



An Approach to Solve Single Machine Job Scheduling Problem using Heuristic Algorithm

Satyasundara Mahapatra¹, Dr. Rati Ranjan Dash², Dr. Sateesh Kumar Pradhan³

¹Department of Computer Science, IISIT, Bhubaneswar, INDIA

²Department of Mechanical Engineering, CET, Bhubaneswar, INDIA

³Department of Computer Science, Utkal University, Bhubaneswar, INDIA

Abstract: Single Machine Job Scheduling (SMJS) problem is a scheduling problem where a batch of n jobs scheduled to be processed through a single machine. The significant of these types of problems are their own inherent value, as well as their role as building schedules by the scheduler for more generalized and complex problems. To enhance the scheduler efficiency of a router in computer networks an efficient and optimized scheduling algorithm is required. This seems to have improved the Quality of Service (QoS) of routers in computer networks. Minimizing the Waiting Time Variance (WTV) is one of the vital criteria for developing a heuristic algorithm. As minimizing the WTV is a NP-Hard problem, a heuristic based novel solution approach is intended in this paper. This approach illustrated with small size problems along with randomly new generated problems. The experimental results are compared with existing approaches. The experimental result shows the proposed approach gives best performance for the compared approaches and problems tested. The findings of intended algorithm are shown in the form of table for consonant problems.

Keywords: Job Scheduling, Quality of Services, Single Machine, Waiting Time Variance, Computer Network.

I. Introduction

Scheduling is a decision-making problem used on a regular basis in many manufacturing and service provider industries. This mechanism deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. The form of resources (machines) and tasks (jobs) are varying from organization to organization. In computer and network organization routers are treated as resources and data packets are treated as jobs. Each job may have a certain priority level, an earliest possible starting time, a due date etc. Hence 'Timeliness' is one of the major attribute of QoS to measure the performance of a process. Provide a schedule by minimizing WTV in a single machine is the objective of this paper.

SMJS problems are important for various reasons. The environment of single machine is very simple and designed for special cases. The most important properties of single machine are neither machines in parallel nor machines in series. The results of single machine not only provide insights into the single machine environment but also provide a basis for heuristics that are applicable to more complicated machine environments. In general, scheduling problems in more complicated machine environments are often decomposed into sub-problems that deal with single machines. In the single machine job scheduling problem 'n' number of jobs has to be processed by a single machine with one or more type of processing objectives, order and constraints. In consideration of a job has to wait for processing due to non availability of the machine in real time, as it is processing another job and may also due to the precedence process constraint. Many exact and heuristic algorithms have been proposed over the years by the researchers for solving the different SMJS problems. Research on design and implementation of scheduling algorithms has been carried out by many scholars to provide an improved QoS for Internet applications in network.

There are number of studies has been done by considering the variance minimization of job completion time or completion time variance (CTV) in due course of time by the scientists. A tabu search-based procedure, a dynamic programming algorithm and a branch and bound algorithm for the CTV problem have been developed and discussed in [1][19][3]. Several heuristic methods have been discussed for the CTV problem in [9][18][5]. The appropriateness of considering non-regular measures of performance with respect to scheduling problems is discussed in [12]. They considered the problem of scheduling n jobs on a single machine so as to minimize the variance of job completion times, called the CTV problem. The V-shaped property of the optimal sequence for CTV and WTV is explain in [2][5]. The problems of scheduling with non-regular penalty functions are discussed in [9][18].

There are only a few studies are done on job scheduling to minimize WTV, although a job sequence for WTV and a job sequence for CTV are discussed in [12]. Two heuristic methods are discussed in [5] for better performance in reducing WTV for a number of small data sets. Another two heuristic methods named as Verified Spiral (VS) and Balanced Spiral (BS) are also discussed in [15] and compared with the methods

discussed in [5]. In a computer and network system, it is desirable to generate the sequence by minimize WTV rather than CTV, as CTV relies on the variable sizes of jobs.

This paper proposes a heuristic based novel approach as a solution to the problem with an effect of minimization of WTV where the jobs are distributed in the scheduling sequence with an extremely less computational cost. The effectiveness with respect to WTV is done by comparing the result with existing heuristic methods. The findings are shown in the form of table for corresponding problems.

The rest of the paper is structured as follows. The problem of WTV is briefly described in Section 2, In Section 3, different existing heuristic methods are discussed along with the proposed heuristic method with pseudo code. In Section 4 problems are generated for the testing as examples, In Section 5 the heuristic algorithms are implemented with the generated problems and the effectiveness with respect to WTV is tested and Section 6 contain the conclusion of the work and suggestion for future studies.

II. Problem Formulation for Job Scheduling Problem

The Single Machine Job Scheduling problem by minimization of WTV can typically describe as an allocation of jobs to a machine by considering the concept that once a job get in to the machine for processing, it was not leaved from the machine until the processing time of that job was over. The decision whether the job “ j ” (i.e the job number) is scheduled to the machine successfully, then “ i ” the allocation variable is 1 (One) or 0(Zero) otherwise, which can be represented by an integer. These decision variable depends upon the position of job in the job sequence, which is represented by “ d_{ij} ” for $i=1,2,\dots,n$ and $j=1,2,\dots,n$. The job to be scheduled first is placed at first position thus processed first; the job to be scheduled second is placed at second position thus processed second, and so on. Then the waiting time for job j at position i is represented as ‘ wt_i ’ and the processing time of job j is represented by ‘ pt_j ’. The WTV of jobs in a complete sequence is obtained as follows in equation (1).

$$WTV = \frac{1}{n-1} \sum_{k=1}^n \left(wt_k - \frac{1}{n} \sum_{i=1}^n wt_i \right)^2 \quad (1)$$

The objective is to minimize the variance of waiting time of n number of jobs can be found by the following equation (2).

$$WTV_{Minimize} = Minimize \frac{1}{n-1} \sum_{k=1}^n \left(wt_k - \frac{1}{n} \sum_{i=1}^n wt_i \right)^2 \quad (2)$$

Subject to:

$$\sum_{i=1}^n d_{ij} = 1, \quad i = 1, 2, \dots, n; \quad (3)$$

$$\sum_{j=1}^n d_{ij} = 1, \quad j = 1, 2, \dots, n; \quad (4)$$

$$d_{ij} = 0 \text{ or } 1; \quad i, j = 1, 2, \dots, n; \quad (5)$$

$$wt_1 = 0; \quad (6)$$

$$wt_i = wt_{i-1} + \sum_{j=1}^n d_{i-1,j} * pt_j, \quad i = 1, 2, \dots, n; \quad (7)$$

The constraint that each position of the sequence is used exactly once by a job is described in equation (3). Each job is assigned to a position in the sequence is exactly once is described in equation (4). The integer constraint for decision variable is described in equation (5). The waiting time for first job is described in equation (6) and the waiting time wt_i of the job at position i ($i \geq 2$) is described in recursive equation (7).

III. Job Scheduling Methods

In the literature, various job sequencing methods are developed in course of time and also improved further. Out of these some are exact method and some others are heuristic or meta-heuristic method. To search the best all possible sequence has to be tested with desired objective one by one. For small problems this is possible, but it becomes complex and difficult as the problem size increase. Hence some heuristic methods are developed and adapted to obtain an optimum solution without calculating the total possible combinations of sequences. As one by one calculation expenses a lot of computer resources. Such as Computer memory, Computer network, CPU, time and so on. On the other side heuristic methods are good for providing a near optimum solution without searching for the most optimum solution with fewer resources (i.e time). These heuristics are used for calculation of CTV and WTV.

Out of number of heuristic methods Verified Spiral (VS) and Balanced Spiral (BS) are described below with the help of Pseudo code.

A. Verified Spiral (VS)

The Verified Spiral (VS) method developed by incorporating Schrage's conjecture and Hall and Kubiak's proof about the placement of the first, second and third longest jobs are described in [7]. And then, modify the spiral placement of remaining jobs by adding a selection between two positions to place the next job, either before the tail (end) or after the head (beginning) of the job sequence, based on which position produces a smaller variance of waiting times for jobs already in the sequence.

Pseudo code of the Verified Spiral method

procedure VS

Arrange all the jobs of job pool in ascending order on the basis of their processing time.

Pick the n^{th} job from the job pool and placed in the last position of the job sequence.

Pick the $(n-1)^{\text{th}}$ job and placed in the last-but-one position of the job sequence.

Pick $(n-2)^{\text{th}}$ job and placed in the first position of the job sequence.

Pick the 1^{st} job and placed in the second position of the job sequence.

repeat

Remove the longest job from the job pool and place the longest job either before the first job or after the first job in the job sequence, depending on which position of the job sequence produces a smaller WTV so far.

until (the job pool is empty)

return (arranged sequence)

end procedure

B. Balanced Spiral (BS)

To reduce the computational cost associated with the VS method, the Balanced Spiral (BS) method replaces the verification computational cost of WTV during the placement of individual job selection process for job sequencing. The developed BS method is based on an observation of balanced left (L) and right (R) optimal sequence of the processing time of individual jobs to get near optimum job sequence. To explain the balancing behavior of a job sequence problem is considered. The basic concept is to balance a job sequence arrived in ascending of their job processing time. By locating the processing time of different jobs on the right side (R) and left side (L) such that approximate sum of the processing time of jobs on both the side are almost same is known as Balance. For optimal sequence the first longest job is placed last, the second longest job is placed first, and the third longest job is placed in last-but-one. The rest of the jobs in the job pool are placed on right side or left side by balancing the processing time of individual.

Pseudo code of the Balanced Spiral method

procedure BS

Arrange all the jobs of job pool in ascending order on the basis of their processing time.

Pick the n^{th} job from the job pool and placed in the last position of the job sequence.

Pick the $(n-1)^{\text{th}}$ job and placed in the last-but-one position of the job sequence.

Pick $(n-2)^{\text{th}}$ job and placed in the first position of the job sequence.

/(Note: Let the sequence L store the processing time of $(n-2)^{\text{th}}$ job and sequence R store the processing time of $(n-1)^{\text{th}}$ job. Remember that processing time of n^{th} job is not including in R.)//

Set $SUM_L \leftarrow$ sum of the processing times of all jobs in L sequence.

Set $SUM_R \leftarrow$ sum of the processing times of all jobs in R sequence.

Set $k \leftarrow 4$ //k represent for counter

Repeat

If ($SUM_L < SUM_R$)

Append the k^{th} job to the last position of the L sequence.

$SUM_L \leftarrow SUM_L +$ Processing time of k^{th} job.

End if

If ($SUM_L \geq SUM_R$)

Add the k^{th} job to the fast position of the R sequence.

$SUM_R \leftarrow SUM_R +$ Processing time of k^{th} job.

End if

$k \leftarrow k+1$

until (the job pool is empty)

return (arranged sequence)

end procedure

C. Proposed Method (RSS)

In our locality, the fishmonger are sales a complete unit of fish. Sometimes a large fish has to be distributed equally to two or more customers. These fishmongers are skilled enough to distributes along with the time of cutting to reduce the post measuring for equality and it is generally are most equal. This distribution style to serve two customers is used in this proposed method, which named as RSS. This is a method of allocation of jobs in the scheduling sequence with an extremely less computational cost with sorting of job on the basis of their processing time.

Pseudo code of the proposed RSS method

procedure RSS

Arrange all the jobs of job pool in ascending order on the basis of their processing time.

```

n ← total number of elements inside the job pool
tnp ← n //tnp: Total number of elements left inside the job pool for place in job sequence //
ctr ← 1
repeat
  if (ctr is an odd number)
    if (tnp < 2)
      Add only one job from kth position backwards to the fast position of the RIGHT sequence and Decrement tnp value by 1.
    else
      Add two jobs from kth position backwards to the fast position of the RIGHT sequence and Decrement tnp value by 2.
  end if
  if (ctr is an even number)
    if (tnp < 2)
      Append only one job from kth position backwards to the last position of the LEFT and sequence Decrement tnp value
      by 1.
    else
      Append two jobs from kth position backwards to the last position of the LEFT sequence and Decrement tnp value by 2.
    end if
  end if
  ctr ← ctr + 1
until (ctr ≤ n/2)
return (arranged sequence)
end procedure

```

The effectiveness of the proposed RSS method is shown in the next section by compared with VS and BS methods.

IV. Problem for testing

In the above section three heuristic algorithms are depicted. In this section for finding the effectiveness of these algorithms some problems are selected. Twenty four small test cases have been selected to find the effectiveness of the algorithms. The number of jobs varies from 5 to 10 in these cases. There are twelve integer time cases and twelve are of non-integer cases are considered. The data has generated with the help of Normal, Poisson, Exponential and Uniform distribution for all the cases. The same is represented in Table – 1. The case number and the number of jobs along with the table, shows the processing times of the jobs in order.

TEST CASE NO.	NO.OF JOBS	PROCESS TIME									
1	5	19.13	23.08	24.32	12.92	21					
2	5	22	21	20	31	12					
3	6	27.2	30.16	15.11	16.79	22.6	18.1				
4	6	20	21	23	18	9	20				
5	7	18.31	22.64	25.28	20.07	28.07	15.92	10.01			
6	7	21	17	21	22	22	17	19			
7	8	14.69	22.25	18.64	19.49	12.85	16.18	22.05	16.05		
8	8	13	28	27	12	22	18	21	9		
9	9	8.35	12.75	21.67	21.96	22.26	19.35	20.92	17.62	24.31	
10	9	23	18	14	25	22	17	19	23	24	
11	10	20.69	10.46	18.18	15.76	16.18	14.36	20.39	30.53	16.42	18.6
12	10	13	23	15	12	18	16	20	27	20	26

Table – 1: Twelve small size test cases with their processing time.

V. Performance Analysis of the Algorithms

To measure the performance of the proposed RSS algorithm, all possible sequences are generated by placing the jobs randomly for each problem of Table - 1. Each generated possible sequence is considered as one sub example of all possible Optimal Sequences. There are 120 numbers job sequences (e.g.5!) as there are five jobs in the test case – 1 & 2. There are 720 numbers of job sequences (e.g.6!) as there are six jobs in the test case – 3 & 4. There are 5,040 numbers of job sequences (e.g.7!) as there are seven jobs in the test case – 5 & 6. There are 40,320 numbers of job sequences (e.g.8!) as there are eight jobs in the test case – 7 & 8. There are 3,62,880 numbers of job sequences (e.g. 9!) as there are seven jobs in the test case – 9 & 10. There are 36,28,800 numbers of job sequences (e.g. 10!) as there are ten jobs in the test case – 11 & 12. But the above discussed three job scheduling heuristic methods generate only one job sequence for each test case of Table - 1. The basic aim is to calculate WTV for the cases and optimize it and the optimum sequence is always V-shaped. The WTV and Mean Waiting Time (MWT) for test case – 1 to test case – 12 of Table – 1 are shown in Table – 2 and Table – 3 and are rounded to four significant digits using double precision arithmetic.

Table – 2 shows the Comparative results of WTV by three heuristic methods along with optimal method for twelve small size test cases. It was found VS method gives ten numbers of optimum results and two numbers of

near optimum results, and BS method gives eight optimum results and four near optimum results. The proposed RSS method gives nine optimum results and three near optimum results. All the sequence obtained by proposed RSS method in each test case is also satisfying the V-shape property.

TEST CASE NO.	NO.OF JOBS	OPTIMUM	VS	BS	RSS
1	5	854.4793	854.4793	854.4793	854.4793
2	5	834	834	834	834
3	6	1269.9586	1269.9586	1269.9586	1269.9586
4	6	1003.6	1003.6	1003.6	1003.6
5	7	1451.6582	1451.7374	1451.985	1451.7374
6	7	1693.6191	1693.6191	1693.6191	1693.6191
7	8	1614.6995	1614.6995	1614.9288	1614.6995
8	8	1533.0715	1533.0715	1533.1429	1533.0715
9	9	2184.1714	2184.1714	2184.1714	2184.1714
10	9	2747.3612	2747.3612	2747.3612	2747.3612
11	10	2334.2729	2334.295	2334.295	2334.295
12	10	2599.6112	2599.6112	2599.8778	2599.8778

Table – 2: Comparative results of WTV by three heuristic methods along with Optimal method for twelve small size test cases.

The table – 3 given below shows the comparative results of MWT by five heuristic methods for twelve small size test cases. Overall The MWT value of proposed RSS method is considerable better than all the heuristic methods discussed above along with the optimum method.

TEST CASE NO.	NO.OF JOBS	OPTIMUM	VS	BS	RSS
1	5	38.062	38.062	38.062	38.062
2	5	37	38	38	38
3	6	48.585	48.585	48.585	48.585
4	6	44	44	44	44
5	7	55.2172	55.1272	55.9715	55.1272
6	7	58.4286	58.5715	58.5715	58.5715
7	8	59.46	59.46	59.8	59.46
8	8	59.75	59.75	60	59.75
9	9	72.3923	72.4878	72.4878	72.4878
10	9	79.8889	79.8889	79.8889	79.8889
11	10	75.606	75.618	75.618	75.618
12	10	80.5	80.5	80.9	80.1

Table -3: Comparative results of MWT by three heuristic methods along with Optimal method for twelve small size test cases.

Twelve small sized test cases are solved by five heuristic methods discussed in section 3 including proposed RSS method. Each method generates a sequence of jobs with respect to WTV and MWT based on a particular arrangement. The deviation performance of each method's sequence is measured by the WTV and MWT. The percentage of deviation of WTV is represented by WTV_{PD} and defined as follows:

$$WTV_{PD} = \frac{WTV_{Method} - WTV_{Optimal}}{WTV_{Optimal}} \times 100 \quad (8)$$

Where

$WTV_{Optimal}$:The minimum WTV produced by Optimum method.

WTV_{Method} :The WTV generated by the individual heuristic method as discussed above

And the percentage of deviation of MWT is represented by MWT_{PD} and defined as follows:

$$MWT_{PD} = \frac{MWT_{Method} - MWT_{Optimal}}{MWT_{Optimal}} \times 100 \quad (9)$$

Where

$MWT_{Optimal}$:The minimum MWT produced by the Shortest Path Method.

MWT_{Method} :The MWT generated by the individual heuristic method as discussed above

Table-4 given below shows the comparative performance deviation results of WTV for twelve small size test cases by three heuristic methods. Smallest value of WTV_{PD} indicates the best performance of the heuristic method among the three heuristic methods. VS gives optimal sequence in ten test cases and near optimum in two test cases. BS gives optimal sequence in seven test cases and near optimum in five test cases. The proposed RSS gives optimum in nine test cases and near optimum in three test cases. It was found that in some test cases

the WTV_{PD} values of VS, BS and proposed RSS are same. And in some cases it was also found that the proposed RSS gives better near optimum result than BS method.

TEST CASE NO.	NO.OF JOBS	VS	BS	RSS
1	5	0	0	0
2	5	0	0	0
3	6	0	0	0
4	6	0	0	0
5	7	0.005456	0.022516	0.005456
6	7	0	0	0
7	8	0	0.014198	0
8	8	0	0.004659	0
9	9	0	0	0
10	9	0	0	0
11	10	0.000946	0.000946	0.000946
12	10	0	0.010258	0.010258

Table – 4: Comparative performance deviation results of WTV for Twelve small size test cases

The comparative performance deviation results of MWT for twelve small size test cases are given below in Table – 5. Minimum value of MWT_{PD} indicates the best average performance on mean waiting time of a heuristic method out of the three heuristic methods. As shown on the table given below, the overall average performance of proposed RSS method is better than other heuristic methods.

TEST CASE NO.	NO.OF JOBS	VS	BS	RSS
1	5	9.279357	9.279357	9.279357
2	5	10.46512	10.46512	10.46512
3	6	8.203111	8.203111	8.203111
4	6	10.92437	10.92437	10.92437
5	7	12.28504	14.00471	12.28504
6	7	5.128205	5.128205	5.128205
7	8	8.106634	8.724802	8.106634
8	8	17.73399	18.2266	17.73399
9	9	13.77771	13.77771	13.77771
10	9	8.774584	8.774584	8.774584
11	10	9.82847	9.82847	9.82847
12	10	12.58741	13.14685	12.02797

Table – 5: Comparative performance deviation results of MWT for Twelve small size new test cases

To know the effectiveness of the WTV_{PD} and MWT_{PD} of individual heuristic methods a performance deviation analysis was done by comparing the methods among themselves. Each method was tested with 1000 small sized test cases where the jobs are between 5 to 10 numbers. These 1000 test cases are generated with the help of four distribution methods named as Normal, Poisson, Exponential and Uniform distributions. Table – 6 shows that in case of Normal distribution the proposed RSS method is 74.5834% better than VS and 75% better than BS. Table – 7 shows that in case of Poisson distribution the proposed RSS method is 77.9167% better than VS and 80.4167% better than BS method. Table – 8 shows that in case of Exponential distribution the proposed RSS method is 58.3334% is better than VS and 72.0834% better than BS method. And Table - 9 shows that in case of Uniform distribution the proposed RSS method is 85% better than VS and 85.8334% better than BS method.

	BS	RSS
VS	82.5	74.5834
BS		75

Table – 6: Percentage result of WTV_{PD} of small size test cases generated by Normal Distribution

	BS	RSS
VS	92.5	77.9167
BS		80.4167

Table – 7: Percentage result of WTV_{PD} of small size test cases generated by Poisson Distribution

	BS	RSS
VS	61.6667	58.3334
BS		72.0834

Table – 8: Percentage result of WTV_{PD} of small size test cases generated by Exponential Distribution

	BS	RSS
VS	97.9167	85
BS		85.8334

Table – 9: Percentage result of WTV_{PD} of small size test cases generated by Uniform Distribution

In the processes of generating an efficient job sequence schedule by the different heuristic methods the computational average time is treated as computational cost. From the above discussed heuristic methods it was

found that except optimal method all the heuristic methods are needed a sorting mechanism at the beginning of the process. In other word it can be says that only sorted data are required for processing through any heuristic methods. Among the entire available sorting techniques Quick sort is a sorting technique which takes $O(n \log n)$ times for computation. The computational cost of VS method is much larger then BS and proposed RSS method, as the calculation of WTV is made multiple times as may be required in a test case on the basis of the job numbers, and the computational cost of BS method is also larger than the proposed RSS method, as the calculation of total processing time is made multiple times as may be required in a test case on the basis of job numbers. As a result by minimizing WTV an optimum or near optimum sequence in comparatively very less and more acceptable computational cost has produced by using the concept of cutting a large fish into small pieces and distributed among the customers uniformly by a fishmonger's named as 'RSS'.

VI. Conclusion

Application of proposed RSS heuristic method for minimizing WTV in job scheduling is described in this article with pseudo code. Two existing heuristic methods namely VS and BS methods are also described with the help pseudo code. Twelve numbers of small size test cases where the job varies from 5 to 10 numbers are generated randomly with the help of four distribution methods namely Normal, Poison, Exponential and Uniform distribution. These twelve small size test cases are again tested with RSS method and two existing heuristic methods. The experimental results are compared and found that the generated job sequence for individual test case is optimum or near optimum with minimum computational cost. To know the performance of WTV_{PD} , the proposed RSS method is again tested and compared by two existing methods with 1000 numbers small size test cases randomly generated with four distribution methods. It was also found that the proposed RSS method gives optimum or near optimum sequences with minimal computational cost. To strengthen the service quality of the resources like router for scheduling data packets, scheduling services of web request in a web server in the computers and network the proposed RSS method may be used. This proposed method can also be applied for scheduling jobs in batch on a manufacturing unit of an organization. In future this work will be extended with large size of test cases generated by different distribution methods.

VII. References

- [1] Al-Turki U, Fediki C and Andijani A Tabu search for a class of single-machine scheduling problems. *Computers and Operation Research*. 28(12):1223–30, 2001.
- [2] Cai X V-shape property for job sequences that minimize the expected completion time variance. *European Journal of Operational Research*. 91(1): 118–23, 1996.
- [3] De P, Ghosh JB and Wells CE On the minimization of completion variance with Bicriteria Extension. *Operation Research*. 40(6): 1148, 1992.
- [4] De P, Ghosh JB and Wells CE Scheduling to minimize the coefficient of variation. *International Journal of Production Economics*. 44: 249–53, 1996.
- [5] Eilon S and Chowdhury IG Minimizing waiting time variance in the single machine problem. *Management Science*. 23: 567–74, 1977.
- [6] Gowrishankar, K., Rajendran, C. and Srinivasan, G. Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *European Journal of Operational Research*. 132: 643–665, 2001.
- [7] Hall NG and Kubiak W Proof of a conjecture of Schrage about the completion time variance problem. *Operations Research Letters*. 10(8): 467–72, 1991.
- [8] Jurisch B, Kubiak W and Jozefowska J Algorithms for minclique scheduling problems. *Discrete Applied Mathematics*. 72(1–2): 115–39, 1997.
- [9] Kanet JJ Minimizing variation of flow time in single machine systems. *Management Science*. 27(12): 1453–9, 1981.
- [10] Kubiak W New results on the completion time variance minimization. *Discrete Applied Mathematics*. 58(2): 157–68, 1995.
- [11] Manna DK and Prasad VR Pseudo polynomial algorithms for CTV minimization in single machine scheduling. *Computers and Operations Research*. 24(12): 1119–28, 1997.
- [12] Merten AG and Muller ME ,Variance minimization in single machine sequencing problems. *Management Science*. 18: 518–28, 1972.
- [13] Mittenthal J, Raghavachari M and Rana AI V- and Λ -shaped properties for optimal single machine schedules for a class of non-separable penalty functions. *European Journal of Operational Research*. 86(2): 262–9, 1995.
- [14] Mosheiov G Minimizing mean absolute deviation of job completion times from the mean completion time. *Naval Research Logistics*. 47(8): 657–68, 2000.
- [15] Nong Ye, Xueping Li, Toni Farley and Xiaoyun Xu Job scheduling methods for reducing waiting time variance. *Science Direct. Computer and Operation Research* 34: 3069-3083, 2007.
- [16] Postel J The TCP maximum segment size and related topics. RFC 879, 1983.
- [17] Schrage L Minimizing the time-in-system variance for a finite jobset. *Management Science*. 21(5): 540–3, 1975.
- [18] Vani V and Raghavachari M Deterministic and random single machine sequencing with variance minimization. *Operations Research*. 35(1): 111–20, 1987.
- [19] Viswanathkumar G and Srinivasan G A branch and bound algorithm to minimize completion time variance on a single processor. *Computers and Operations Research*. 30(8): 1135–50, 2003.
- [20] Ye N QoS-centric stateful resource management in information systems. *Information Systems Frontiers*. 4(2): 149–60, 2002.
- [21] Ye N, Bashettiwalli H, Li X and Farley T Batch scheduled admission control for service dependability of computer and network resources. *Information Knowledge Systems Management* 5: 211–226, 2006.