optimally allocate series of operations for each job

across available machines, respecting temporal and resources constraints. Scheduling in

the job shop is an important aspect of a shop floor management system, which has

significant impact on the performance of the shop floor (Holthaus and Rajendran, 2000). The job shop problem may be characterized by a number of jobs to be processed, in which jobs consists of one or more operations to be performed in a specified machine with in amount of time. JSSP is worst among the NP hard problems and among the worst combinatorial problems. Not only it is hard, but even among the members of the latter class, it appears to belong to the more difficult ones (Wang and Zheng, 2001).

Many researchers used search algorithms to solve combinatorial optimization problems. Some authors have shown that the optimal schedules can be found the set of active schedules based upon regular performance measures (Giffler and Thomson 1960, Kenneth Baker 1974). Genetic Algorithms (GA) are stochastic, guided search strategy. They work on a population of points, and follow the Darwinian principle of survival of the fittest in nature. A population of solutions is evolved to a next set of solutions through genetic operators, and mechanism of selection just like the process of evolution in nature (Dirk and Christian, 2004). Simulated Annealing (SA) is a stochastic heuristic algorithm in which the solutions are searched for in hill climbing processes constantly commenced by random moves. Because of its ease of use, SA is an extremely popular method for solving large-sized and practical problems like job shop scheduling, time tabling and traveling salesman (Aydin, 2004).

Modern heuristic search for job shop scheduling

There have been a number of papers, in which meta-heuristics is used to solve job shop scheduling problems. In this section, the literature that addresses the scheduling problems in the job shop manufacturing context is reviewed.

Genetic Algorithms

Genetic Algorithm is an adaptive search method used to solve hard combinatorial problems. GA mimics the mechanics of natural selection and evaluation. In any evolutionary process, only the most suited elements in a population are to survive and generate offspring, thus transmitting biological heredity to the new generations. In a similar way, GA also starts with a particular initial population and subsequent generations are created using generic operators. Population is a set of strings, each representing a potential solution to the given problem. Each string is called a chromosome and the elements of the chromosome are called genes. For creating the offspring only promising string are selected so that the generic material is transferred effectively. `Generic operators' are applied on the mating pool to create offspring. The set of strings chosen to create offspring is called mating pool. After the creation of new set of strings, the population is evaluated and only the best elements are chosen for the next generation and finally the search ends with a best possible set of solutions.

Hybrid GA (Kopfer and Mattfeld, 1997) proposed by the authors that applies a local search operator in an evolutionary framework and evaluated with the results obtained for a well-known test suite of particularly hard, but medium-sized job shop scheduling problems (including the famous MT 10x10 problem). Their results encourage using GA framework for future research.

The researchers (Ponnabalam and Jawahar, 1999) proposed to solve job shop scheduling problems with the makespan objective. Three perturbation schemes were used to study the effect on the problems considered. Which were pair wise exchange, insertion, and random insertions. The performance of the Simulated Annealing (SA) has

been compared with the GA often SA gives better results. The CPU time was also very close to the time of Genetic Algorithm.

Solving stochastic job shop scheduling problems (Yasunari Yoshitomi, 2002) based on a genetic algorithm; it was expanded for stochastic programming. The roulette strategy is adopted for selecting the optimum solution in terms of the expected value. Within this algorithm, it is expected that the individual that appears most frequently must give the optimum solution. The effectiveness of this approach is confirmed by applying it to stochastic job shop scheduling problem.

An approach (Ivan, 2004) for scheduling of customers orders in factories of plastic injection machines as a case of real world flexible job shop scheduling problem. As an approach addressing the issue of efficient scheduling routine a hybrid evolutionary algorithm combining priority dispatching rules with GA is developed. The results obtained for evolving a schedule of 400 customers orders on experimental model of factories of plastic injection machines indicate that the business delays in order of half-an-hour can be achieved.

An approach (Goncalves, 2005) of hybrid genetic algorithm for job shop scheduling problem proposed. The chromosome representation of the problem is based on random keys. The schedules are constructed using a priority rule in which the priorities are defined by the genetic algorithm. This approach is compared with other approaches. The computation results validate the effectiveness of the proposed algorithm.

Open job shop scheduling (Senthilkumar, 2006) is a kind of job shop scheduling in which operations can be performed in any order. An approach was developed a heuristic for the open job shop scheduling problem using GA to minimize makespan. The results are statistically compared and found to be significantly better than the earlier reported results.

To minimize production losses (Yu-Su Shum and Dah-Chuan Gong, 2007) due to unanticipated machine breakdown, a preventive maintenance plan must be implemented to ensure machines are kept in good condition. When the system size increases, with many more machines, the corresponding mathematical programming becomes complicated. The proposed GA problem solving procedure gives the solution quality.

Simulated Annealing Algorithms

The Simulated Annealing (SA) is a search technique proposed by (Kirkpatrick et al, 1983) this is based on the statistical mechanics and motivated by an analogy to the annealing of solids in statistical physics where a solid is cooled to the ground state noting its behavior at intermediate temperature. On the other hand it can be used for improvement approach to combinatorial optimization problems. As in solid annealing, this method starts with an initial temperature. The temperature is reduced slowly and at each step, the solution is repeatedly improved by making small local perturbations until no further improvements. But SA technique differs from other search methods in that respect and it accepts inferior solutions with certain probability. In this way it attempts to reduce the danger of getting stuck with local optimal solutions. It has been found that the mentioned proposing simulated annealing for scheduling problem show good results.

The simulated annealing technique appears to be a robust approach to scheduling problems.

Two results (Kolonko, 1999) about heuristic solutions to the job shop scheduling problems are presented. The novel features are an adaptive temperature control that allows reheating of the SA and a new type of time oriented crossover of schedules. Though the procedure uses only standard properties of the job shop scheduling it yields excellent results on the classical test examples.

The authors (Esin Onbasoglu, Linet and Ozdamar, 2001) developed five different parallel simulated annealing algorithms and compare them on an extensive test bed used previously for the assessment of various solution approaches in global optimization. Previous attempts with other approaches, such as sequential SA, adaptive partitioning algorithms and clustering algorithms, to identify the global optima of these functions have failed without exception.

New stochastic learning algorithm developed (Albrecht et al, 2001, 2002) and first results of computational experiments on fragments of liver CT images. The algorithm is designed to compute a depth-three threshold circuit, where the first layer is calculated by an extension of the perceptron algorithm by a special type of simulated annealing. They present a pattern classification method that combines the classical perceptron algorithm with simulated annealing. For a sample set S of n-dimensional patterns labeled as positive and negative, their algorithm computes threshold circuits of small depth where the linear threshold functions of the first layer are calculated by simulated annealing with the logarithmic cooling schedule.

A hybrid algorithm (Purushothama and Jenkins, 2003) has been developed for the solution of the unit commitment problem. This hybrid technique uses simulated annealing as the main algorithm. At each temperature, fresh solutions are generated randomly, and with a high likelihood of being feasible. Local search is made in the neighborhood of the best solution, using a heuristic decommitment technique. The hybrid algorithm is robust and has improved convergence, compared with earlier algorithms. The results obtained in system studies indicate its potential for solving the unit commitment problem.

The evolutionary simulated annealing (Aydin and Fogarty, 2004) algorithm, its distributed implementation and its application to two combinatorial problems are presented. This consists of a population, a simulated annealing operator, instead of the more usual reproduction operators used in evolutionary algorithms, and a selection operator. The problems tackled are well known combinatorial optimization problem, namely, the classical job shop scheduling problem and the uncapacitated facility location problem. They used to measure the efficiency of metaheuristics with respect o both the quality of the solutions and the central processing unit (CPU) time spent.

The new approach (Saber et al, 2006) presented to the fuzzy unit commitment problem using the absolutely stochastic simulated annealing method. All the solutions, both higher and lower cost, are associated with acceptance probabilities, for example the minimum membership degree of all the fuzzy variable. Besides the number of bits flipping is decided by the linguistic fuzzy control. Excess units with system dependent distribution handle constraints efficiently and reduce overlooking the optimal solution.

To reduce the economic load dispatch calculations, a sign bit vector is introduced with imprecise calculation of the fuzzy model as well.

Job Based Representation

This job based (Esquivel et al, 2002; Viswanath Kumar Ganesan, 2006) representation consists of a list of *n* jobs and a schedule is constructed according to the sequence of jobs. For a given sequence of jobs, all operations of the first job in the list are scheduled   first, and then the operations of the second job in the list are considered .The first operation of the job under treatment is allocated in the best available processing time for the corresponding machine the operation requires and the second operation is allocated, and so on, until all operations of the job are scheduled. This process is repeated with each of the jobs in the list considered in the appropriate sequence.

Any permutation of the jobs corresponds to the schedule. Suppose a chromosome is given as [2,3,1]. The first job to be processed is job2. The operation precedence constraint for the job j2 is [m1, m2, m3] and the corresponding processing time for each machine is [3,2,3]. Each of its operation is scheduled in the best available processing time for the corresponding machine the operation requires.

Priority Rule Based Representation

In a priority rule based (Ivan, 2004; Jayamohan, 2004; Goncalves, 2005; Tsung, 2007) genetic algorithm, a chromosome is encoded as a sequence of dispatching rules for job assignment and a schedule is constructed with priority dispatching heuristic based on the sequence of dispatching rules. Priority dispatching rules are the one most frequently applied heuristics for solving scheduling problems in practice because of their easy implementation and their low time complexity.

In this paper the Priority dispatch rules used are:

1.   SPT (Shortest imminent Processing Time)

2.   LPT (Least imminent Processing Time)

3.   LWKR (Least Work Remaining)

4.   MWKR (Maximum Work Remaining)

Methodology

This research paper interested to bring out suboptimal solutions through metaheuristics like Genetic Algorithm and Simulated Annealing methods. Heuristic techniques are often motivated by the question 'what is the best way to generate a good solution' while a genetic algorithm targets the specific question 'what is the best solution to the problem'. Two representations like Priority Dispatching Rules based and job based representations are consider to select the operations.   Priority rules are type of heuristic rules, which tend to guide the scheduling process. Job based representation uses a list of n jobs, the schedule from which is constructed according to the sequence of jobs. The chromosome is a set of permutation of jobs, defining the scheduling priority of each job with respect to others. Ten bench mark problems are taken for both representations the performances are compared on each of the algorithms with the objective measures of makespan time and mean flow time**.**

GA Job Based Representation

Determination of makespan and mean flow time in the job based representation is done through the inputs of number of machines and number of jobs. Once these are known, the routing matrix is got considering that all the jobs go through all machines. From this source   the population size is got from the user. The population size represents the different combination of the routings. On this different combination the makespan and the mean flow time of the schedule is got from this makespan and the mean flow times.

The   probability of occurrence is got by dividing the makespan by the sum of makespan and similarly the mean flow time by the sum of mean flow times. This gives the probability of the function. The total sum of the probabilities gives the cumulative probability of the individuals, which gives '1' now a mutation probability, is given. This determines a value between the probabilities the corresponding chromosomes are selected. Crossover is applied here to determine a new set of schedule this gives a new set of population and makespan and the mean flow time goes on till the best of the makespan and the mean flow time is achieved.

A genetic algorithm to any problem would involve the following stages:

- Representation of feasible solution to the problem as chromosomes and generation of the initial population.

- Evaluation of the population using fitness function.

- Generation of new population using the generic operators.

- Selection of new population using the offspring generated.

Representation of the chromosome

A population consists of strings called the chromosomes, each representing a sequence of schedule. The schedules form a makespan and the mean flow time out of the process involved in each of them.

Creation of initial population

Population is a set of chromosome, which consists of different schedules, each representing the same sequence of operations given by the user.

Evaluation of the population

Survival of the fittest is the concept of genetic algorithm. The resulting makespan and the mean flow time is the fitness function of the chromosome. It is done by the determination of the makespan and the mean flow time then the probability of occurrence is calculated summing up the makespans and the mean flow times individually and getting divided with each individual makespan and the mean flow time. The sum of them gives the cumulative probability of the population.

Generation of offspring

The mating pool is selected from the population; the crossover operator is used in the process to get the new offspring. These new offspring gives the next set of generation.

Crossover

Crossovers as the name suggests swaps a portion of the two parent chromosomes of the mating pool to create two new chromosomes for the new generation single point crossover is used in the process. In this a single point is selected and they are interchanged. This results in a new set of chromosome, which undergoes the process.

Termination

After the crossover is over the new set of chromosomes are derived, this set of chromosomes are used to for further iterations and the optimal results are obtained.

GA Rule Based Representation

In this paper Giffler Thompson algorithm is used for the determination of makespan and the mean flow time. The rules are used here for solving the conflicts occurring in the problem. When a problem arises for usage of rule, in this case the genetic algorithm is used to determine the rule to be used at the correct place to get the optimal result.   In this process the genetic algorithm is used as follows. The set of rules are selected and the genetic algorithm concept is applied to the input.

SA Job Based Representation

The simulated annealing technique is used to determine the makespan and the mean flow time of the job based representation. The simulated annealing uses the solid annealing technique to under take the process. In this technique an initial temperature is set up and from that initial temperature the improvement is made through several neighborhood temperatures to attain the best solution. In this problem the makespan and the mean flow time is considered as the temperature.

The generic procedure applied for the SA to function

Step 1: Get an initial solution, S.

Step 2: Set an initial temperature T>0.

Step 3: While not frozen, do the followings.

Step 3.1: Do the following L times.

Step 3.1.1: Sample a neighbour NS from S.

Step 3.1.2: Let Δ = value (NS)-value (S).

Step 3.1.3: lf $\Delta< 0$, then set S = NS.

Else

S = NS with the probability exp. (-Δ/T)

Step 3.2: Set T = r x T ,   where r is reduction factor.

Step 4: Return S.

Here the makespan or the mean flow time initially obtained is considered as the initial temperature. The temperature of the neighbor point or the schedule is considered. The makespan or the mean flow time obtained is then compared with the previous; the better solution is then considered with the usual increment in the simulated annealing. This process is continued for further iterations then the optimal result is got from the iterations.
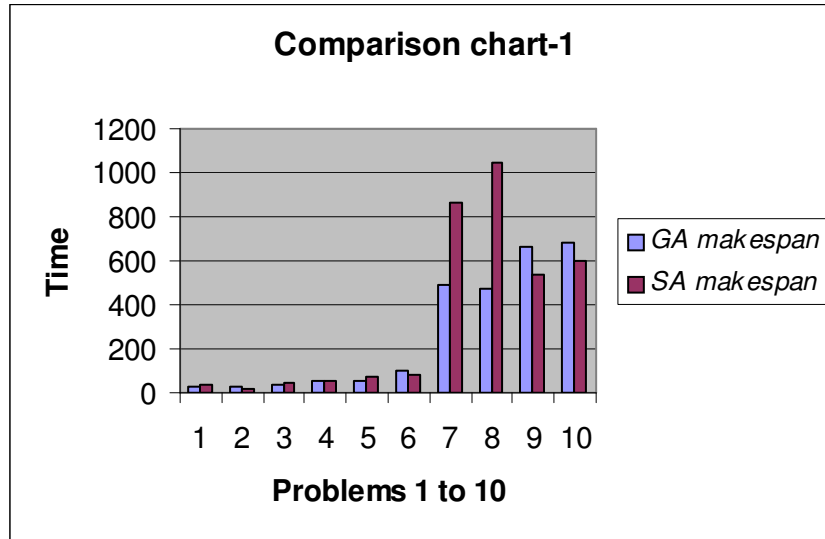
SA Rule Based Representation

The makespan and the mean flow time are determined through the simulated annealing. Considering the various set of rules mentioned above the makespan and the mean flow time are found out. The set of rules are considered as the set of temperatures. An initial temperature is taken or the initial rule is selected randomly the makespan and the mean flow time of the problem given in the data are calculated through the initial rule. Upon the determination of the neighborhood the makespan of the adjacent rule is calculated its compared with the previous makespan the better result is selected. If it doesn't satisfy an increment is made to determine the next immediate neighborhood. Continue the process of getting the better result obtained.

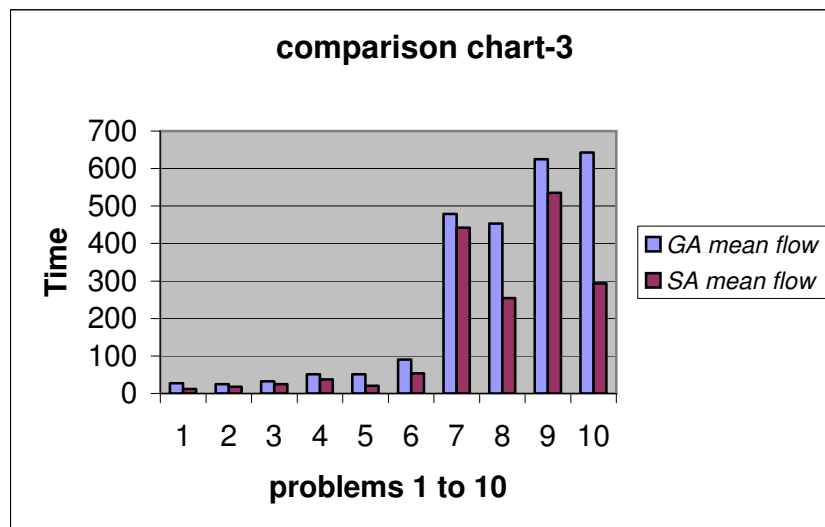## Comparison of Various Objectives Using the Job Based Representation with GA and SA

Pro blem No.&Size	JOB BASED REPRESENTATION	
	Genetic Algorithm	Simulated Annealing

	Makespan	Mean flow time	Makespan	Mean flow time
1(3 X3)	28	27.00	35	11.67
2(4 X4)	29	25.00	18	18.250
3(4 X4)	36	32.25	47	24.67
4(6 X6)	54	50.830	54	37.83
5(6 X6)	55	51.333	75	20.67
6(5 X5)	104	90.600	83	54.00
7(5 X5)	495	479.200	866	442.200
8(7 X7)	477	453.000	1044	254.71
9(8 X8)	668	624.750	535	535.0
10(9X9)	684	642.666	599	293.5

From the above table, it is inferred that under the Job based representation the makespan time is minimum in both the algorithms. From the 10 problems 5 times GA got the minimum makespan time, when compared with Simulated Annealing Algorithm. When the objective of minimizing the mean flow time Simulated Annealing Algorithm performs better in all the cases when compared with Genetic Algorithm.

Comparison chart-1

In chart 1 the comparison of makespan objective using the job based representation with GA and SA.
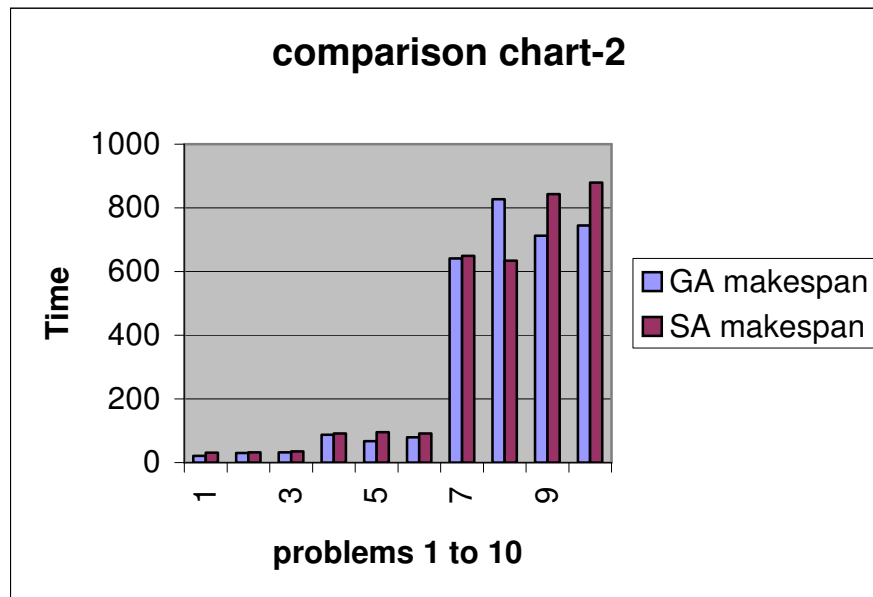

comparison chart-3

In chart 3 the comparison of mean flow objective using the job based representation with GA and SA.

Comparison of Various Objectives Using the Rule Based Representation with GA and SA

Problem No.&Size	RULE BASED REPRESENTATION			
	Genetic Algorithm		Simulated Annealing	
	Makespan	Mean flow time	Makespan	Mean flow time
1(3X3)	21	31.23	31	25.0
2(4X4)	30	23.30	32	17.75
3(4X4)	32	30.85	35	24.67
4(6X6)	87	28.89	91	27.67
5(6X6)	67	43.22	95	29.0
6(5X5)	79	89.0	91	81.80

7(5X5)	641	621	649	599
8(7X7)	827	604.55	634	567.33
9(8X8)	713	685.20	843	654.7
10(9X9)	745	782.7	879	765.6

From the above table, it is inferred that under the Rule based representation the makespan time is minimum in the most of the cases under the algorithm of Genetic Algorithm. From the 10 problems 9 times it got the minimum makespan time, when compared with Simulated Annealing Algorithm. When the objective of minimizing the mean flow time Simulated Annealing Algorithm performs better in all the cases when compared with Genetic Algorithm.

comparison chart-2

In chart 2 the comparison of makespan objective using the Rule based representation with GA and SA.

From the above chart, it is absolutely clear that the Genetic Algorithm is better than the Simulated Annealing Algorithm for makespan time under rule based representation.
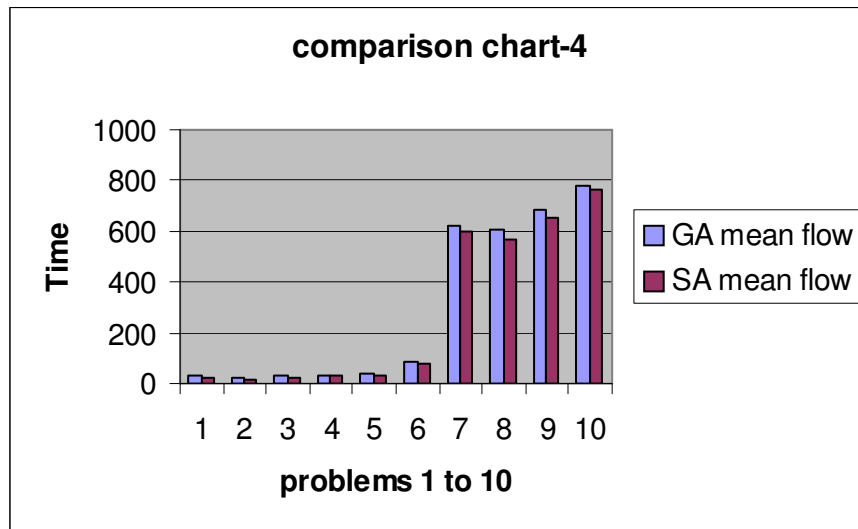
comparison chart-4

In chart 4 the comparison of mean flow objective using the Rule based representation with GA and SA.

From the above chart, it is absolutely clear that the Simulated Annealing Algorithm is better than the Genetic Algorithm for mean flow time under rule based representation.

Conclusion

The over all result shows that the simulated annealing provides better results in the determination of mean flow time in both the representations. The genetic algorithm and the simulated annealing give equal results in the makespan time determination in the job based representation. In the rule-based representation, the genetic algorithm gives better results in the determination of the makespan time than the simulated annealing. All the results of the simulated annealing algorithm seem to be better than the genetic algorithm for the objective mean flow time. Genetic algorithms seem to be better than the simulated annealing algorithm for the objective makespan time.

Acknowledgment

Assumptions

The assumptions made in the job shop are as follows:

1. A Machine can process only one job at a time.

2. An operation cannot be interrupted, implying no job pre-emption.

3. A job has at most "m" operations where "m" is the number of machines.

4. The processing order of a job is given.

5. The processing times are deterministic and include all times such as set time and move times.

6. There are no any machine breakdowns. The machines are continuously available.

7. Set up times are sequence independent and are usually included in the processing times.

8. There are no any manpower constraints, no material constraints and no any tool constraints. The only limitation is that the machine availability.

References

Albrecht, A., et al., (2002) Adaptive simulated annealing for CT image classification, *International journal of pattern recognition and artificial intelligence*, 16, 5, pp: 573-588.

Albrecht, A., et al., (2001) Logarithmic simulated annealing for X-ray diagnosis, *Artificial intelligence in medicine*, 22,3, pp: 249-260.

Aydin, M.E., Fogarty, T.C., (2004) A distributed evolutionary simulated annealing for combinatorial optimization problems, *Journal of heuristics*, 10,3, pp: 269-292.

Aydin, M.E., Fogarty, T.C., (2004) A simulated annealing algorithm for multi-agents systems: A job shop scheduling application, *Journal of intelligent manufacturing*, 15,6, pp: 805-814.

Dirk Mattfeld, C., Christian Bierwirth (2004) An efficient genetic algorithm for job shop scheduling with tardiness objectives, *European journal of operational research*, 155, pp: 616-630.

Esin Onbasoglu., Linet., Ozdamar., (2001) Parallel simulated annealing algorithms in global optimization, *Journal of global optimization*, 19,1, pp: 27-50.

Esquivel et al., (2002) Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem, *Knowledge-Based systems,* 15,1-2, pp: 13-25.

Giffler, B., Thompson, G.L.,(1960), Algorithms for solving production scheduling problems, *International Journal of Operations Research* , No. 8: pp 487-503.

Goncalves, J.F., Mendes, J.J.M., Resende, M.G.C., (2005) A hybrid genetic algorithm for the job shop scheduling problem, *European journal of operational research*, 167, pp: 77-95.

Holthaus, O., Rajendran, C., (2000) Efficient jobshop dispatching rules: further developments, *Production planning and control*, 11,2, pp: 171-178.

Ivan T. Tanev., Takashi Uozumi., Yoshiharu Morotome., (2004) Hybrid evolutionary algorithm based real world flexible job shop scheduling problem: application service provider approach, *Applied soft computing*, 5,1, pp: 87-100.

Jayamohan, M.S., Rajendran, C., (2004) Development and analysis of cost-based dispatching rules for job shop scheduling, *European journal of operational research*, 157, 2, pp: 307-321.

Kenneth R. Baker.,(1974), Introduction to sequencing and scheduling, John Wiley & Sons, Inc.: New York.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., (1983) Optimization by simulated annealing, *Science*, 220, pp: 671-680.

Kolonko, M., (1999) Some new results on simulated annealing applied to job shop scheduling problem, *European journal of operational research*, 113, pp: 123-136.

Kopfer, H., Mattfield, D.C., (1997) A hybrid search algorithm for the job shop problem, Proceedings of the first international conference on operations and quantitative management, jaipur, India, pp: 498-505.

Ponnambalam, S.G., Jawahar, N., Arvindan, P. (1999), A Simulated annealing algorithm for job shop scheduling, *Production Planning and Control* , 10 : pp. 767-777.

Purushothama, G.K., Jenkins, L., (2003) Simulated annealing with local search a hybrid algorithm for unit commitment, *IEEE transactions on power systems*, 18,1, pp: 273-278.

Saber, A.Y., et al., (2006) Fuzzy unit commitment scheduling using absolutely stochastic simulated annealing, *IEEE transactions on Power systems*, 21,2, pp:955-964.

Senthilkumar, P., Shahabudeen, P., (2006) GA based heuristic for the open job shop scheduling problem, *The international journal of advanced manufacturing technology*, 30,3-4, pp:297-301.

Tsung-che Chiang ., Li-Chen-Fu (2007) Using dispatching rules for job shop scheduling with due-date-based objectives, *International journal of production research*, 45, 14, pp: 3245-3262.

Viswanath Kumar Ganesan., Appa Iyer Sivakumar., (2006) Scheduling in static job shops for minimizing mean flow time subject to minimum total deviation of job completion times, *International journal of production economics*, 103,2, pp: 633-647.

Wang, L., Zheng, D.Z., (2001) An effective hybrid optimization strategy for job shop scheduling problems, *Computers and operations research*, 28, pp: 585-596.

Yasunari Yoshitomi., (2002) A genetic algorithm approach to solving stochastic job shop scheduling problems, *International transactions in operational research*, 9,4, pp:479-495.

Yu-Su Shum., Dah-Chuan Gong., (2007) The application of genetic algorithm in the development of preventive maintenance analytic model, *The international journal of advanced manufacturing technology,* 32,1-2, pp: 169-183.