Research Journal of Applied Sciences, Engineering and Technology 9(9): 770-777, 2015
DOI:10.19026/rjaset.9.2623
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: October 10, 2014 Accepted: November 3, 2014 Published: March 25, 2015

**Research Article**

# A Hybrid Evolutionary Algorithm for Discrete Optimization

J. Bhuvana
Department of Computer Science and Engineering, SSN College of Engineering, Chennai-603110, Tamilnadu, India

**Abstract:** Most of the real world multi-objective problems demand us to choose one Pareto optimal solution out of a finite set of choices. Flexible job shop scheduling problem is one such problem whose solutions are required to be selected from a discrete solution space. In this study we have designed a hybrid genetic algorithm to solve this scheduling problem. Hybrid genetic algorithms combine both the aspects of the search, exploration and exploitation of the search space. Proposed algorithm, Hybrid GA with Discrete Local Search, performs global search through the GA and exploits the locality through discrete local search. Proposed hybrid algorithm not only has the ability to generate Pareto optimal solutions and also identifies them with less computation. Five different benchmark test instances are used to evaluate the performance of the proposed algorithm. Results observed shown that the proposed algorithm has produced the known Pareto optimal solutions through exploration and exploitation of the search space with less number of functional evaluations.

**Keywords:** Discrete local search, flexible job shop scheduling, genetic algorithm, memetic algorithm, multi-objective optimization, Pareto optimal

## INTRODUCTION

Optimization problem is characterized by one or more functional objectives and is associated with a solution space. The idea is to identify Pareto optimal solutions from the solution space by either maximizing or minimizing the given objectives. These problems are classified as continuous and discrete with respect to the type of information available in the solution space as continuous real values or discrete finite values. Complex continuous optimization problems can be solved by computing the derivatives whereas the discrete optimizations require choosing an optimal value from fixed set of solutions (Neumann and Witt, 2010). Discrete optimization problems are computationally hard to solve and are mostly NP-hard problems. Applications like project scheduling, production scheduling, man power planning, vehicle outing, telecommunication routing, production facilities design, job Scheduling, space shuttle scheduling, database query design are few of such discrete combinatorial optimization problems (Du and Panos, 1999; Yu, 1998).

Flexible Job Shop Scheduling Problem (FJSSP) is a popular combinatorial (discrete) optimization problem (Jain and Meeran, 1999). This problem is characterized with *n* jobs and *m* machines, where each job will have several levels of operations. Only when all the operations of a job are completed, the job is said to be completed. Operations of one job can be assigned to any machine and the machine has the capability to complete it, hence the problem is named as flexible job shop scheduling. The FJSSP has to deal with the operation sequencing and machine assignment to complete the jobs.

**Problem formulation:** The multi-objective FJSSP is formulated as follows (Chiang and Lin, 2013; Ho *et al*., 2007):

- There are $n$ jobs and $m$ machines.
- Each job $j$, will have $n_j$ operations, where $0 \leq j \leq n$, in a predetermined sequence.
- Operation $O_{ji}$ of job $j$ has to be processed by the machine $m_{ji}$ from the set of machines $M_{ji}$.
- A priori known constant processing time of $O_{ji}$ in machine $k \in M_{ji}$ is $P_{jik}$ and $C_{ij}$ is its completion time. An operation $O_{ji}$ is taken for processing only when its preceding operation $O_{(j-1)i}$ of same job is completed, where $0 \leq i \leq n_j$.
- Each machine can process one operation at a time without interruption.
- All jobs are ready at time 0 and all machines are continuously available.

The three minimizing objectives considered in this work are makespan, total work load and maximum work load. Makespan is the time taken to complete all jobs, total work load is the total time taken by all the machines to complete the jobs and maximum work load is the machine with the largest work load. The set of operations processed by machine $k$ is given by $O_k$ where $1 \le k \le m$, $O_k = \{o_{ji} | m_{ji} = k, 1 \le j \le n, 1 \le i \le n_j\}$. The functional objectives are given by:

$$Makespan\ (C_M): f_1 = \max_{\{j=1\ldots n\}} C_{jn_j}$$

$$Total\ work\ load\ (W_T): f_2 = \sum_{k=1}^{m} \sum_{O_{ji} \in O_k} P_{jik}$$

$$Maximum\ work\ load\ (W_M): f_3 = \max_{\{k=1\ldots m\}} \sum_{O_{ji} \in O_k} P_{jik}$$

The aim of this study is to generate Pareto optimal schedules that minimize the above three objectives.

**Memetic/hybrid GA:** FJSSP is a multi-objective optimization problem with more than one conflicting objectives. When such problems are solved there will not be a single unique solution instead, a set of optimal solutions that are superior to other solutions in the search space is required. Evolutionary Algorithm (EA) is a stochastic optimization approach that works with a population of individuals to optimize multiple objectives simultaneously. Genetic Algorithm (GA) is a kind of evolutionary algorithm that is inspired by natural selection. EA or GA in specific is designed to explore the search space globally and said to lack in fine tuning of individuals that is, in exploiting the search space (Krasnogor and Smith, 2005). To overcome this known drawback of EA, it can be combined with other optimization techniques to enhance the exploitation of the search space. A hybrid or Memetic algorithm (Moscato, 1989) is the combination of a global and a local optimization technique. Several hybrid algorithms have been designed and are used to solve continuous multi objective optimization problems. In this study we have designed a hybrid algorithm for a discrete problem like FJSSP, where the hybrid approach combines a global search and local search technique together. The objective of this hybrid algorithm is to optimize the three objectives of FJSSP and to generate a schedule of *n* jobs in *m* machines in an optimal manner.

## LITERATURE REVIEW

**Discrete optimization techniques:** Branch and bound, rounding-off, penalty function approach, cutting plane, simulated annealing, genetic algorithms, neural networks, Langragian relaxation methods are some of the discrete optimization algorithms (Mahfouz, 1999). Algorithms like GA based approaches can also work with continuous problems and are hence considered to be adaptable. We have used GA in the proposed work because of its versatility and tendency to get integrated with other optimization techniques.

**Review on approaches used to solve multi objective FJSSP:** Kacem *et al*. (2002a) has applied two approaches, approach by localization, that designs an ideal assignment model for jobs and controlled evolutionary approach that work upon the assignment made already. Kacem *et al*. (2002b) has proposed a hybrid method that combines EA with fuzzy logic to solve FJSSP. This hybrid approach was designed to work in two stages, a fuzzy multi-objective evaluation stage and an evolutionary multi-objective optimization stage. Several test instances were used to evaluate his approach. Li-Ning *et al*. (2010) has adopted a two level strategy to solve FJSSP, knowledge model and heuristic model. Heuristic model performs the global search using Ant colony optimization, whereas the knowledge model learns from optimization and use them to guide the current searching.

Ho and Tay (2008) have combined GA with guided local search to schedule the optimization problem. A branch and bound has also been used to find the lower bounds of this multi-objective problem. Li *et al*. (2011) has proposed a hybrid Pareto based Artificial Bee Colony algorithm, where a food source represents both the routing and scheduling component of the schedule. Local search was applied both on the routing component and scheduling component of the food source. Local search is performed in this algorithm during the employed bee phase. Wang *et al*. (2010) introduced an improved GA based immune and entropy approach to solve multi-objective FJSSP, where two principles immune, entropy are used to generate diverse set of solutions and designed to avoid premature convergence. Antibodies are considered as the individuals and antigens are the objectives to be improved. Li *et al*. (2010) has designed a new hybrid search algorithm HTSA, which performs a Tabu search to explore neighbouring individuals on the machine assignment module. Variable neighbourhood search is also applied to perform local search on operation scheduling component.

## MATERIALS AND METHODS

**Proposed algorithm: Hybrid GA with DLS for FJSSP:** A new hybrid Genetic algorithm has been devised to handle the discrete optimization problem, FJSS. This hybrid approach incorporates both global and local search together in-order to improve the quality of the individual. Aim of this algorithm is to schedule *n* jobs and *m* machines in a flexible manner so that the makespan, total work load and maximum work load are minimized in lesser number of functional evaluations. Hybrid GA for FJSSP is given in Algorithm 1. Proposed algorithm has been adapted from NSGA-II.

**Initialization:** Population is initialized with individual chromosomes. In this study 3-tuple encoding method

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (004) | (202) | (100) | (114) | (011) | (300) | (311) | (211) |

Fig. 1: Three-tuple chromosome encoding

has been used where both the routing and sequencing data are encoded in the same gene. A gene has 3-tuple in it as (j i k) denoting the operation $O_{ji}$ is assigned to the machine k. Operations of jobs are assigned to machines in the order it has been encoded in the chromosome. For instance in the sample chromosome in Fig. 1. There are 4 jobs and are assigned to 5 machines, where tuple (0 0 4) is the first operation to be processed that is, $0^{th}$ operation of job 0 is assigned to machine 4. ($0^{th}$) operation of job 2 is assigned to machine 2 and so on. A chromosome encodes all the operations of all the jobs to be completed.

**Re-sequencing:** After encoding, the individuals are decoded by Giffler Thomson method. This method avoids unnecessary delays by re-ordering the genes of the chromosomes. After GT, re-ordered chromosomes are considered to be active schedules once the GT re-ordering is over, the chromosomes are re-sequenced using two prominent methods and Most Operation Remaining (MOR) and Most Processing time Remaining (MPR). MOR arranges the operations in the chromosome with respect to number of pending operations of the same job, whereas the MPR, places the operations with respect to the amount of remaining processing time required by the job to be completed. The proposed algorithm used 40% of population to follow the MOR, 40% of MPR and remaining followed the random dispatching of jobs (Pezzella *et al*., 2008).

Evaluation step computes the three functional objectives, makespan, total work load and maximum work load. Non dominated sorting is applied to arrange the individuals in non-domination order and assign fronts for each individual in the population. To generate wide spread Pareto solutions, individuals from less crowded region of search space are given significance during the evolution, for which the crowding distance is calculated. This gives the distance of an individual from its neighbouring individuals. Binary tournament selection is applied to select the parents for crossover operation. For the crossover, three different crossovers are applied in the proposed algorithm. They are Simulated Binary Crossover (SBX), Assignment Crossover (ASX) and Precedence Preserving Order based Crossover (POX). Randomly any one of the three crossovers is applied on the selected parent chromosomes to generate a new offspring. Polynomial mutation is used if SBX is the crossover operator otherwise Precedence Preserving Shift (PPS) mutation is applied on the chromosome.

**Discrete local search:** This search is designed to improve the solution quality yielded by global search in less number of functional evaluations. It starts with an individual and examines the individuals at its locality and picks the best in it. The locality or the neighbourhood space in this problem is defined by the number of machines.

**Neighbourhood individual selection:** An operation *i* of job *j* which is assigned to machine *k*, is taken as the starting point, its neighbourhood will be $k^* = \{k^* \neq k | 1, 2 ... m\}$, where *m* is the number of machines. Criterion that is used to choose the next step is the processing time. The neighbour individual is selected if its processing time is less when compared with others. In real time, it is equivalent to releasing the operation *i* of job *j* from machine *k* to another machine *k** which can quickly finish this operation. A reassignment of one operation of one job to a new machine in discrete space creates a new offspring. There are $N_{OP}$, number of operations encoded in a single chromosome and hence $N_{OP}$ number of new offspring can be created through this DLS. By this reassignment scheme the discrete space is exploited in a finer manner to obtain the best locally optimal solution.

**Individual evaluation in LS:** To evaluate the new offspring that is generated, it has to be evaluated and need to be compared with its previous point in the local search. Three functional objectives of this new offspring say, makespan, total work load and maximum work load are computed. We have used the adaptive weighted sum approach to combine these three functional objectives into single objective, (Bhuvana and Aravindan, 2011). Weights are associated with the objectives to characterize the significance of each objective. Higher the weight greater significance will be given to that objective. This adaptive weight strategy collects the knowledge about the significance of objectives from the objective space itself. Single objective function is constructed using this adaptive weights method and the individual is evaluated. This new single functional objective is compared with the single objective of the starting or previous point in the local search space. If new individual is better than its parent individual, it will be allowed to replace it.

**Depth of DLS:** To achieve a balanced exploration and exploitation, we decided to depth limit the local search. The depth of DLS is limited to single step, so that the time spent in local search will not add burden to the overall search process. DLS is also designed to stop investigating its locality whenever it finds the first local minimum. This has the advantage in reducing the number of functional evaluations incurred in visiting the neighbourhood individuals in its locality. Working of DLS is presented in Algorithm 3.

**Algorithm 1:** Hybrid algorithm for flexible job shop schedule problem

procedure $MA_{DLS}$

Input: number of jobs, number of machines, number of operations per job, processing time for each operation
Output: schedule of jobs
Encode each individual in population of size, PopSize.
Call Re-sequenceing-Individual() on initial population
Compute fitness of each individual.
while Stopping criteria not met do
    Apply non dominated sorting and compute front
    Compute crowding distance
    Apply binary tournament selection
    Apply crossover on selected parents using one of following;
    SBX (); ASX (); POX ()
    Apply Polynomial mutation if crossover is by SBX () otherwise apply PPS mutation
    for m←1, PopSize do
    Apply discrete-local-search ($Offspring_m$)
    end for
    Add the offspring to the population
  end while
end procedure

**Algorithm 2:** Algorithm to modify the encoded sequence
Procedure RE-SEQUENING-INDIVIDUAL ()
  Input: Number of jobs, number of machines, number of operations per job, processing time for each operation
  Output: Re-sequenced initial schedule
  Apply Giffler Thomson algorithm
  Apply Most Operations Remaining (MOR);
  Apply Most Processing time Remaining (MPR);
end procedure

**Algorithm 3:** Algorithm to perform Discrete Local Search (DLS)
procedure DISCRETE-LOCAL-SEARCH ($O_m$)
  Input: Offspring after crossover, $O_m$.
  Output: New offspring after local search
  Compute adaptive weights for the objectives of $O_m$.
  Construct signal objective $F_{Om}$ using adaptive weights.
  for $n$←1. $N_{op}$ do ⊳ $N_{op}$, number of operations in chromosome.
    Redirect operation $i$ of jobs $j$ of gene n from machine $k$, to other machine $k$*
      where $k^* = \{\text{argmin } (P_{ijk}) | k^* \neq k\}$, $0 \leq j \leq n$; $0 \leq k \leq m$;
      ⊳ $n$ is the number of jobs, $m$ is number of machines.
  Generate a new offspring $O_{new}$ by inserting the new gene in place of actual.
  Evaluate the new offspring $O_{new}$.
  Compute adaptive weights and new single objective $F_{new}$, for $offspring_{new}$.
  if $F_{new}$ is better than $Fo_m$ then
    Replace actual individual, $O_m$ by one step locally optimized new individual, $O_{new}$.
      return ($O_{new}$).
    end if
  end for
end procedure

Once the local search on all offspring is over, the new offspring set is merged with the parent population and the steps are repeated until the termination condition is met. Working of Hybrid GA with DLS for FJSSP is given in Algorithm 1.

**Materials:** Test cases to evaluate the performance of the proposed hybrid GA with DLS are taken from Kacem *et al*. (2002a, b). There are totally five test cases, 4×5, 8×8, 10×7, 10×10 and 15×10. Test data has number of jobs, number of machines, number of operations per job and processing time for each operation in each machine. Pareto optimal solutions for these test cases are available in http://web.ntnu.edu.tw/~tcchiang/publications/IJPE2013-MOFJSP.txt.

## RESULTS AND DISCUSSION

**Experiments conducted:** The Hybrid GA with DLS for FJSSP was implemented in C programming language in Intel (R) Core (TM) i5-3470 CPU @3.20 GHZ system. Population size is set to 50 and the hybrid GA with DLS was made to run for 30 iterations with single step local search on the offspring. Crossover probability is 1 and Mutation probability is set to (1/Number of Operations). We have compared our results with that of other works; MOEA-GLS by Ho *et al*., (2007), PHMOEA by Nagamani *et al*. (2013), HTSA by Li *et al*. (2010), P-DABC by Li *et al*. (2011), AdRep by Unachak and Goodman (2010), MOGA by Wang *et al*. (2010).

Totally five test instances are taken to evaluate the performance of the hybrid GA with DLS. The performance criteria we have taken for evaluation is the number of new individuals evaluated totally. Since the DLS is designed to stop whenever it first meets the local minimum, the upper bound for this criterion is the product of population size, number of operations and number of iterations. Table 1 to 5 present the functional objectives obtained by the proposed hybrid GA with DLS and compared with the already known works in literature.

**Case I: 4×5:** The first test instance taken from Kacem *et al*. (2002a, b) has 4 jobs with 12 operations to be scheduled in 4 machines. Proposed hybrid GA with DLS was able to identify all the four known Pareto

Table 1: Pareto optimal solutions for 4×5

| Problem instance 4×5 | Number of new individuals evaluated | Optimal solutions obtained | | |
|---|---|---|---|---|
| | | Makespan ($C_M$) | Total work load ($W_T$) | Maximum workload ($W_M$) |
| MOEA-GLS | 40000 | 16 | 32 | 8 |
| | | 16 | 33 | 7 |
| PHMOEA | 20000 | 13 | 33 | 7 |
| | | 12 | 32 | 8 |
| P-DABC | 90000 | 11 | 32 | 10 |
| | | 12 | 32 | 8 |
| | | 13 | 33 | 7 |
| MOGA | 40000 | 11 | 32 | 10 |
| | | 11 | 34 | 9 |
| | | 12 | 32 | 8 |
| HTSA | 24000 | 11 | 32 | 10 |
| | | 12 | 32 | 8 |
| Proposed | 18000 | 11 | 32 | 10 |
| | | 12 | 32 | 8 |
| | | 13 | 33 | 7 |
| | | 11 | 34 | 9 |

Table 2: Pareto optimal solutions for 8×8

| Problem instance 8×8 | Number of new individuals evaluated | Optimal solutions obtained | | |
|---|---|---|---|---|
| | | Makespan ($C_M$) | Total work load ($W_T$) | Maximum workload ($W_M$) |
| MOEA-GLS | 40000 | 11 | 73 | 10 |
| PHMOEA | 50000 | 14 | 78 | 11 |
| | | 14 | 73 | 12 |
| P-DABC | 51200 | 14 | 77 | 12 |
| | | 15 | 75 | 12 |
| | | 16 | 73 | 13 |
| AdRep | 40000 | 14 | 77 | 12 |
| | | 15 | 75 | 12 |
| | | 16 | 73 | 13 |
| | | 16 | 77 | 11 |
| MOGA | 40000 | 15 | 81 | 11 |
| | | 15 | 75 | 12 |
| | | 16 | 73 | 13 |
| Proposed | 30258 | 14 | 77 | 12 |
| | | 15 | 75 | 12 |

Table 3: Pareto optimal solutions for 10×7

| Problem instance 10×7 | Number of new individuals evaluated | Optimal solutions obtained | | |
|---|---|---|---|---|
| | | Makespan ($C_M$) | Total work load ($W_T$) | Maximum workload ($W_M$) |
| MOEA-GLS | 40000 | 15 | 61 | 11 |
| | | 16 | 60 | 12 |
| | | 15 | 62 | 10 |
| P-DABC | 59500 | 12 | 11 | 12 |
| | | 61 | 63 | 60 |
| | | 11 | 11 | 12 |
| HTSA | 196000 | 11 | 61 | 11 |
| | | 11 | 62 | 10 |
| Proposed | 35505 | 11 | 62 | 10 |
| | | 11 | 61 | 11 |

optimal solutions reported earlier in literature so far. These Pareto optimal solutions are obtained in lesser number of functional evaluations when compared with the rest of the algorithms compared. Lesser the number of functional evaluations faster will be the response of the algorithm.

**Case II: 8×8:** The second test instance taken has 8 jobs with 27 operations to be scheduled in 8 machines. The obtained Pareto optimal solutions are listed in Table 2. Gantt chart for one Pareto optimal solution is shown in Fig. 2.

**Case III: 10×7:** The third test instance has 10 jobs with 29 operations to be scheduled in 7 machines. Our proposed algorithm, Hybrid GA with DLS has generated two schedules whose makespan is 11 with a total work load of 62, 61 and maximum work load as 10 and 11.

**Case IV: 10×10:** The next test case taken has 10 jobs and 10 machines with overall 30 operations to be scheduled. Our Proposed algorithm was able generate three optimal schedules. Functional objectives are listed in Table 4 for this test instance.

Table 4: Pareto optimal solutions for 10×10

| | | Optimal solutions obtained | | |
|---|---|---|---|---|
| Problem instance 10×10 | Number of new individuals evaluated | Makespan ($C_M$) | Total work load ($W_T$) | Maximum workload ($W_M$) |
| MOEA-GLS | 40000 | 8 | 41 | 7 |
| | | 8 | 42 | 5 |
| | | 7 | 43 | 5 |
| | | 7 | 42 | 6 |
| PHMOEA | 250000 | 7 | 43 | 5 |
| | | 8 | 42 | 5 |
| P-DABC | 100000 | 8 | 41 | 7 |
| | | 7 | 43 | 5 |
| | | 8 | 42 | 5 |
| AdRep | 3000000 | 7 | 42 | 6 |
| | | 7 | 45 | 5 |
| | | 8 | 41 | 7 |
| | | 8 | 42 | 5 |
| MOGA | 40000 | 8 | 42 | 5 |
| | | 8 | 41 | 7 |
| | | 7 | 42 | 6 |
| | | 7 | 45 | 5 |
| HTSA | 300000 | 7 | 42 | 6 |
| | | 7 | 43 | 5 |
| | | 8 | 42 | 5 |
| Proposed | 31307 | 8 | 41 | 7 |
| | | 8 | 43 | 5 |
| | | 7 | 42 | 6 |

Table 5: Pareto optimal solutions for 15×10

| | | Optimal solutions obtained | | |
|---|---|---|---|---|
| Problem instance 15×10 | Number of new individuals evaluated | Makespan ($C_M$) | Total work load ($W_T$) | Maximum workload ($W_M$) |
| MOEA-GLS | 40000 | 11 | 91 | 11 |
| | | 11 | 93 | 10 |
| PHMOEA | 300000 | 11 | 91 | 11 |
| | | 11 | 93 | 10 |
| AdRep | 5600000 | 11 | 91 | 11 |
| | | 11 | 93 | 10 |
| MOGA | 40000 | 11 | 91 | 11 |
| | | 11 | 98 | 10 |
| | | 12 | 95 | 10 |
| HTSA | 810000 | 11 | 91 | 11 |
| | | 11 | 93 | 10 |
| Proposed | 84000 | 11 | 91 | 11 |
| | | 12 | 95 | 11 |

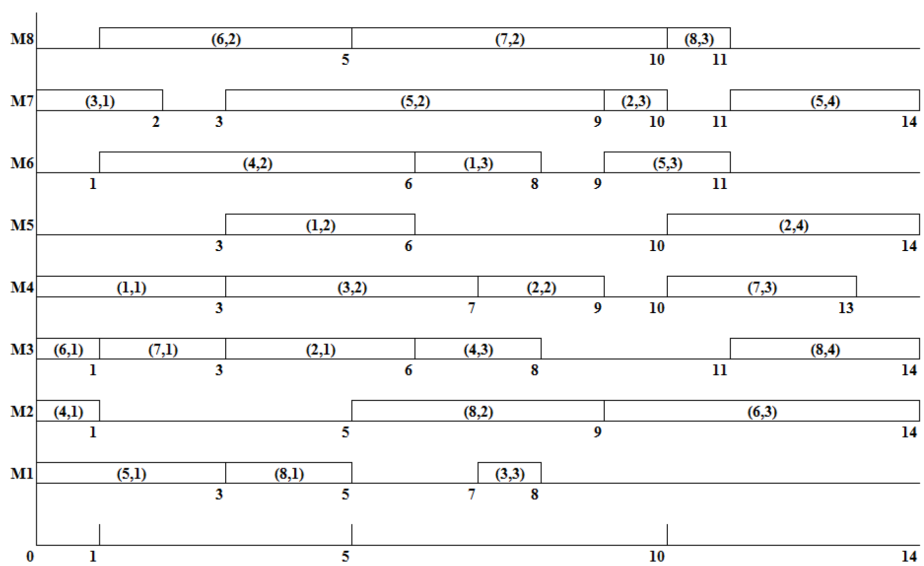


Fig. 2: Gantt chart for the test instance 8×8

**Case V: 15×10:** This test instance is larger than any other instance taken here and has 15 jobs with a total of 56 operations to be scheduled in 10 machines. Our hybrid GA with DLS for FJSSP was able to get optimal makespan of 11 and total work load 91 and maximum work load as 11.

## CONCLUSION

In this study, we have introduced a Hybrid GA with Discrete Local Search (DLS) for Flexible Job Shop Scheduling Problem. The objective of the study is to design a local search that exploits the discrete solution space with less number of functional evaluations. We have designed a new problem specific Discrete Local search which is combined with the genetic algorithm that performs the global search. By this integration, the solution space is explored and exploited in the balanced manner.

We have compared the proposed algorithm with the other known works in the recent literature. Performance criterion taken by this study is the number of new individuals entered the population after evaluation. By this criterion, the results clearly show that our proposed algorithm has the ability to generate the optimal schedules in less number of functional evaluations. In this study the new Discrete Local search is combined with a Genetic algorithm, but DLS can be integrated with any other global optimization algorithm like ACO, PSO, ABC and so on. This study can be further extended by adding explicit features to reduce the burden of overloaded machines, etc., in DLS and its performance can be studied.

## ACKNOWLEDGMENT

The researcher would like to thank the management of SSN College of Engineering for funding the High Performance Computing Lab (HPC Lab) where this research was carried out. The author expresses sincere thanks to Dr. Chandrabose Aravindan, Professor, Department of CSE, SSN College of Engineering for his constructive suggestions and valuable guidance.

## REFERENCES

Bhuvana, J. and C. Aravindan, 2011. Preferential local search with adaptive weights in evolutionary algorithms for multiobjective optimization problems. Proceeding of the International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp: 358-363.

Chiang, T.C. and H.J. Lin, 2013. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling. Int. J. Prod. Econ., 141(1): 87-98.

Du, D.Z. and M.P. Panos, 1999. Handbook of Combinatorial Optimization: Supplement. Vol. 1, Springer, Dordrecht.

Ho, N.B. and J.C. Tay, 2008. Solving multiple-objective flexible job shop problems by evolution and local search. IEEE T. Syst. Man Cy. C, 38(5): 674-685.

Ho, N.B., J.C. Tay and E.M.K. Lai, 2007. An effective architecture for learning and evolving flexible job-shop schedules. Eur. J. Oper. Res., 179(2): 316-333.

Jain, A.S. and S. Meeran, 1999. Deterministic job-shop scheduling: Past, present and future. Eur. J. Oper. Res., 113(2): 390-434.

Kacem, I., S. Hammadi and P. Borne, 2002a. Approach by localization and multi- objective evolutionary optimization for flexible job-shop scheduling problem. IEEE T. Syst. Man Cy. C, 32: 1-13.

Kacem, I., S. Hammadi and P. Borne, 2002b. Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic. Math. Comput. Simulat., 60: 245-276.

Krasnogor, N. and J. Smith, 2005. A tutorial for competent memetic algorithms: model, taxonomy and design issues. IEEE T. Evolut. Comput., 9(5): 474-488.

Li, J.Q., Q.K. Pan and Y.C. Liang, 2010. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. Comput. Ind. Eng., 59(4): 647-662.

Li, J.Q., Q.K. Pan and K.Z. Gao, 2011. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems. Int. J. Adv. Manuf. Tech., 55(9-12): 1159-1169.

Li-Ning, X., C. Ying-Wu, W. Peng, Z. Qing-Song and X. Jian, 2010. A knowledge-based ant colony optimization for flexible job shop scheduling problems. Appl. Soft Comput., 10(3): 888-896.

Mahfouz, S.Y., 1999. Design optimization of structural steelwork. Ph.D. Thesis, University of Bradford.

Moscato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech Concurrent Computation Program, C3P Report 826.

Nagamani, M., E. Chandrasekaran and D. Saravanan, 2013. Pareto-based hybrid multi-objective evolutionary algorithm for flexible job-shop scheduling problem. IOSR J. Math., 9(1): 36-45.

Neumann, F. and C. Witt, 2010. Combinatorial Optimization and Computational Complexity. Bioinspired Computation in Combinatorial Optimization. Springer-Verlag, Berlin, Heidelberg, 2010: 9-19.

Pezzella, F., G. Morganti and G. Ciaschetti, 2008. A genetic algorithm for the flexible job-shop scheduling problem. Comput. Oper. Res., 35: 3202-3212.

Unachak, P. and E. Goodman, 2010. Solving multiobjective flexible job-shop scheduling using an adaptive representation. Proceeding of the 12th Annual Conference on Genetic and Evolutionary Computation, pp: 737-742.

Wang, X., L. Gao, C. Zhang and X. Shao, 2010. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. Int. J. Adv. Manuf. Tech., 51(5-8): 757-767.

Yu, G., 1998. Industrial Applications of Combinatorial Optimization. Vol. 16, Kluwer Academic Publishers, Dordrecht.