Implementation and Evaluation of Novel Parallel Hybrid Approach for Solving Job Shop Scheduling Problem

Chaitali Anil Kulthe[1], Y. R. Kalshetty[2]

PG Student, CSE Department, SVERI's College of Engineering, Pandharpur, India

Assistant Professor, CSE Department, SVERI's College of Engineering, Pandharpur, India

Abstract: ***Since from last three decades, genetic algorithms (GA) are most popular approach for solving number of optimization research problems. The shop scheduling problem is well known and widely studied problem in which the number of jobs should be processed over the set of available machines so that optimization criteria should be satisfied. To solve the problem of job shop scheduling (JSS) problem, there are number of methods already proposed with goal of improving the efficiency and performance of problem solving. The efficiency of JSS problem solutions is evaluated in terms of three time related performance metrics such as flow average time, waiting time and total execution time. The aim of any JSS problem solution is to minimize the performance of these three metrics. In this paper, we designed novel solution for solving the job shop scheduling problem using genetic algorithm. The proposed solution is based on parallel genetic algorithm in which modified crossover and mutation operations introduced. The processing of genetic algorithm is performed parallel which helps in reduction of time performance while solving any of JSS problem. In this paper we implemented the proposed approach using MATLAB and evaluated the performance on different test cases of JSS problems such as Dmu07, YN01, YN04, LA38, 3x3 and 6x6.***

Keywords: Job shop scheduling, Genetic Algorithm, Mutation, Crossover, Population, Parallel, Waiting Time, Flow Time, Execution Time

1. Introduction

Genetic algorithms are related with the class of the evolutionary algorithms. These are algorithmswhich are based on the characteristic of the natural evolution, & they will be divided intofour major types of algorithms: genetic algorithms (GA), genetic programming language, evolution tactical& the evolutionary programming language. All these types of algorithms are based on apopulation of the separated. Evolutionary algorithms which is have been used to many problems inmanagement, e.g., to the location, inventory as well as production, scheduling and the distribution or timetablingproblems.

The use of the evolutionary algorithms which is for the shop scheduling issues which is started into the 1980.There are two most first applications to flow shop scheduling problems have been given by Werner, & the first of application to the job shop scheduling problems which can be found in [1]. Genetic algorithms are the most popular variant of evolutionary algorithms. The structureand components of elementary genetic algorithms has been discussed e.g. by Goldberg [2] or Beasley et al. [3]. Evolution strategies have been originally developed for optimizationproblems into the engineering. Here one is can defined the pioneering executions by Rechenberg [4]and Schwefel [5].

Although both types of the evolutionary algorithms having many similar features, therealso exist some differences. Evolution strategies normally work directly withthe real-valued vectors; genetic algorithms always use strings of bits. While in genetic algorithmsrecalibration into the form of with the use of crossover operators plays a dominant role, evolutionstrategies mainly use mutation into the form of normal updates of the particular real variables.Evolution strategies also use some type of recombination, always into the form of the discrete recombinationfor the generating the offspring (i.e., it is decided for each component the importantof which is of the two separated parents is used for the offspring) and intermediate recombination todefine the current strategy parameters. When into the genetic algorithms has beenoften the parameters forapplying specific genetic operators are constant, the technique of the parameters into the evolution strategiesmostlyunderlie an adaptation process. Genetic algorithms are particularly used to thecatabolically optimization issues so that in the following, we mainly focus on this classof evolutionary algorithms.

Job shop scheduling problems are computationally complex problems. Because JSS problems are NP-hard -- i.e., they did not be sort them into the polynomial time --brute-force or undirected search methods are not typically feasible, for the issues at listof every size. Thus JSSP's define to be solved using a combination of search and heuristics get critical or the near optimal of this solutions. Around the different search methodologies used forJSS problems, the Genetic Algorithm (GA), it is excited by the process of the Darwinian evolution, hasbeen recognized as the basic manual search strategy & the optimization system which is always oftenvery useful into theattacking combinatorial issues. Since from Davis proposed this first GA-basedmethod for solving scheduling problems in 1985, GAs has been used with increasingfrequency to solve JSS problems. In contrast to the basic search methods like as simulatedannealing & the tabu-search, which are based on manipulating one feasible solution, theGA is using the population of the solutions into the search, giving it more resistance to prematureconvergence on basic minima. The most important critics in applying GAs to the highly constrainand combinatorial optimization problems such as JSS problem is maintaining theexpiry of the solutions. This issues are basically solved by modifying the breedingusers or the giving penalties on the infeasible solutions in the fitness function. Althoughresistant to premature convergence, GAs is not immune. One strategy to decrease thepremature convergence into the GA is parallelization of the GA into the disjoint subpopulations,in which is also the most realistic model of nature than a single population. There are two different kinds of the parallel GAs (PGAs) that which are widely used: coarse-grain Gas(cgGAs) and fine-grain GAs (fgGAs). Both will be studied in the context of JSS problems.Two basic models are identified in literature: fine-grained PGAs (Manderick and Spiessens, 1989) and coarse-grained PGAs (Christou and Meyer, 1996; Lin et al., 1994). In the former model, thechromosomes are spatially arrayed in some manner within the population and can only interactwith chromosomes within their immediate vicinity (or neighbourhood). In the latter, numerous subpopulationsevolve independently, such as standard GAs. At certain intervals, separated migratefrom the one of subpopulation to the other to promote diversity. These parallel genetic algorithm methods are suffered from the limitations of efficiency.

Therefore, in this paper, we proposed the efficient parallel genetic algorithm for solving the JSS problems. The proposed approach is based on coarse-grained paradigm with some refinements and modifications. The initial population of size N isdivided into k equally sized subpopulations or "villages" that evolve independently. When fitnessstagnates in a village, due to the increasing number of members in the$L$subset, the village is "merged"with another one (as described below), resulting in $k-1$ remaining subpopulations (of generallydifferent size). The proposed approach is discussed in detail during next sections of this paper. Section II, presenting the introduction and discussions of JSS problems. Section III presenting the related works and methods for solving the JSS problems. Section IV presenting the details on proposed algorithm called GAPr (GA parallel) and its flowchart. Section V presenting the results and discussions by considering the different JSS problems test cases.

2. Job Shop Scheduling

Sequencing in job shops involves the timing of specific operations and tasks. It is a short-term planning process which has a dramatic impact on: production costs, capacity utilization, meeting customer delivery (promise) dates, work-in-process inventory, shop congestion, etc. A job shop is defined as a functional organization whose departments or work centers are organized around particular types of equipment or operations, such as drilling, forging, spinning, or assembly. Products flow through departments in batches corresponding to individual orders, which may be either stock orders (orders prepared for inventory) or customer orders.Typical characteristics of job shop include products being made to order and customer orders having differing: processing requirements, material requirements, processing times, processing sequences, set-up times and costs. These differing requirements lead to trade-offs among utilization, order delays, inventory levels, process times, set-up costs, etc. In addition, scheduling difficulty is enhanced because processing schedules cannot be determined until actual orders are received. Task scheduling in job shops actually involves seven basic activities, including:

- Due-date setting - a negotiation process with the customer
- Authorization to release orders to the shop floor
- Loading - assigning jobs to work centers, a problem that arises when two or more work centers may process a job
- Routing - assigning jobs to machines within work centers
- Sequencing – priority decisions regarding the order in which jobs in the wait queue will be processed
- Vendor scheduling
- Monitoring performance using tools such as Input/output analysis

A solution to the sequencing problem depends upon the size of the problem: (1) single machine (work center) or one-stage problem, (2) 2-machine (work centers) or two-stage problem, or (3) an m-machine (work centers) problem. Generally heuristic rules will be applied to determine a solution as we cannot use complete enumeration to evaluate all potential solutions. Heuristic rules are used for establishing the priority of jobs in the work center queue. There are numerous heuristics available.

Heuristic Sequencing Rules

1) **First-Come, First-Serve (FCFS):** The job which arrives first enters service first (local rule).
   Advantages: simple, fast, "fair" to the customer.
   Disadvantages: generally, the least effective as measured by traditional performance measures as a long job makes others wait resulting in idle downstream resources and it ignores job due date and work remaining (downstream information).
2) **Shortest Processing Time (SPT): the job which has the smallest operation time, enters service first (local rule).**
   Advantages: simple, fast, generally a superior rule in terms of minimizing completion time through the system, minimizing the average number of jobs in the system, usually lower in-process inventories (less shop congestion) and downstream idle time (higher resource utilization), and usually lower average job tardiness.
   Disadvantages: ignores downstream, due date information, and long jobs wait (high job wait-time variance).
3) **Earliest Due Date (EDD):** the job which has the nearest due date, enters service first (local rule).
   Advantages: simple, fast, generally performs well with regards to due date, but if not, it is because the rule does not consider the job process time.
   Disadvantages: high priority of past due job and it ignores work content remaining.
4) **Critical Ratio (CR) Rule:**sequences jobs by the time remaining until due date divided by the total remaining processing time (global rule). The job with the smallest ratio of due date to processing time enters service first. The ratio is formed as (Due Date-Present Time)/Remaining Shop Time, where remaining shop time refers to: queue, set-up, run, wait, and move times at current and downstream work centers.

Advantages: recognizes job due date and work remaining (incorporates downstream information)
Disadvantage: past due jobs have high priority, does not consider the number of remaining operations

5) **Slack per Operation**: is a global rule, where job priority determined as:(Slack/# of Remaining Operations)
Advantages: recognizes job due date and work remaining (incorporates downstream information)
Disadvantage: past due jobs have high priority
6) **Least Changeover Cost (Next Best rule):**Sequences jobs by set-up cost or time (local rule).
Advantages: simple, fast, generally performs well with regards to set-up costs.
Disadvantages: does not consider the job process time, due date and work remaining.
Additional useful shop floor planning and control tool: Gantt chart, graphical aid useful for loading and scheduling work. Chart developed by Henry Gantt in early 1900's.

3. Related Works

In [6], author Mikkel T. Jensen et.al proposed the Robust and Flexible Job Shop SchedulesUsing Genetic Algorithms. A robustness measure is definedand its properties are investigated. Through experiments, itis shown that using a genetic algorithm it is possible to find robust and flexible schedules with a low makes pan. These schedules aredemonstrated to perform significantly better in rescheduling after a breakdown than ordinary schedules.The rescheduling performance of the schedules generatedby minimizing the robustness measure is compared with the performance of another robust scheduling method taken from literature, and found to outperform this method in many cases**.**

In [7], author S. Hajri et.al introduced the controlled Genetic Algorithm by Fuzzy Logic and Belief Functions for Job-Shop Scheduling. A controlled genetic algorithm (CGA) based on fuzzy logic and belief functions to solve job-shop scheduling problems. For better performance, they propose an efficient representational scheme, heuristic rules for creating the initial population, and a new methodology for mixing and computing genetic operator probabilities.

In [8], author Min Liu et.al proposed an adaptive annealing genetic algorithm for the job-shop planning and scheduling problem. The genetic algorithm, the simulated annealing algorithm and the optimum individual protecting algorithm are based on the order of nature, and there exist some application limitations on global astringency, population precocity and convergence rapidity. An adaptive annealing genetic algorithm is proposed to deal with the job-shop planning and scheduling problem for the single-piece, small-batch, custom production mode.

In [9], author Yang Xiaomei et.al proposed a Genetic Algorithm for Job Shop Scheduling Problem Using Co-evolution and Competition mechanism. This algorithm is based on the mechanism of co-evolution and natural selection. The parentsand the genetic operators are selected by the competitive principle. Therefore, this algorithm can not only denote parallelism in the course of GA, but also develops the solution quality to the Job Shop scheduling problem. The computation results validate the effectiveness of the proposed algorithm.

In [10], author Jianshuang Cui introduced hybrid heuristic neighborhood algorithm for the job shop scheduling problem a hybrid heuristic neighborhood algorithm (HHNA) is proposed for the job shop scheduling problem. The hybriddesign is mainly composed of two iterative phases. Phase one aim at generating new scheduling solutions through the conditioned neighborhood swapping. An evaluationprocedure is designed to judge the feasibility of a new solution. An infeasible solution will be discarded. Phase two is an improved critical path algorithm that is used todo the intensification search. The computational results demonstrate that the algorithm is highly efficiency both in solution quality and time.

In [11], author Jason Chao-Hsien Pan et.al proposed the hybrid genetic algorithm for no wait job shop scheduling problems. A no wait job shop (NWJS) describes a situation where every job has its own processing sequence with the constraint that no waiting time is allowed between operations within any job. A NWJS with objective of minimizing total completion times is a NP hard problem and hybrid genetic algorithm (HGA) to solve this complex problem.

In [12], author Yan-wei Zhao et.al proposed new hybrid parallel algorithm for consistent-sized batch splitting job shop scheduling on alternative machines with forbidden intervals. The batch splitting scheduling problem on alternative machines with forbidden intervals, based on the objective to minimize the makes pan. A scheduling model is established, takingbefore-arrival set-up, processing, and transfer time into account.

In [13], author Omid Gholami et.al proposed approach for solving parallel machines job-shop scheduling problems by an adaptive algorithm.An adaptive algorithm with a learning stage for solving the parallel machines job-shop problem is proposed. A learning stage tends to produce knowledge about a benchmark of priority dispatching rules allowing a scheduler to improve the quality of a schedule which may be useful for a similar scheduling problem. Once trained on solving sample problems (usually with small sizes), the adaptive algorithm is able to solve similar job-shop problems with larger size better than heuristics used as a benchmark at the learning stage. For using an adaptive algorithm with a learning stage, a job-shop problem is modeled via a weighted mixed graph with a conflict resolution strategy used for finding an appropriate schedule.

In [14], author Su Nguyen et.al proposed computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem theuse of genetic programming for automatically discovering newdispatching rules for the single objective job shop schedulingproblem (JSP). Experimental results show that theirpresentation that integrates system and machine attributes can improve the quality of the evolved rules.

Analysis of the evolved rules also provides useful knowledge about how these rules can effectively solve JSP.

In [15], authorHong Li YIN proposed the Genetic Algorithm Nested with Simulated Annealing for Big Job Shop Scheduling Problems. Genetic algorithm has demonstrated considerable success in providing efficient solutions to many non-polynomial-hard optimization problems. A novel method that is genetic algorithm nested with local search procedure. After crossing and mutation operations in every generation of genetic algorithm, a local search operation be carried out form every population individual.

4. Proposed Methodology

Below figure 1 is showing the proposed approach working step by step. As showing the figure, along with the steps of GA, parallel processing is achieving by applying the path relinking and offspring terminologies. Below subsections are discussing the details of each point in figure.

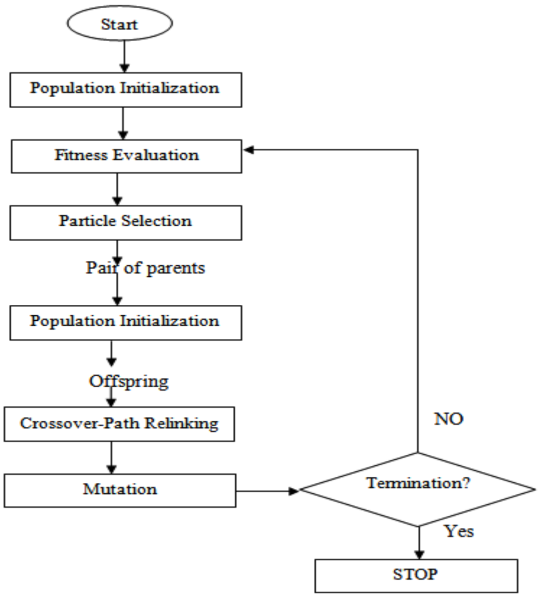

Figure 1: Proposed GAPr Method for Solving JSS Problems

4.1. Crossover

For parallel processing the existing crossover operator is modified. New crossover operator introduced based on the ideas of PR that emerged in the context of intensifying tabu-search methods and found widespread adoption in many other metaheuristic frameworks. PR refers to a combinatorial optimization heuristic technique by which two solutions are labeled as the "source" and the "destination" solution; and an algorithm performs "guided moves" from the source solution so as to reach the destination solution, while in the processconstructing several "intermediate" solutions, assuming that one of them will be of higher quality than either of the two initial solutions. To define the PR crossover operator, the concept of the "Hamming distance" Hd (. . .) between two schedules is required.Clearly, Hd (s1, s2) = Hd (s2, s1) for any pair of feasible schedules s1 and s2.

Let us be a feasible schedule for the JSSP. The notation s−1(z) denotes the position of operation z in s.Also, the notation s(u) denotes the schedule resulting when applying a move u ∈ N1(s) to the schedule s.

Algorithm 1: PR crossover operator (s1, s2)
1. Set h = Hd (s1, s2).
2. Setg = argmins∈ {s1, s2} {Cmax(s)}, i = argmaxs∈ {s1, s2} {Cmax(s)}.
3. Set S = i.
4. Create the move set N+ = {u (x, y) ∈ N1(S): g−1(y) < g−1(x) ∧Hd(S(u), g) = Hd (S, g) − 1}
5. Select a move u ∈ N+ if N+ not empty, otherwise select move u in N1
6. Set S = S(u).
7. Repeat Steps 4–6 until Hd (i, S) ≤ maxV × Hd (i, g) or Hd (S, g) ≤ (1 − maxV) × Hd (i, g), for a maximum of maxV × Hd (s1, s2) operations where maxV is a distance parameter.

4.2 Mutation

The mutation operator is designed such that essentially performs a swap defining the N1 neighbourhood of the individual being mutated. More specifically, two operations belonging to the critical path are randomly selected and their positions in the chromosome are swapped. As already mentioned, such a mutation results in a feasible schedule, so there is no need to apply the GT algorithm as a repair method to bring the new individual back into the set of feasible schedules.

4.3 Subpopulation Processing

The initial population of size N is divided into k equally sized subpopulations or "villages" that evolve independently. When fitness stagnates in a village, due to the increasing number of members in theLsubset, the village is "merged" with another one (as described below), resulting in k − 1 remaining subpopulations (of generally different size).

Algorithm 2: Subpopulation Processing
1) For each subpopulation r, consider the following:
   a) Identify the best solution S∗r.
   b) Create subset Lr consisting of individual's Si for which the following holds true: Cmax(Si) < ρ ·Cmax (S∗r).
2) For each subpopulation r, check whether fitness stagnates by calculating the Hamming distance of all the individual's Si, which represent different solution schedules from the population's bestsolution in Lr.
3) When fitness stagnates in subpopulation q, consider the following:
   a) Merge subpopulation q with population j if |Lj| < |Lr|, ∀ r = 1, . . ., k, r = j, r = q and the ρ-backbones of the two populations are "not" the same.
   b) Reduce subpopulation number to k − 1.
4) After crossover and mutation, the population is updated and an offspring may enter the population of the next generation only if its fitness value is better than the worst member of the current generation, implementing therefore an elitist strategy.

5. Results and Discussions

The design and implementation of proposed GAPr method is done by using MATLAB by considering the different JSS problems such as Dmu07, YN01, YN04, LA38, 3x3 and 6x6. The comparative study is performed between existing GA based approach and proposed GAPr method in terms of three vital performance metrics. Below graphs are showing the comparative study among both methods. The simulation is done by considering below parameters settings:
Number of particles: 20
Number of iterations: 500
Mutation ratio: 0.35

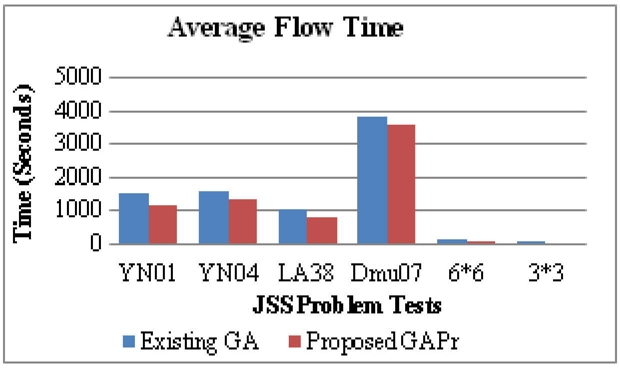

Figure 2: Performance Analysis of Average Flow Time

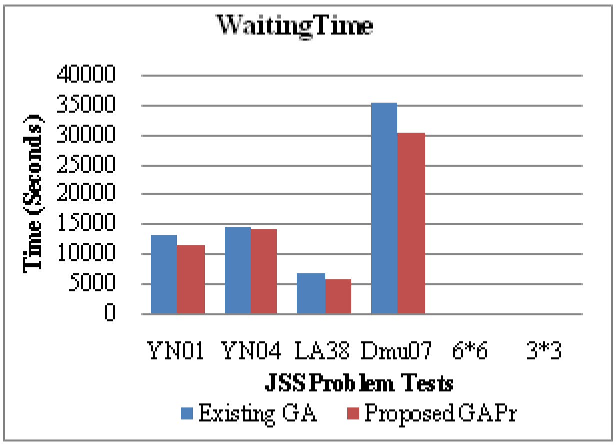

Figure 3: Performance Analysis of Waiting Time

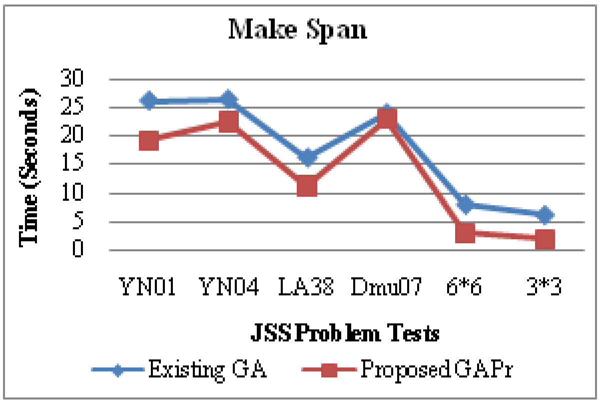

Figure 4: Performance Analysis of Total Completion Time (Make span)

The simulation results showing in figure 2,3,4 for average flow time, waiting time and make span time respectively claiming our proposed method outperforming the existing method for all the tests performed with different problems.

6. Conclusion and Future Work

In this paper, new improved method proposed for efficiency in solving the JSS problems. This new method is called GAPr. The parallel execution of GA method is performed by modifying the existing operators of GA such as mutation, crossover. The algorithms utilized here is path re-linking and offspring in order to execute the tasks parallel while solving the any JSS problem. The experimental results showing the proposed approach is having more efficiency in terms of average flow time, waiting time and make span time. The efficiency improvement is done by approximate 35 %. Future work will be real time testing and analysis of proposed work.

References

[1] Davis, L. Job Shop scheduling with genetic algorithms. In: Grefenstette (ed.), Proceedings of the First International Conference on Genetic Algorithms and Their Applications. Lawrence Erlbaum: Hillsdale, NJ, 1985, 136 – 140.
[2] Goldberg, D.E. Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, 1989.
[3] Beasley, D.; Bull, D.R.; Martin, R.R. An overview on genetic algorithms: Part I, Fundamentals. University Computing, 1993, 15, 58 – 69.
[4] Rechenberg, I. Evolutionsstrategie. OptimierungtechnischerSystemenachPrinzipien der biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973.
[5] Schwefel, H.P. NumerischeOptimierung von Computer-Modellenmittels der Evolutionsstrategie.Birkhäuser, Basel, 1977.
[6] Mikkel T. Jensen, “Robust and flexible scheduling with evolutionary computation,” Ph.D. dissertation, Dept. Comput. Sci., Univ. Aarhus, Aarhus, Denmark, 2001.
[7] S. Hajri, N. Liouane, S. Hammadi, “Evolution programs for job-shop scheduling,” in IEEE/SMC Conf. Proc., Orlando, FL, 1997, pp. 720–725.
[8] Min Liu, Zhi-jiang Sun, Jun-wei Yan, Jing-song Kang, A Genetic algorithm for parallel simulated annealing. In Proceedings of international conference on parallel problem solving from nature. Netherland.
[9] Yang Xiaomei1, Liang Jiye, Theory, Algorithms, and Systems. Springer Publishing Company, Incorporated, 2008
[10] Jianshuang Cui, Parallel GRASP with path-relinking for job shop scheduling. Parallel Computing 2003; 29:393–430.
[11] Jason Chao-Hsien Pan, Han-Chiang Huang, A hybrid genetic algorithm for the job shop scheduling problems. Computers & Industrial Engineering 2003; 45:597–613.
[12] Yan-wei Zhao & Hai-yan Wang & Xin-li Xu & Wan-liang Wang, (2008) A batch splitting job shop scheduling problem with bounded batch sizes under multiple-resource constraints using genetic algorithm. Proceedings of the 2008 IEEE International Conference on Cybernetics and Intelligent Systems. 220–225

[13] Omid Gholami, and Yuri N. Sotskov, 2012. “Job-shop Problems with Objectives Appropriate to Train Scheduling in a Single-track Railway.” In SIMULTECH 2012 – Proceedings of 2nd International Conference on Simulation and Modelling Methodologies, Technologies and Applications, Rome, May 21, 2012, 425–430.

[14] Su Nguyen, Mengjie Zhang, Mark Johnston, Kay Chen Tan, “Evolving reusable operation-based due-date assignment models for job shop scheduling with genetic programming,” in Proc. 15th EuroGP, 2012, pp. 121–133.

[15] Hong Li YIN, “New search spaces for sequencing problems with applications tojob-shop scheduling.” Management Science, vol.38, No.10, pp. 1495-1509.